User Guide

FortiMonitor 25.3.0

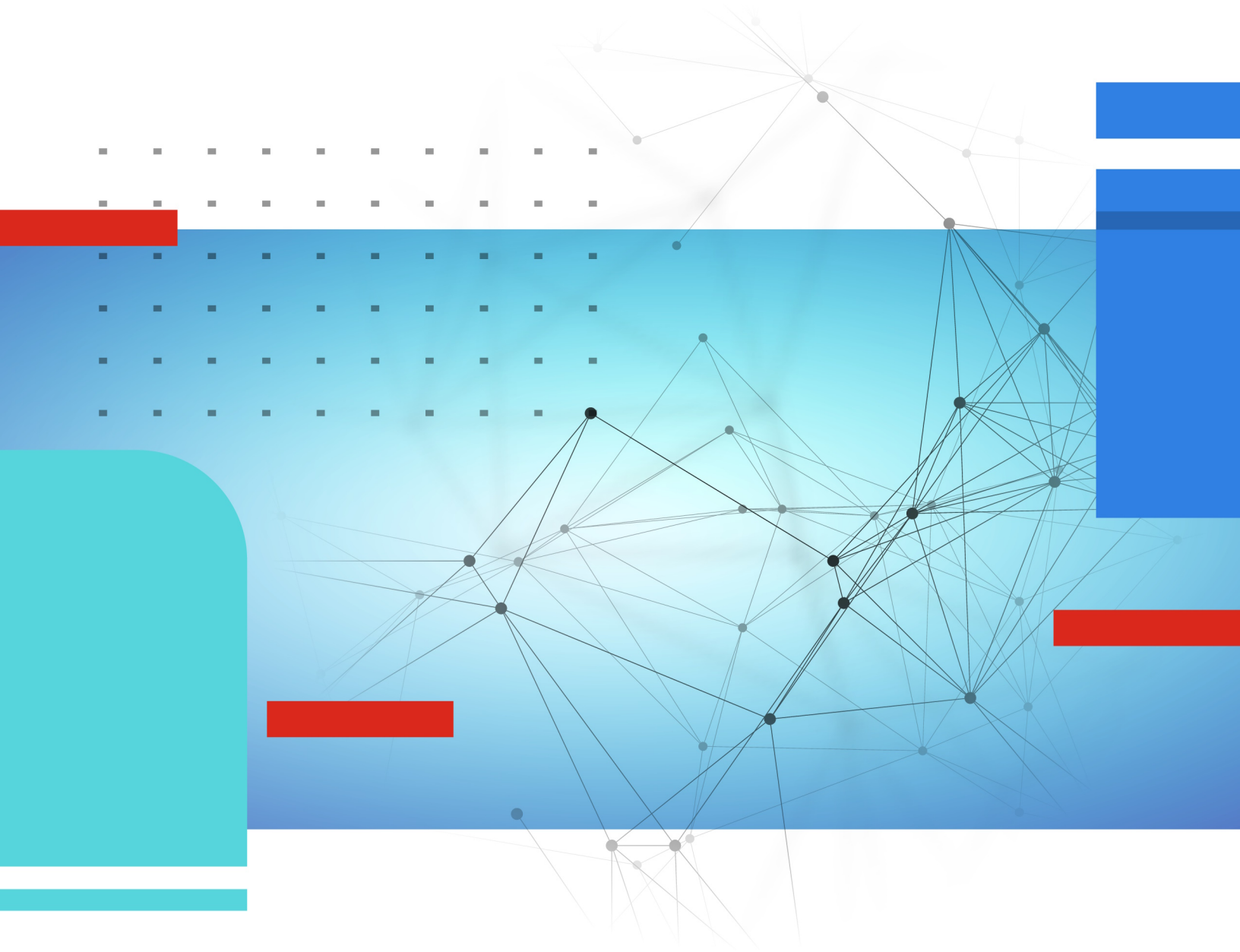


TABLE OF CONTENTS

Getting started **22**
Monitoring Approach 22
Part 1: Add your infrastructure to FortiMonitor 22
FortiMonitor Agent 23
OnSight vCollector installer 24
OnSight Discovery 25
FortiMonitor Cloud Discovery 25
Add a basic instance 25
Add via API 26
Part 2: Monitoring 26
Monitoring types 26
Basic methods 26
Advanced methods 28
Example manifest file 30
Part 3: Alert Timelines 31
Alert Timelines 31
Maintenance period 35
Acknowledging an incident 35
Part 4: Visualization 36
Infrastructure Map 36
Dashboard 37
Instance details page 39
Status page 40
Part 5: Team Management 41
Users and groups 42
Access control 43
On-Call schedules 44
Single Sign-On (SSO) 44
Part 6: Reports 46
Canned reports 47
Scheduled reports 48
Report history 48
Ad-hoc reports 48
Part 7: CounterMeasures 49
Slack integration 49
Enable CounterMeasures 49
Example workflow 50
CounterMeasure actions 50
CounterMeasure metric threshold 51
CounterMeasure status 51
CounterMeasure output 52
Billing 52
Supported browsers 53
Chrome 53
Firefox 53
Safari 53

Internet Explorer 53
Infrastructure Map 54
Clustering 54
Diving 54
Filtering 55
Views 56
Users, Groups, and Authentication 57
Add users to your account via FortiMonitor 57
Add users using FortiMonitor 57
Sync users from FortiCloud 59
Create FortiCloud users from FortiMonitor 60
Access control 62
Create a Group 63
Single sign-on (SSO) 64
Control Panel Configuration 64
General 64
Certificate 65
User Configuration 65
Mixing login types 66
SSO configuration for Multi-tenant users 66
Set up SSO Configuration for SAML + ADFS 3.0 66
Set up SSO with Okta 84
Set Up SSO with Azure AD 92
Create an SSO integration in FortiMonitor 93
Enable and configure Azure Active Directory SSO in Azure 94
Create a SAML signing certificate and download the Azure metadata 97
Continue the SSO configuration in FortiMonitor 99
Upload the metadata.xml file from FortiMonitor to Azure 103
Set up SSO with Google Workplace 104
Availability 111
Actions 112
Details tab 113
General information 113
Device information (Fabric devices) 113
Device information (SNMP devices) 113
Timelines 114
Locations 114
Settings 115
Others 115
Monitoring tab 116
Config 116
Performance 116
Incidents tab 118
Recent 118
Custom config 118

Maintenance 119
Activity 120
Assign Alert Timelines 120
Primary Alert Timeline 120
Agent Heartbeat Alert Timeline 121
Metric Heartbeat Alert Timeline 122
SNMP Heartbeat Alert Timeline 123
Create an Instance Group 124
Multi-region support 126
Overview 127
Data residency 127
Data residency vs. monitoring location 127
Installing Agents, OnSights, and NetFlow appliances in a Multi-Region setup 127
Regions 128
Requirements 129
User access 130
Selecting a region 131
Adding a user 132
Add a user using FortiMonitor 132
Add a user using FortiCloud 132
My Account 133
Can I see what changes have been made to my account and by whom? 133
Manage email preferences 134
Dashboards 135
Create a dashboard 136
Configure dashboard widgets 137
Instance and metric tags 137
Dashboard-level scoping 138
Favorites and search 139
Setting a default dashboard 139
Dashboard widgets 140
140
Active Incident Counter 140
Active Incident Details 141
Advanced Line Graph 144
Aggregated Availability 146
Aggregated Metric 147
Bar Chart 149
Fabric topology widget 151
Gauge 154
Geomap 155
Use case 157
Example 157
Infrastructure Counter 158

Kubernetes Container Counter 159
Kubernetes Pod Counter 161
Line Graph 163
Metric Heatmap 165
Metric Rollup 166
Metric Summary 168
Network device widgets 170
Network Interfaces 171
Resolved Incident Counter 173
Resolved Incident Details 175
Server Heatmap 177
Instance Scorecard 178
Instance Summary Table 179
Single Metric 181
Stoplight 183
Up/Down 184
Worst Offenders 186
Clone a widget 190
Topology Maps 191
Data collection 192
Topology page 192
Basic controls 193
Filters and toggles 193
View instance details 193
Monitor an unmonitored instance 195
Create and save views 195
Monitoring 197
No Unaddressed outages 197
No Persistent Alerts 197
FortiMonitor is an extension of your team 198
Add monitoring to an existing server 198
Monitoring check types 199
Monitor Flapping Metric incidents 199
Advanced monitoring and tuning 203
Automation 208
Tags and Attributes 212
Tags 212
Attributes 222
Use Cases 222
Creation 223
Examples 223
Bulk Upload 224
Monitoring locations 226
Specifying confirmation check locations 227
Global monitoring network 228
Advanced monitoring features 232
Compound Services 232
Server Templates 233

Forcing DNS Updates 233
SNMP 233
Compound metrics 233
Force a DNS update after an IP address change 234
Monitoring policies 235
Default rulesets 235
Custom rulesets 236
Filterable Dimensions 237
Tag matching logic 238
Actions 240
Test a ruleset 241
Test the Monitoring Policy Workflow 242
Monitoring guides 243
MySQL monitoring 243
Redis monitoring 246
Templates 248
Create a template from an existing server 249
Create a template from scratch 250
Default templates 251
Linux 252
Windows 255
Network devices 258
Virtual machines 258
Configure a template 259
Configure a new or existing template 259
Configure template metrics 264
Apply server templates 265
Apply a template manually from the Instance Details page 265
Apply a template manually from the template's page 266
Apply a template during initial instance configuration 266
Apply a template during Agent installation 267
Apply a template to an instance group 267
Metric override when applying templates 268
Template pattern matching 269
A brief example 269
Try it yourself 269
Disassociate a template from multiple instances 270
Cloud monitoring 270
AWS 271
Connecting FortiMonitor and CloudWatch 271
API Limits and Throttling 272
Existing Monitoring 272
EC2 Incident Confirmation 272
AWS integrations 272
Amazon DynamoDB 273
Amazon EBS 274
Amazon EC2 275

Amazon EC2 ASG 277
Amazon ElastiCache 278
Amazon ElasticSearch 279
AWS ELB 281
AWS ELBv2 282
AWS Lambda 284
Amazon RDS 285
Amazon RDS Cluster 286
Amazon Redshift 288
Amazon S3 289
Amazon SNS 291
Amazon FSx 291
Amazon SQS 292
AWS VPN 293
Azure 293
Connecting FortiMonitor and Azure 294
API Limits and Throttling 294
Existing Monitoring 295
Virtual Machine Incident Confirmation 295
Azure integrations 295
Azure App Service 295
Google Cloud Platform 297
Connecting FortiMonitor and GCP 297
GCP integrations 298
Kubernetes 298
Prerequisites 298
Resource requirements 298
Deploying FortiMonitor 298
Upgrading FortiMonitor 300
Uninstalling FortiMonitor 300
Kubernetes Details page 300
Application monitoring 306
Custom and default applications 307
Application monitoring usage examples 308
Overlay/underlay network quality 308
Measure application performance from multiple different vantage points 308
Get started with Application Monitoring 311
Application Monitoring page 311
Status and Availability 312
Monitoring locations 313
Add an Application 321
Add instances 322
Create checks 325
Add monitoring locations 326
Path monitoring 328
Prerequisites 328
Configure Path Monitoring 328

Change an existing configuration 329
View Path Monitoring 330
Timescale selector, filters, and toggles 330
Timescale selector and date 330
Filters and toggles 330
View node details 331
View link details 331
SD-WAN application monitoring 332
SD-WAN overview 332
FortiMonitor monitoring approach 332
Synthetic checks and network visibility 332
Get started with SD-WAN application monitoring 334
FortiGate and OnSight configuration for SD-WAN synthetic monitoring 334
Introduction 334
Prerequisites 336
Initial steps 336
FortiGate configuration 336
Add the firewall addresses to the firewall policy 338
FortiMonitor configuration 339
Set up SD-WAN application monitoring 341
Verify the configuration 344
Add SD-WAN synthetic checks to an existing application 345
Security Fabric 347
Monitoring different Fabric environments 348
Fortinet environment 348
Combination of Fortinet and non-Fortinet devices 348
Deployment options 349
FortiMonitor integration 351
Full template support 351
Fabric Topology dashboard widget 351
Network Interfaces dashboard widget 351
Network device performance monitoring 352
Note on FortiAPs and FortiSwitches 352
Supported metrics 352
Device health for FortiGates 353
Device health for FortiSwitches 353
Device health for FortiAPs 353
Device health for FortiExtender 353
Using FortiManager 354
Using SNMP 354
Limitations 356
Fabric Tunnel connected to FortiMonitor cloud (FortiOS 7.0 and newer) 356
Prerequisites 356
FortiMonitor configuration 360
Manage Fabric connections 364

Fabric 7.x OnSight proxy ... 365
Prerequisites ... 365
FortiMonitor Configuration ... 368
Fabric group ... 372
Prerequisites ... 372
Create an API Key and API User ... 373
FortiMonitor Configuration ... 373
Fabric integration via FortiManager proxy ... 377
Supported Fortinet devices ... 377
Prerequisites ... 377
FortiManager configuration ... 377
FortiMonitor configuration ... 379
Network device monitoring ... 381
Add a network device manually ... 381
Add network devices with OnSight Discovery ... 383
Configure the network device ... 384
Monitor metric performance ... 385
NetFlow monitoring ... 387
Supported NetFlow versions ... 387
Enable NetFlow monitoring ... 387
Installation modes ... 388
Standalone installer ... 388
Virtual appliance ... 389
Configure the NetFlow source device ... 389
View the NetFlow dashboard ... 390
FortiMonitor NetFlow commands ... 390
Estimator mode ... 391
Switch to Collector mode ... 391
VMware ... 392
Supported versions ... 392
Control panel configuration ... 392
Instance Tree ... 394
Infrastructure Map and Dashboards ... 394
Supported metrics ... 394
Host metrics ... 394
Cluster metrics ... 395
Datastore metrics ... 396
VMware Credentials page ... 396
Network checks ... 396
Network service checks ... 397
DNS ... 397
Exchange - OWA ... 398
FTP network check ... 399
Git ... 399
Jabber ... 400

Mail checks 400
MySQL network check 403
MSSQL network check 404
Network quality 404
Oracle network check 405
Ping 406
PostgreSQL network check 406
SSH 407
SSL expiration check 409
TCP network checks 411
Synthetic monitoring 412
Synthetic Monitoring check types 412
Test checks and Traceroute 413
Incidents 414
HTTP 414
HTTPS 420
Browser synthetic (multistep) check 425
Record browser interaction using Katalon 426
Metric configuration 426
Available multistep commands 428
JavaScript synthetic 430
Set up a JavaScript synthetic check environment 431
Configure incident alert and response time thresholds 434
Incident alerts thresholds 435
Response time thresholds 436
Dynamic variables 437
Standard Variables 438
Demonstration of Usage 438
FortiMonitor Agent 440
Benefits of using the FortiMonitor Agent 440
440
Linux 440
Windows 441
Install the FortiMonitor Agent 441
Install an Agent on a new server 441
Install the Agent on a server already in FortiMonitor 443
Installer parameters 443
Add a monitoring agent resource 446
Install the FortiMonitor Agent without root access 448
Install the FortiMonitor Agent on a Docker host 449
Update the FortiMonitor Agent 450
Redhat or CentOS 450
Debian or Ubuntu 450
Linux or Unix tar installer 450
Windows 450
MacOS 451
Uninstalling the FortiMonitor Agent 451
Add FortiMonitor Agent checks 452

FortiMonitor Agent Security 454
Network Communication + Encryption 454
Data and Metrics 455
Open Source 455
Non Administrative User 455
Network proxy configuration 455
Windows Agent 455
Linux Agent 456
MacOS Agent 456
Windows Agent Auto-update 456
New Windows instances 456
Oneliner 456
Manifest file 457
Enable Auto-update in existing Agent installs 457
Logging 458
Log file location 458
Update schedule 458
Installation logs 458
Add plugins to a blacklist (FortiMonitor Agent) 458
Linux 458
Windows 459
Add metrics using Windows Performance Monitor 459
Activating your metrics 462
Option 1: Activating your metrics via the Control Panel 462
Option 2: Activating your metrics via the agent config file 463
Prepare a FortiMonitor Agent for server imaging/cloning 463
Rebuild the FortiMonitor Agent metadata 464
IP matching 465
Start or end Maintenance from the FortiMonitor Linux Agent 465
Migrating a Panopta Agent configuration to FortiMonitor Agent 466
Linux Panopta Agent migration 466
Windows Panopta Agent migration 467
Troubleshooting the FortiMonitor Agent 468
Installation 468
Agent communication and connectivity 469
System time 469
FortiMonitor Linux Agent 469
FortiMonitor Windows Agent 475
Agent/Metric not showing up in the Control Panel 481
Automate the FortiMonitor Agent installation 482
Use the FortiMonitor Agent manifest file 482
Remove an instance automatically (FortiMonitor Agent) 487
Custom metrics and incidents 487
Native agent plugin 488
Command line calls 488
JSON-based custom metrics 488
JSON-based custom incidents 488

API 488
Custom Metric Management 488
Custom metric management 488
Custom metrics using the command line (FortiMonitor Agent) 490
Custom plugins for Linux (FortiMonitor Agent) 491
Custom plugins for Windows (FortiMonitor Agent) 493
Create custom incidents 496
Custom incidents using JSON files (FortiMonitor Agent) 498
Custom metrics using JSON files 500
Application plugins 502
Active Directory 502
Apache Kafka (Linux) 503
Apache Webserver 505
Apache Zookeeper 507
ASP.NET 508
Bandwidth (Linux) 509
Bandwidth (Windows) 509
Cassandra 510
CouchDB 511
CPU (Linux) 513
CPU (Windows) 514
Disk (Linux) 515
Disk (Windows) 517
Elasticsearch 518
Exim 519
File Information (Linux) 519
File Information (Windows) 520
HAProxy 520
Microsoft Exchange 523
Microsoft Hyper-V 523
I/O 524
IIS 524
JBoss 525
JMX (Linux) 527
JMX (Windows) 528
Linux Log Monitoring 529
Memcache 531
Memory (Linux) 532
MongoDB 533
MSSQL 534
MySQL 537
Nagios 546
Nginx 550
NodeJS 552
NTP (Linux) 552
NTP (Windows) 553
Oracle WebLogic (Linux) 554
Oracle Weblogic (Windows) 558
Oracle (Linux) 561

Package Upgrades 563
Patches 564
PHP-FPM 564
PHP OPCache 565
PostgreSQL 567
Processes (Linux) 568
Processes (Windows) 569
Prometheus 569
RabbitMQ 573
Redis 575
Sendmail 576
Control Panel Configuration 576
Additional permissions 576
TCP 576
Tomcat (JMX) 577
Tomcat (Legacy) 583
Tomcat (Windows) 588
UDP 592
UWSGI 593
Varnish 594
Windows Log Monitor 595
Windows Service 597
SharePoint 598
Uptime 598
Integrate with Domino 599
Monitor Beanstalk with the FortiMonitor Agent 601
Endpoint Agent 602
Usage scenarios 602
Remote site availability and performance 602
Collected server information 603
Available metrics 603
Add a Windows Endpoint FortiMonitor Agent 603
Resource requirements 603
Add a new Windows Endpoint FortiMonitor Agent 604
Enable DEM on an existing Agent 604
MacOS Endpoint Agent 605
Supported MacOS versions 605
Add a MacOS Endpoint Agent 605
CounterMeasures 606
Uninstall the MacOS Endpoint Agent 606
CounterMeasures 606
A simple example 607
Next steps 607
Linux CounterMeasures (FortiMonitor Agent) 607
Enabling CounterMeasures 607
Configuring a CounterMeasure 608
Viewing CounterMeasure output 609

CounterMeasure States 610
Approval 610
Using CounterMeasures in Templates 610
Utilizing Remote Actions 610
Standard Linux CounterMeasure actions (FortiMonitor Agent) 611
Custom Linux CounterMeasure actions (FortiMonitor Agent) 612
Windows CounterMeasures (FortiMonitor Agent) 618
Enabling CounterMeasures 618
Configuring a CounterMeasure 620
Viewing CounterMeasure output 620
CounterMeasure States 621
Approval 621
Using CounterMeasures in Templates 622
Standard Windows CounterMeasure Actions 622
Custom Windows CounterMeasures actions (FortiMonitor Agent) 622
CounterMeasures Security 628
CounterMeasures Slack Integration 628
Setup 628
OnSight vCollector 628
System requirements 629
OS requirements 629
Resource requirements 629
Network connectivity requirements 630
Docker IP change requirements 630
Deployment methods 631
631
One-liner installer 631
Installation 631
OnSight virtual image installation 632
Network configuration 633
NTP server configuration 633
OnSight AMI installation 633
Resource requirements 633
OnSight installation 633
OnSight for Azure installation 636
Resource requirements 636
OnSight installation 637
Automate the FortiMonitor OnSight deployment 638
OnSight Manifest File Configuration 638
Disable auto-provisioning 639
Monitor a device that has not been discovered with OnSight 639
OnSight discovery 642
Add auto-discovery rules 642
Add discovered instances 643
View monitored instances 644
Upgrading the OnSight 644
Update history 645

2020.68 or newer OnSight versions 645
2020.51 to 2020.67 646
2020.50 and lower OnSight versions 647
Manage and configure your OnSight vCollector 647
Management and configuration commands 647
OnSight custom plugins 648
CounterMeasure action plugins 649
Custom metric plugins 649
Requirements 649
Directory Structure 649
Creating a custom CounterMeasure 650
Creating a custom metric plugin 652
Deployment 653
Authentication 654
Secret management framework 654
Secret Management for Hashicorp Vault 655
Fabric extension 659
SNMP extension 660
Local development environment 661
Custom metric caching 664
TemplatePlugin.py 665
Additional setup and helpful tips 667
Access the OnSight web console 667
Set up FortiMonitor OnSight in HA or for failover 667
Use OnSight as a proxy for the FortiMonitor Agent 668
Move instances from an OnSight to another 672
OnSight security 673
Set up SSH access to the OnSight virtual appliance 673
AMI OnSight vCollector rotation procedure 674
Rotating clustered OnSights 675
Third-party integration 677
Integration workflow 677
Prerequisites 677
Arista CloudVision integration 678
Integration installation 678
Control panel configuration 678
Additional integration commands 680
Alkira integration 680
Integration installation 680
Control panel configuration 680
Additional integration commands 681
SNMP 681
Create an SNMP credential 681
View the list of devices associated with an SNMP credential 683
Configure the network device 684
Monitor metric performance 685
SNMP vendor documentation 686

SNMP traps 687
SNMP polling 688
Configure SNMP alert thresholds 689
SNMP troubleshooting 689
Trap processing 692
Plugin installation 692
Creating a custom SNMP trap plugin 692
Alerting 693
Alert Timelines 693
Create an Alert Timeline 694
Configure who gets alerted during an outage 697
Change the amount of time between an outage happening and the first notification 700
Respond to an incident and delay further alerts 700
Prevent active outages from creating alerts 702
Simulate an outage 702
Disable Incident Resolved alerts 704
Email 705
SMS 707
Troubleshooting SMS alerts 707
SMS and voice numbers for alerts 708
Which wireless carriers are compatible with FortiMonitor alerts? 708
Integrations 708
ServiceNow 709
Slack 715
Microsoft Teams 716
HipChat 724
Campfire Chat 725
StatusPage.io 727
PagerDuty 729
OpsGenie 732
Webhooks 733
Jira and Jira Service Desk 737
Zendesk 740
Salesforce 743
Limit the number of confirmation checks for an outage 749
Set up an On-Call schedule 750
Create an On-Call schedule 750
Add On-Call events 750
Default On-Call contact 752
Using On-Call schedules 752
Notifications when rolling on and off 752
Set up iCal integration 753
Multiple server outages 754
Can I notify someone separately just for one service on a server? 754
Long-running incidents 755
Setting aside alerts 755
Parent-child alerting for Fabric and SNMP devices 756

Previous Behavior 756
Incident Hub 756
Filter incidents 757
View incident details 758
Acknowledge an incident 758
Set an Incident Lead 759
Start a maintenance 760
View Incident Solutions 760
Switch between Master tenant and Cross-tenant views 761
Perform other actions 761
View incident details 762
Incident Header overview 762
Set Incident Lead 763
Acknowledge an incident 763
Start a maintenance 764
Add incident tags 764
Additional actions 765
Event panel 765
Timeline and Messages 766
Performance 768
Incident History 769
Metric History 770
Incident Solutions 770
Apply a solution to an incident 772
Manage Incident Solutions 777
Create an incident using email 779
Create a custom incident 779
Create an email gateway 779
Email content options 780
View email history 783
Reports 783
Available reports 784
Agent Status Report 784
Availability Report 785
Capacity Planning Report 786
Incident Report 787
Incident Metric Export Report 788
Tag Audit Report 788
Tags & Attributes Report 789
User Audit Report 790
Instance Configuration Report 791
Interface Utilization Report 792
Metric Export Report 793
Metric Configuration Report 794
SD-WAN Report 794
Service Performance Report 795

MTTR and MTTA Report 796
Historical SLA Performance Report 797
User Incident Activity Report 798
Create a report 798
Public status pages 799
Customizing Status Pages 802
Exporting Metric Reports for long duration time periods 802
Maintenance Schedules 804
Creating custom Maintenance Schedules 805
Maintenance Schedules page 805
Configure a Maintenance period 805
Immediate Maintenance 807
Extend a Maintenance Schedule 808
End an existing maintenance 808
Maintenance history 809
API 809
API keys 810
View the API library 811
Custom incidents using API 814
Add a custom incident filter 814
Create a custom incident 814
Resolve an incident 815
HTTP status code summary 816
Using Postman with FortiMonitor's API 816
Authenticating with Postman 817
Using a GET request 817
Filtering servers by FQDN 818
Using a POST request 819
Using a PUT request 820
Using a DELETE request 821
Multi-tenancy 821
Multi-tenant account requirements 822
Key benefits 822
White labeling and reselling 822
Global infrastructure management 822
Manage subtenants 822
Use a single SSO configuration 822
Switch between subtenants 823
Add a user with access to all subtenants 824
Add users to subtenants 824
Create a subtenant 825
Manage a multi-tenant environment 826
Infrastructure map 826
Dashboards 826
Share an OnSight vCollector 826
Monitoring policies 827

Templates 828
Search 828
Multi-tenant dashboards 828
White label options 829
Primary requirements 829
Login and domain options 830
SSL options 830
Email options 830
Logo and favicon options 830
Custom CSS and JS 831
Mobile app 831
Install and configure the mobile app 832
Install from the Apple App Store and Google Play 832
Configure push notifications 832
Mobile configuration 832
Control panel configuration 832
View Active Incidents 834
View incident details 835
Acknowledging incidents and other actions 836
Acknowledge an incident 836
Schedule Maintenance 836
Delay Alerts 837
Cancel Alerts 838
Escalate Alerts 838
NCM 838
NCM installation and initial configuration 838
Prerequisites 838
Install the NCM 839
Create a new credential 841
Discovery 843
Backup 843
Uninstall the NCM 844
NCM Supported Devices 844
Backup 854
Basics 854
Automatic exec-mode switching 854
Startup vs. running config 854
Only unique backups (when config changes) are stored 854
Backup filters 855
Search syntax help 855
Regex search examples 856
Regex cheat-sheet 856
Mass Config Push 856
Basics 856
Output Grouping 857

State of device before Push - require "enable" or "configure" CLI mode 857
Command parsing 858
Command sending modifiers and control sequences 858
How commands are sent to devices 859
Command parsing examples 860
Debugging failed pushes - "INTERACTION_ERROR" group 861
Device mode table 861
VT100 Control / Command sequences 862
Device variables 863
Defining device variables 863
Single device variable edit 863
Bulk (multi-device) variable edit 863
Variable formatting 864
Using variables in Config Push 864
Variable substitution and the "\" escape character 864
Variable chaining and Push Behavior Modifiers 865
NCM Slack configuration 865
System login 870
Access accounting 870
Discovery 871
What does discovery do? 871
When is Discovery performed? 872
How to find out why device didn't discover? 872
Scheduling 873
The Default Schedule 873
Explicit schedules 873
User security and access roles 873
Account security roles 873
Device access restrictions 875
Accounts with restricted access 876
Sensitive data stripping 876
Always check for desired behavior 877
Supported devices 877
Integrations 878
Alerting 878
Third-Party Integrations 878
Cloud Monitoring 878
Security Fabric 879
SNMP MIBs 879
FAQ 881
What do you count as a Service? 881
Check Frequency 881
How do you minimize false alerts? 881
What are you checking when you check my servers? 882
From where is my site monitored? 882

Can you monitor my custom application? 882
Does FortiMonitor do SaaS monitoring? 882
Can FortiMonitor collect information needed to check an SLA? 883
I have servers behind a firewall, can I monitor those? 883
Does FortiMonitor offer a partnership program? 883
Do I need the Server Agent and the OnSight vCollector? 883
Data Retention and Compliance 884
FortiMonitor compliance 884
Encryption 884
Data retention 884
Data deletion requests 884
Get support 884

Getting started

Welcome to the FortiMonitor documentation site. You can find some of the most helpful pages for getting started below. If you cannot find a solution, feel free to contact your account manager or our support team.

- Part 1: Add your infrastructure to FortiMonitor
- Part 2: Monitoring
- Part 3: Alert Timelines
- Part 4: Visualization
- Part 5: Team Management
- Part 6: Reports
- Part 7: CounterMeasures

Monitoring Approach

FortiMonitor allows you to monitor your public and private infrastructure using 3 different dimensions of monitoring. Using all 3 dimensions together allows you to correlate different types of events/incidents across your infrastructure.

1. Monitor publicly accessible servers and services using our globally distributed monitoring probe network. Test for uptime, performance, and synthetic transactions from any of our 50+ PoPs.
2. Use the FortiMonitor OnSight vCollector to achieve the same in-depth monitoring and secure reach servers and devices from behind your firewall.
3. Install our Server Agent to collect key operating system health and application performance metrics and be alerted when different thresholds are breached.

Part 1: Add your infrastructure to FortiMonitor

This article describes the methods you can use to add your infrastructure to FortiMonitor. Each method varies in effort, requirements, and level of automation. Review the different deployment methods listed in this article to determine which

option works best for your environment.

Refer to the following table to see the features and requirements of each option.

Method	Features					Requirements	
	External monitoring	Server metrics	SNMP	CounterMeasures	Auto-discover instances	Instance access	Network access
FortiMonitor Agent	✓	✓		✓		✓	
OnSight Discovery	✓		✓		✓		✓
FortiMonitor Cloud Discovery	✓				✓		
OnSight vCollector installer	✓		✓		✓	✓	✓
Add a basic instance	✓		✓				
Add via API	✓		✓				

For example, if you want to use CounterMeasures, add your infrastructure using the FortiMonitor Agent method. Otherwise, if you only want to monitor SNMP-related metrics, add your infrastructure manually or use OnSight Discovery.

FortiMonitor Agent

The FortiMonitor Agent is a local monitoring utility that is deployed directly on your physical server, VM, or cloud server. It allows you to monitor your server's resources, processes, and services. The FortiMonitor Agent also supports several plugins to monitor applications like web servers, databases, caches, and so forth. It is also extensible, allowing you to customize it to meet your own needs.

The FortiMonitor Agent can be used to monitor applications running on your servers such as Apache, Cassandra, SQL Server, and more. The Agent is compatible with Windows, Linux, and Unix.

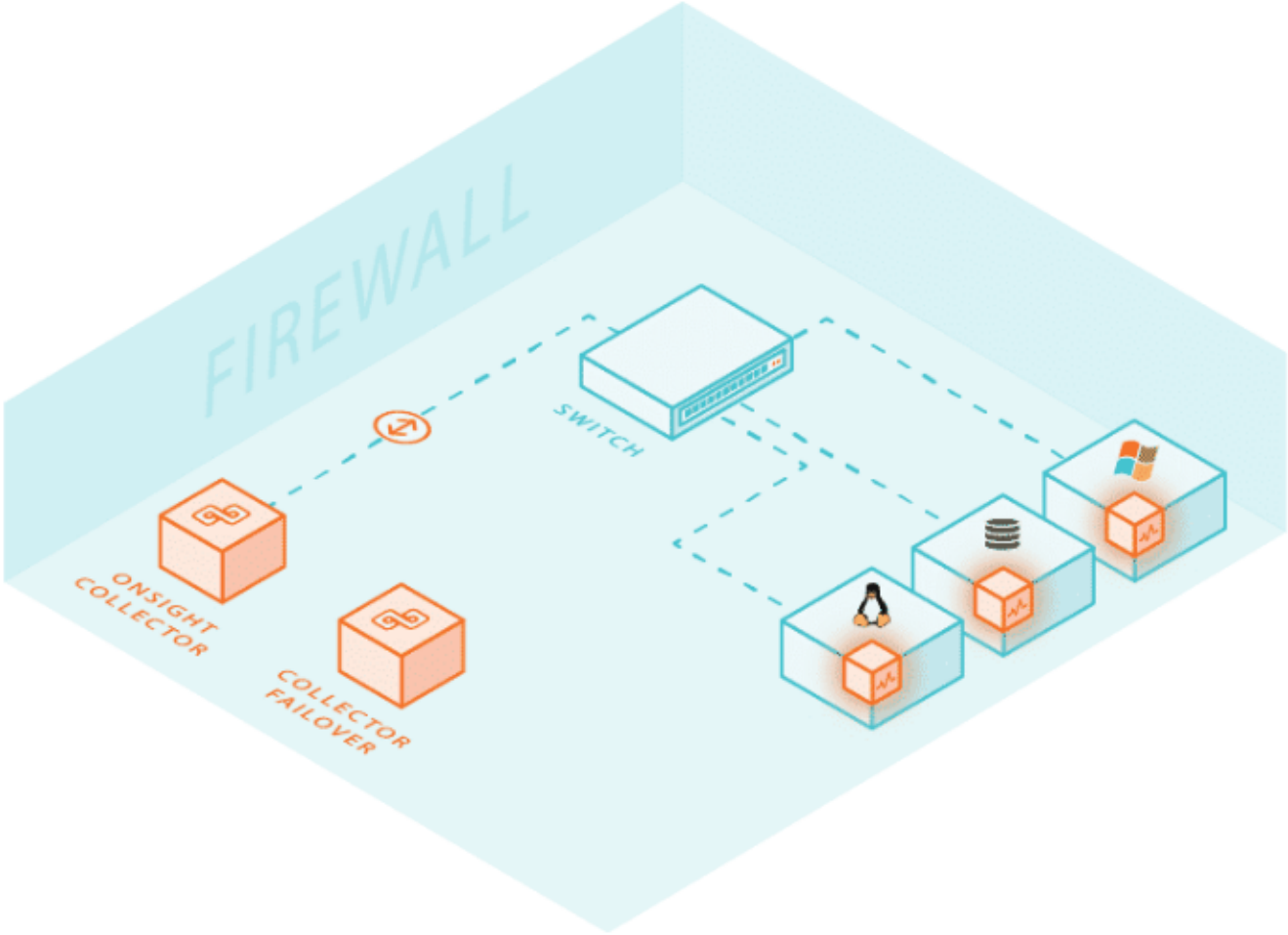
Click on the following links to learn more about the FortiMonitor Agent:

- FortiMonitor Agent
- Install the FortiMonitor Agent
- Automate the FortiMonitor Agent installation

OnSight vCollector installer

The Onsight vCollector sits behind your firewall or within your virtual private cloud. It can perform network checks, SNMP polling, and serve as a proxy for FortiMonitor Agents. It can also be used to provide deeper monitoring on network devices such as switches, routers, and firewalls.

The Onsight vCollector can be installed on a Linux VM, as an AMI in an AWS environment, as a Virtual Machine in Azure, or on any other hypervisor, you are using. Docker images are also available for container-based environments.



Once deployed, an OnSight vCollector can scan your network or private cloud for instances and devices using a specified CIDR range. The discovered instances and devices can be added automatically or manually. Instances and devices discovered by the OnSight can be found under the Discovered Instances tab of your OnSight's details page.

OnSight Automation

You can deploy an OnSight vCollector instance to a Linux VM by following the instructions provided in the OnSight standalone installer. You can also automate the deployment of OnSight vCollectors by following the instructions provided in Automate the FortiMonitor Agent installation.

The following list provides the minimum Linux VM requirements for a standalone Onsight vCollector deployment:

As your monitoring expands, we recommend 1 core + 1 GB Ram per 1,000 checks.

AWS AMI

If you are deploying the OnSight vCollector as an AMI in your AWS environment, follow the procedures in OnSight AMI installation.
The following list provides the minimum requirements for an AWS AMI OnSight vCollector deployment:

- Minimum of a t3.medium instance - as your monitoring expands, we recommend 1 core + 1 GB Ram per 1,000 checks
- See Network connectivity requirements.

Virtual image

A virtual image of the OnSight vCollector is available if you want to deploy it to your hypervisor. For a detailed procedure, see OnSight virtual image installation.

OnSight Discovery

Once you've installed the OnSight vCollector, you can use it to discover your infrastructure. The OnSight discovery tool allows you to scan your infrastructure and automatically add devices to your FortiMonitor account. After installing the OnSight, you can customize how the tool searches for new devices. You can define an IP range, the re-scan frequency, and the scan type to determine precisely what devices should be added. Servers and devices discovered by OnSight can be found under the Discovered Instances tab on the OnSight page.

Click on the following link to learn more about the OnSight Discovery tool:

- OnSight discovery

FortiMonitor Cloud Discovery

OnSight Cloud Discovery allows you to scan your infrastructure from different cloud platforms (AWS or Azure). Data gathered from these cloud platforms should be used as an augmentation of, not a replacement for, the data obtained by the FortiMonitor server agent and external monitoring. The server agent can provide more detailed and accurate data across any OS distribution or application you may be running on your compute instance. Also, FortiMonitor's external monitoring ensures you're getting the full picture of your current operating environment as well as a view into what your customers are experiencing.

- To grant FortiMonitor access to Cloudwatch, see AWS monitoring
- To grant FortiMonitor access to Azure, see Azure monitoring

Add a basic instance

Manually adding an instance allows you to monitor publicly available websites, endpoints, servers, services, and more via FQDN or IP. To add a basic instance, perform the following steps:

1. Ensure you are logged in to the FortiMonitor control panel.
2. From the navigation menu, click **Add**.
3. In the modal that appears, locate the **Website/Endpoint** card then click **Select**.

You are now in the add instance flow. From here, you can configure details about your instance and how you'd like to monitor it.

If you want to monitor instances that are not publicly accessible, add your instances using the Onsight vCollector method.

Add a network device (Basic)

You can add a network device for basic accessibility monitoring using an OnSight vCollector.

1. Ensure you are logged in to the FortiMonitor control panel.
2. From the navigation menu, click **Add**.
3. Locate the **Network Device (basic)** card from the **Infrastructure and Resource Catalog** then click **Select**.

You are now in the add instance flow. From here, you can configure details about your instance and how you'd like to monitor it.

Add via API

You can add your infrastructure to FortiMonitor via FortiMonitor's API. For more information, see the following topics:

- API
- API keys
- View the API library

Part 2: Monitoring

This article describes the different methods you can use to apply monitoring to your servers in FortiMonitor. Review each method in the following list to determine the best option for your environment.

Monitoring types

Application monitoring - Monitor system and application metrics.

External monitoring - Monitor instances from public checker nodes, such as HTTP, Ping, Multi-step checks, etc. The OnSight vCollector can also perform the same checks as FortiMonitor's public nodes, but in this case, for instances in your private network.

Network device monitoring (SNMP) - Collect SNMP metrics for network devices and other hardware that use the SNMP protocol.

Basic methods

Apply monitoring manually

This method allows you to apply monitoring to an existing server or instance. This method is best used if you only want to apply monitoring to a few existing instances. After selecting the services or application that you want to be monitored,

you can configure the metrics on the following screen based on your requirements.

For more information, see Add monitoring to an existing server.

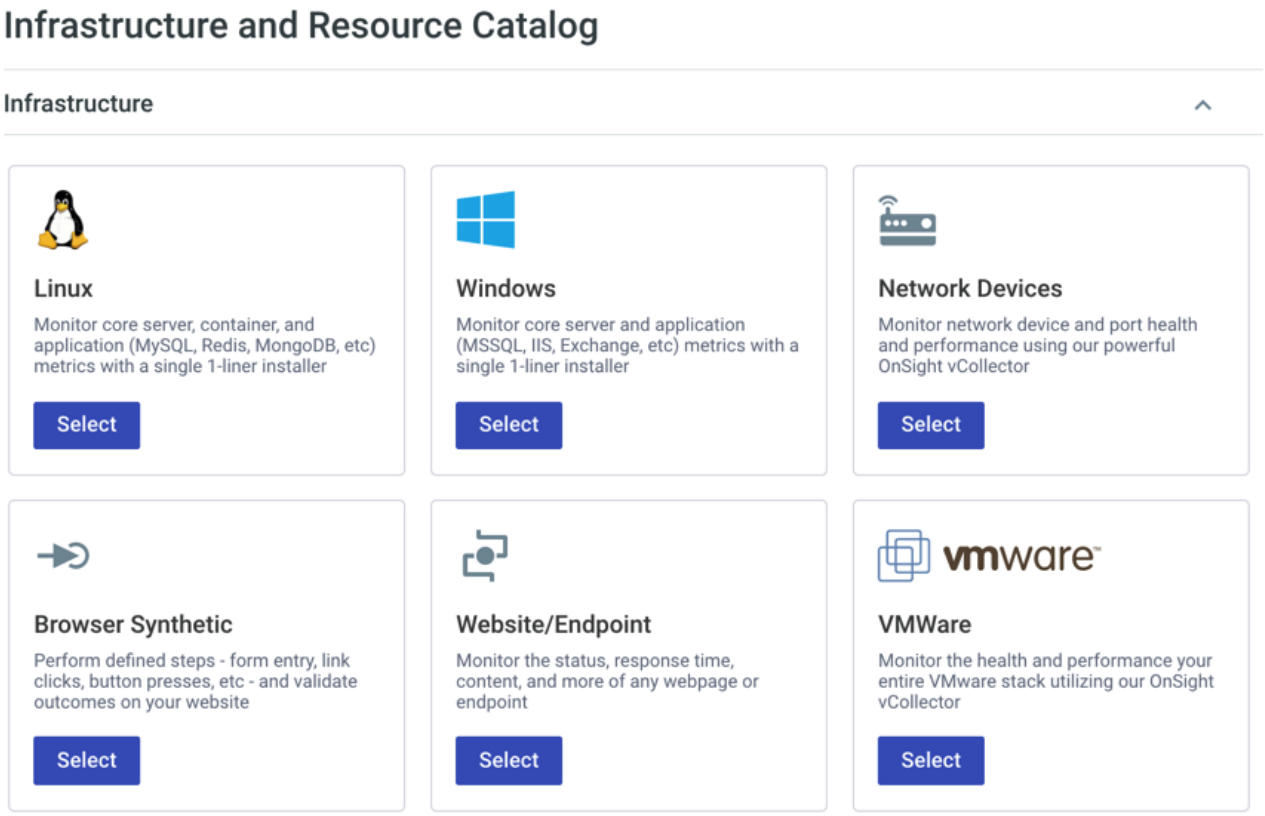
You can also apply a template to any instance that you create manually. One of the steps is to choose a template to apply to the new instance. Any configuration on the template will be applied to the server upon the creation of the instance.

Apply monitoring during the installation of FortiMonitor Agent

Installing the FortiMonitor Agent on a server will automatically add the server to your FortiMonitor account. You have the option to add network checks (HTTP/HTTPs/Ping, etc.) and Agent metrics (CPU, Memory, Disk) to your servers during the installation of the FortiMonitor Agent.

To install the FortiMonitor Agent to a server:

1. Click **Add** from the navigation menu.
2. From the **Infrastructure** section of the catalog, you can choose where to install the Agent.

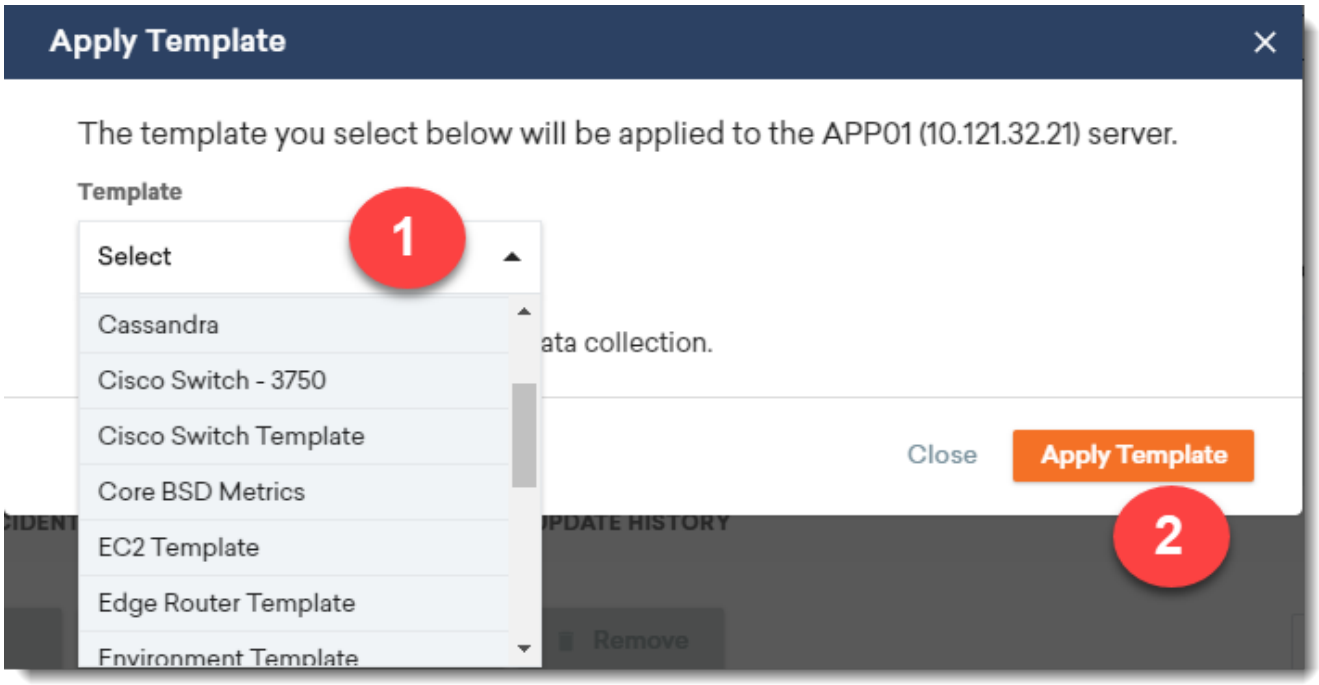


See Install the FortiMonitor Agent for more information on how to install the FortiMonitor Agent and apply monitoring.

Apply templates manually (API or UI)

You can use this method If you have a large infrastructure and want to apply the same monitoring to a group of related servers or instances. You can apply a template manually from the **Instance Details** page by selecting **Apply Template**.

Select the template that you want to apply to the server then click **Apply Template**.



Any monitoring or metrics configuration on the template will be applied to the selected server. For a detailed procedure on how to apply server templates, see Apply server templates.

Create server templates

You can also build a template from scratch or clone an existing server into a reusable template. If you're backing into using templates after your initial setup and already have monitoring in place, the templates will automatically match metrics which are of the same type to avoid any duplication. FortiMonitor also allows you to create your own template. You can create templates by following the steps detailed in Templates.

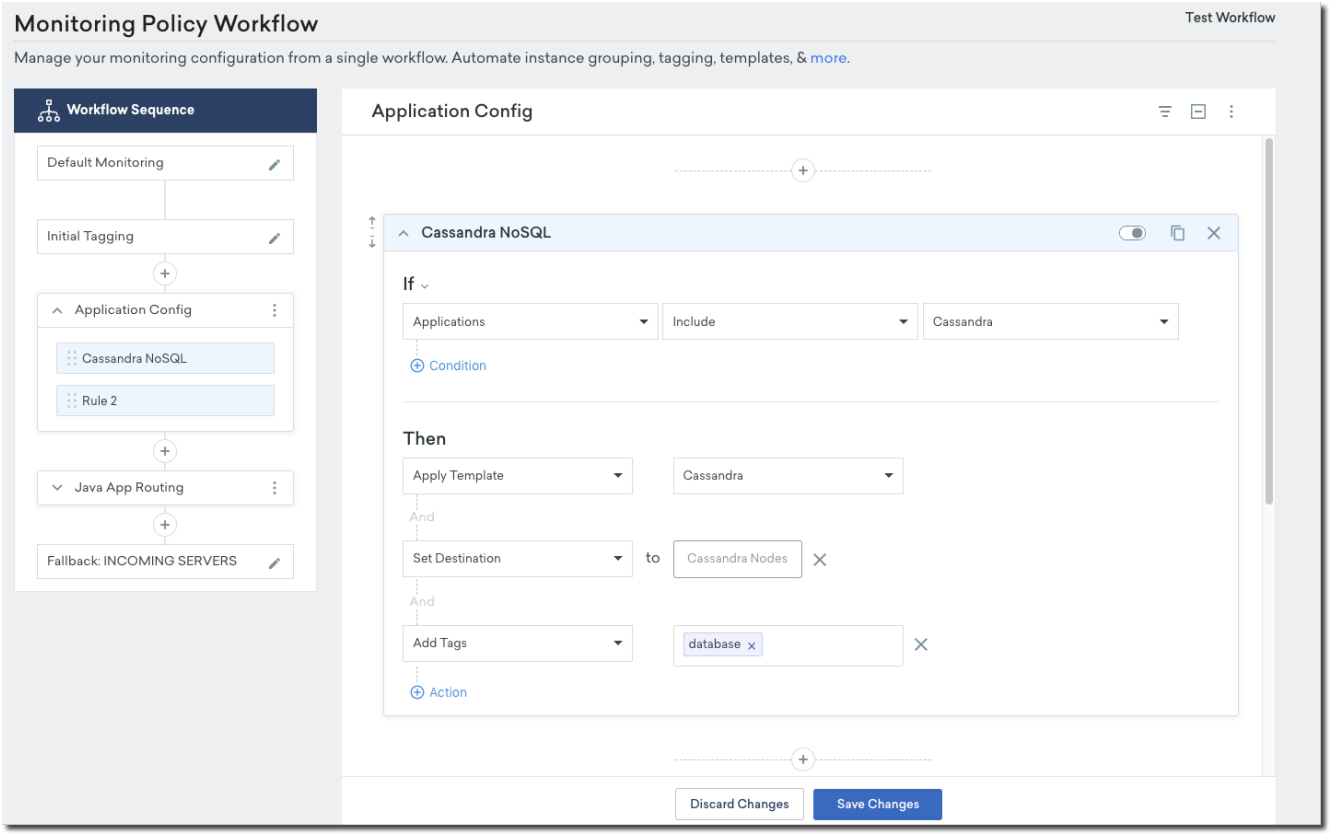
Templates can also be applied manually by utilizing FortiMonitor's API. For more information, see API keys.

Advanced methods

Monitoring policy framework

In large-scale deployments, applying monitoring to servers manually can be time-consuming and may be prone to errors. Monitoring policies allow you to manage your monitoring configuration from a central location using a workflow-like interface. This method provides a level of automation that can both save time and avoid errors due to human intervention.

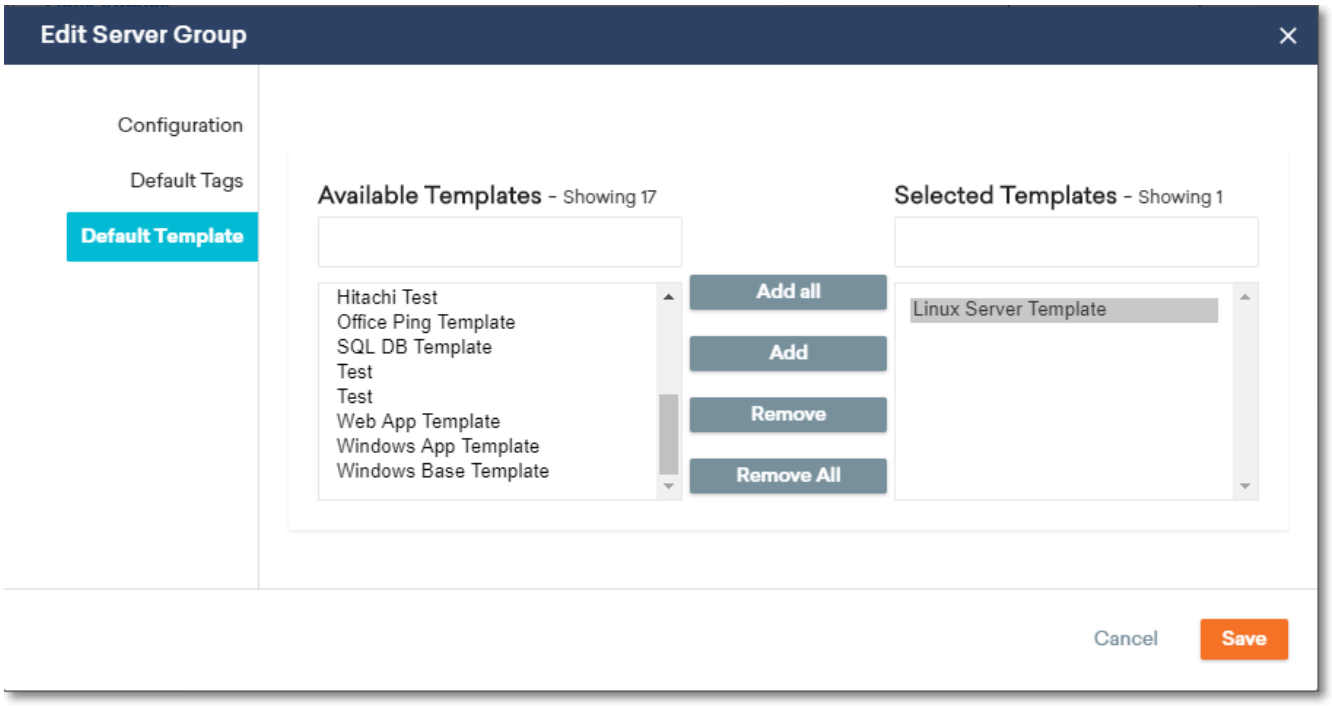
The Monitoring Policies Workflow is a series of transformations (policies) that are applied to your instances as they are added to FortiMonitor. The policies can and usually include a condition - e.g. if Apache is on the instance - and an outcome - apply a template, add a tag, etc. The policy conditions can be either a collection of OR or AND predicate, forming a basic IF...THEN statement. The policies are executed sequentially and will override any settings that were previously set. FortiMonitor also provides a few default policies that you can utilize or use as a basis for your own custom policies.



For more information, see Monitoring policies.

Apply default templates to a group

You can apply a default group template to a group of servers by selecting the group from the **Instances** drawer then clicking **Edit** on the **Instance** page. You can now edit the group to apply a default template. For example, if you want to apply monitoring to a group of Linux servers, you can choose the **Linux server** template and all configured monitoring and metrics will be applied to the group of servers. Any servers added to the group will automatically have this template applied to those servers as well.



For more information, see Use default group templates.

Use the FortiMonitor Agent manifest file

If you have a large infrastructure, using the **FortiMonitor Agent manifest file** allows you to streamline the configuration of your servers in FortiMonitor. You can use the manifest file to set configuration values and have the FortiMonitor Agent add the server with its preconfigured values to FortiMonitor. For more information on how to use the FortiMonitor Agent manifest file, see Use the FortiMonitor Agent manifest file.

You need to install the FortiMonitor Agent to use this method.

Example manifest file

```
[agent]
customer_key = afsdyngoaeppmfqefa
server_key = ashe-pokf-bfhb-eabn
aggregator_url = myappliance.localsite.com
server_group = 3467
fqdn = www.panopta.com
server_name = Panopta
interface_mapping = private:10.100.100.2,private2:10.100.100.13
templates = 8
tags = tag, anothertag, anotherone
partner_server_ID = 828765
```

```
disable_server_match = true
custom_plugin_url = https://s3.amazonaws.com/custom-panopta-plugins/my-custom-plugins.zip
enable_countermeasures = true
countermeasures_remote_plugins = https://s3.amazonaws.com/some-s3-bucket/custom-plugins.zip
countermeasures_refresh_plugins = 6
```

See Parameters for a detailed description of each parameter.

Part 3: Alert Timelines

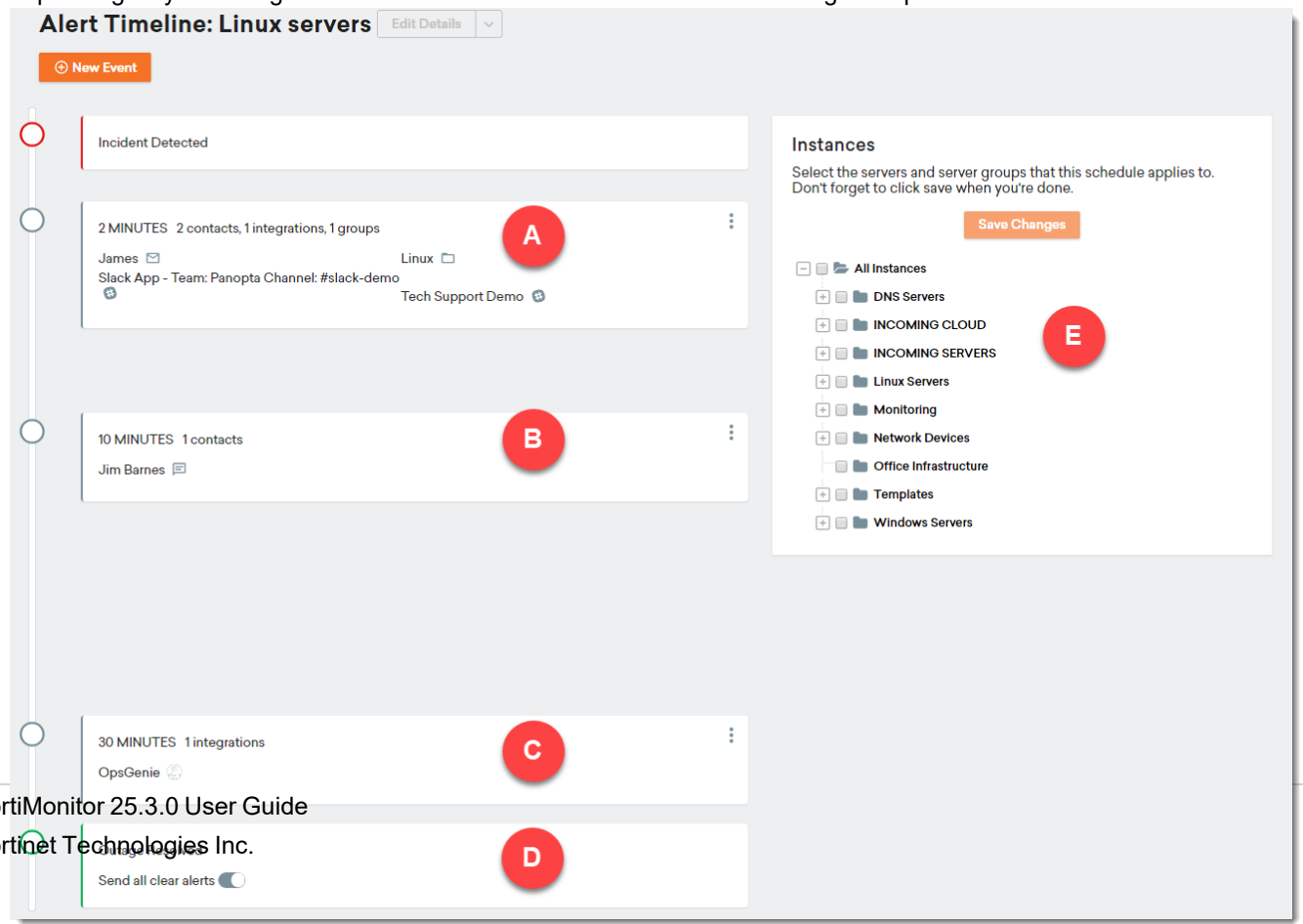
All incident notifications in FortiMonitor are managed in an **Alert Timeline**. An Alert Timeline is a highly configurable workflow for managing alerts when an incident is detected. It pivots off of a number of variables that you can utilize to best suit your team's alerting needs by involving multiple people, teams, and integrations.

Alert Timelines

With Alert Timelines, you can configure the following:

- Who gets alerted
- Which notification medium to use (direct or via third-party integration)
- The amount of time between the identification of an incident and the first and succeeding notifications

Once an incident is detected or a certain threshold that you have configured is breached, alerts will be sent out depending on your configuration. You can see how this works in the following example:



A	**First event**
	Users and groups are notified depending on the configured contact method, On-Call schedule, or third-party integration. At this point, any person in your team can acknowledge it and take responsibility, escalate it to the next level, or delay or cancel all future alerts.
B	**Second event**
	In this example, 10 minutes have passed without any resolution and an SMS is sent to the specified user.
C	**Third event**
	After 30 minutes, the alert has escalated further and has been reported to a third-party integration (OpsGenie). **Note**: You can build as many events as needed.
D	**Outage resolved**
	When the incident is resolved, you have the option to send an "all clear" message. Any previously notified users, groups, or integrations will be notified on the communication channels they were previously notified on.
E	**Instance tree**
	This Alert Timeline is applied to your selected instances and groups

You can set a specific alert timeline on the metric, instance, or group level. All of this can also be controlled by Templates.

For more information, including instructions on how to create an Alert Timeline, see Alert Timelines.

Who gets alerted

When you create or edit a new event in the Alert Timeline, you can select the users, groups, and/or integrations that you want to be notified during an incident. Note that the users and groups, including their contact methods, must first be configured in the **Users, Groups, and On-Call** page.

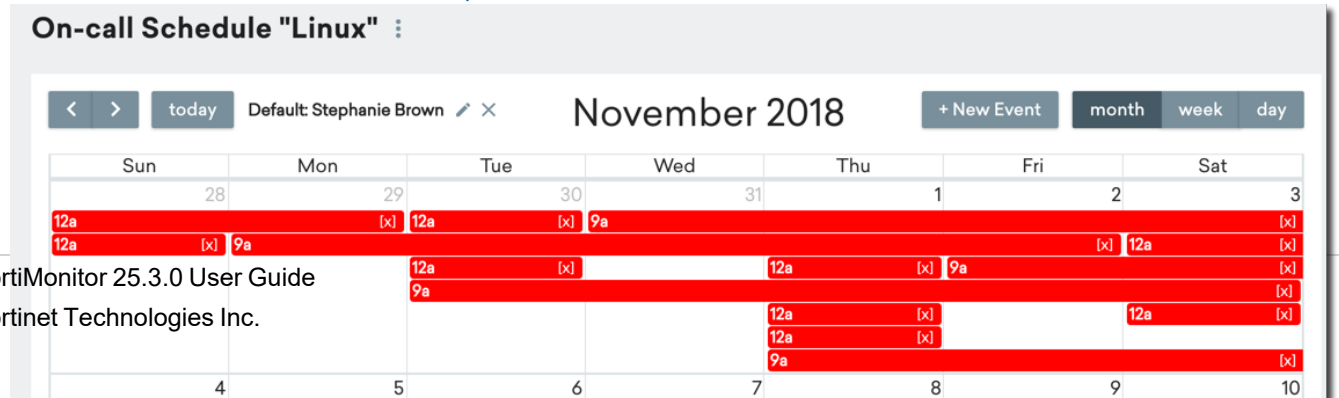
To configure who gets alerted during an incident, see Configure who gets alerted during an outage.

To add users or groups to FortiMonitor, see Add users to your account.

On-Call schedule

Similar to adding users, groups, and integrations, you can also add **On-Call schedules** to the Alert Timeline. On-Call schedules allow you to set dynamic alerting behavior based on the specific on-call rotations within your organization. Schedules can be set to any level of granularity.

To create an On-Call schedule, see Set up an On-Call schedule.



Channels and integrations

You can configure Alert Timelines to send notifications to users or groups through several direct and third-party channels.

Notifications can be sent using the following direct channels:

- Email
- SMS
- Voice

The following third-party integrations are supported:

- ServiceNow
- Slack
- Microsoft Teams
- HipChat
- StatusPage.io
- PagerDuty
- OpsGenie
- WebHooks

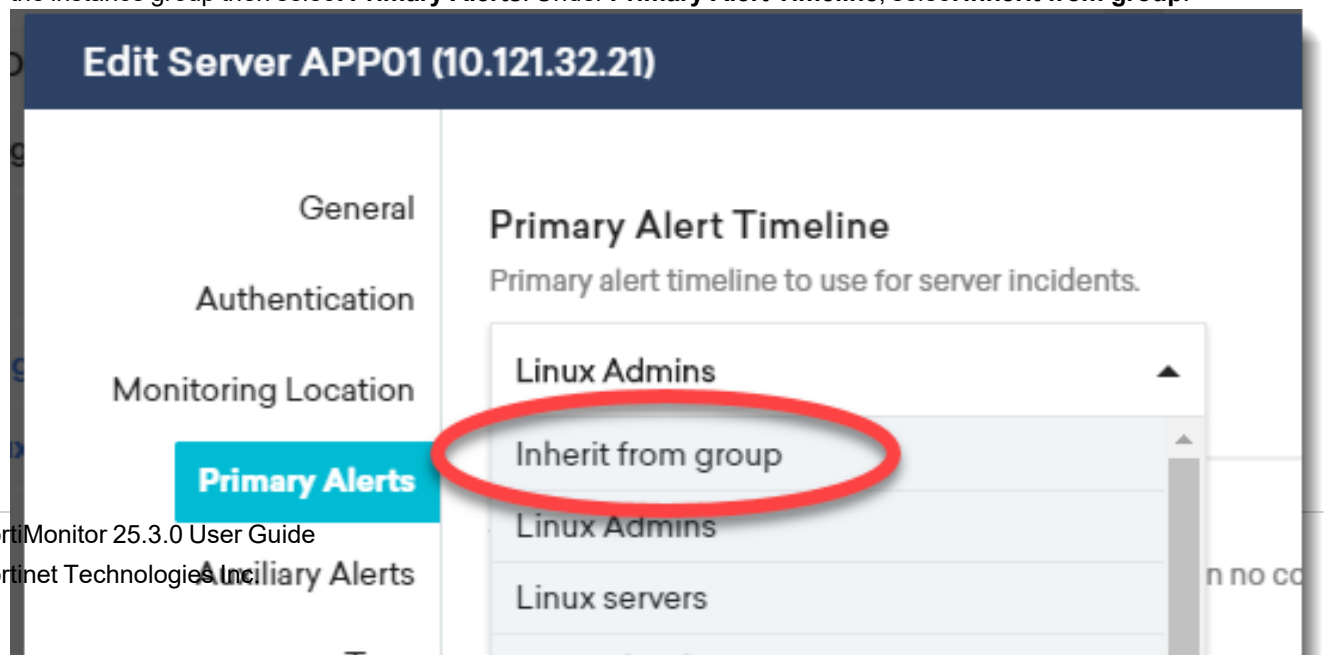
See Integrations for details on how to configure each integration.

Using Alert Timelines

Alert Timelines can be applied to an instance, an instance group, or a metric. Metric-level Alert Timelines can override both the Group-level and instance-level timelines to add granularity to how FortiMonitor sends notifications for incidents.

Apply an Alert Timeline to an instance group

When you apply an Alert Timeline to an instance group, the Alert Timeline can be assigned to all instances in that group, and notification will be sent to the Alert Timeline's configured contacts if there is an incident in one or more instance in that group. This method is especially useful if you have a large number of instances that are logically grouped. For example, you can create an Alert Timeline for a group of database servers to notify database admins during an incident. The Alert Timeline is centrally managed in the database instance group. You can configure each instance to inherit the instance group's Alert Timeline. To configure an instance to inherit an instance group's alert timeline, select **Edit** from the instance group then select **Primary Alerts**. Under **Primary Alert Timeline**, select **Inherit from group**.

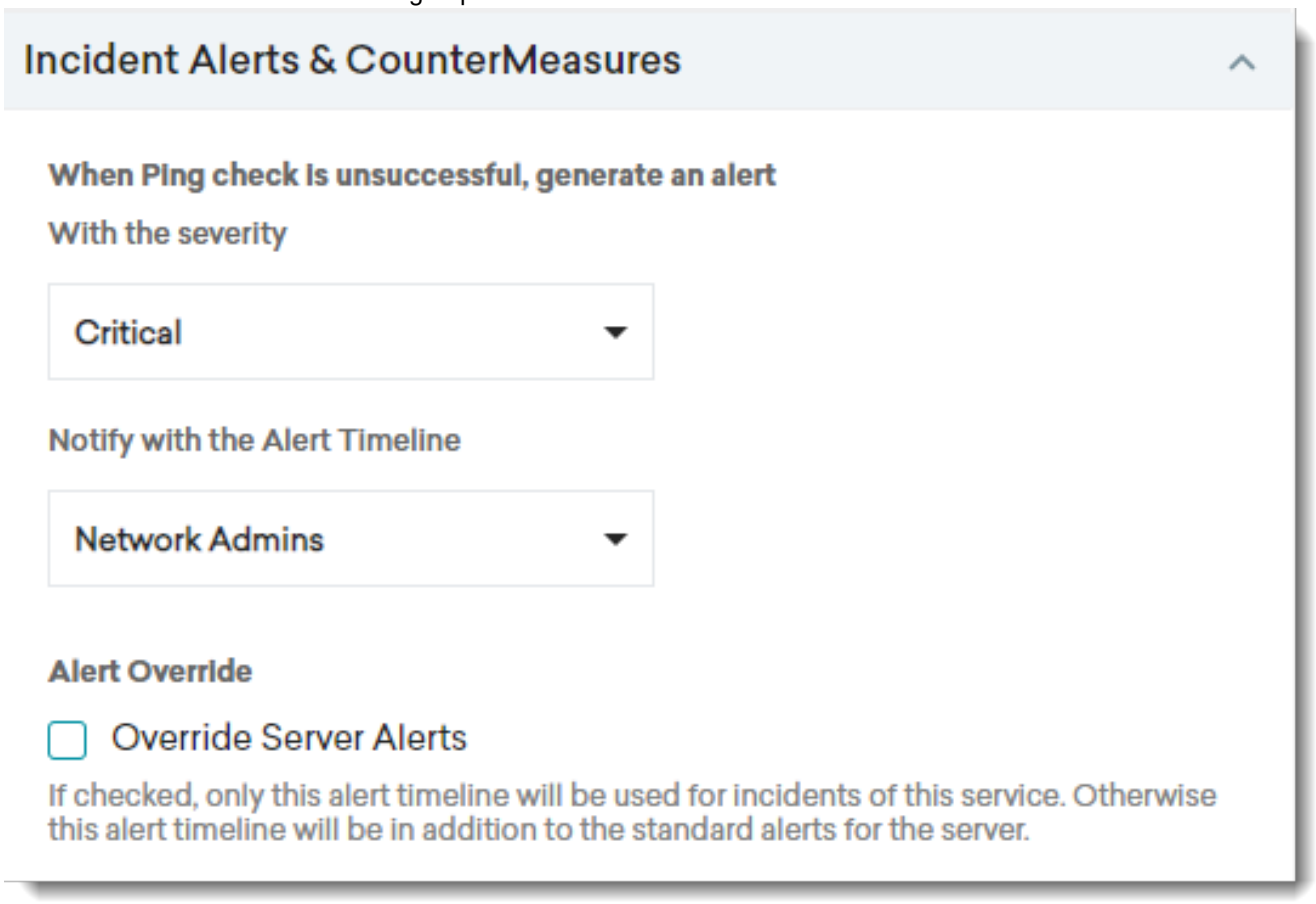


Apply an Alert Timeline to an instance

You can also apply an Alert Timeline to individual instances. This can be useful if you want to apply an Alert Timeline to an instance in addition to the Alert Timeline assigned to the instance's group. For example, you have an Alert Timeline assigned to a group of database servers but you want to notify supplemental team members to database servers that are critical to your operations. You can apply instance-level timelines to those database servers.

Override an Alert Timeline

In addition to the Alert Timeline assigned to an instance or an instance group, you can configure Alert Timelines on the metric-level. The metric-level timelines can be used to override the notification function configured in the instance's or instance group's Alert Timeline. It can also be used to supplement the normal notification function. This is useful if you want to notify team members who are responsible for only a specific metric. For example, you can configure the Ping check metric to only notify the Network Admins group and override the settings that you have configured in the Alert Timeline for the instance or instance group.



Default Alert Timelines

A Basic Alert Timeline is created when you first sign up to FortiMonitor. If an instance is added without specifying which Alert Timeline to use, the **Basic Alert Timeline** is used. If that has been removed, the first Alert Timeline in your account is used.

Maintenance period

In FortiMonitor, you can set up a maintenance period anytime you do routine maintenance, upgrades, or other tasks that may cause your instance or applications to trigger incidents. Adding a maintenance period to an instance or instance group can override the Alert Timeline so that FortiMonitor can monitor these instances depending on the maintenance period's configuration. See the following table for more options on how to handle monitoring and notifications for instances under a maintenance period.

Option	Description
Pause monitoring	All monitoring for the instances covered by the Maintenance Period will be paused - meaning no incidents will be created and no alerts will be sent
Continue monitoring but don't create incidents	All monitoring for the instances covered by the Maintenance Period will continue - however, incidents will not be created, regardless of the monitoring data collected
Continue monitoring & creating incidents but don't send alerts	All monitoring for the instances covered by the Maintenance Period will continue and incidents will be created accordingly - however, no alerts will be sent for any detected incidents
Continue monitoring, creating incidents and sending alerts	Things will operate as normal - however, an Alert Timeline override must be selected to be used for alerting

In addition to overriding the Alert Timeline's alerting and escalation configuration as shown in the table above, you can also control the general handling of the maintenance period, the timing, and the scope of your infrastructure which it covers. There are many ways to select which instances will be part of a maintenance period. You can select by instance group, individual instance, or by tag. See Maintenance Schedules for more information.

Acknowledging an incident

Acknowledging an incident marks the incident with your username and can optionally add a message to the incident log. You also have the option to send a message to all configured users and channels in the Alert Timeline or delay further alerts for a set amount of time. There are several ways to acknowledge an incident:

- See **Incident Hub** and View incident details**.**
- From the **Infrastructure Map**, select the instance that has an incident then click **Take Action** > **Acknowledge**.
- From the **Instance Details** page, click the incident from the incident header. This will take you to the instance's incident page. You can acknowledge the incident from there. See View incident details.
- From the navigation header, select the **Notification** icon. Select the incident then click **Acknowledge**.
- From the email you receive during an incident, select the **Acknowledge** link. This will take you to the control panel.

In addition to acknowledging an incident, the following actions are also available depending on which page you are on:

- **Broadcast** - Send a message to all configured users and channels in the Alert Timeline
- **Exclude from Availability** - Exclude the instance from the availability calculation
- **Delay alerts by** - When you acknowledge an incident, you have the option to postpone alerts for a set amount of time
- **Enable all clear** - this functionality alerts all previously alerted users/integrations that the incident has resolved
- **Cancel alerts** - cancels succeeding alerts. No further alerts will be sent

- **Escalate to the next event** - Immediately send a notification based on the next event, overriding the configured amount of time in the Alert Timeline. This means that even if you have configured the Alert Timeline to send a notification after 10 minutes, you can override this and send the notification earlier.

Once an incident is resolved, the Alert Timeline will stop further escalations and send an "all clear" message to all the users, groups, and integrations defined in the Alert Timeline.

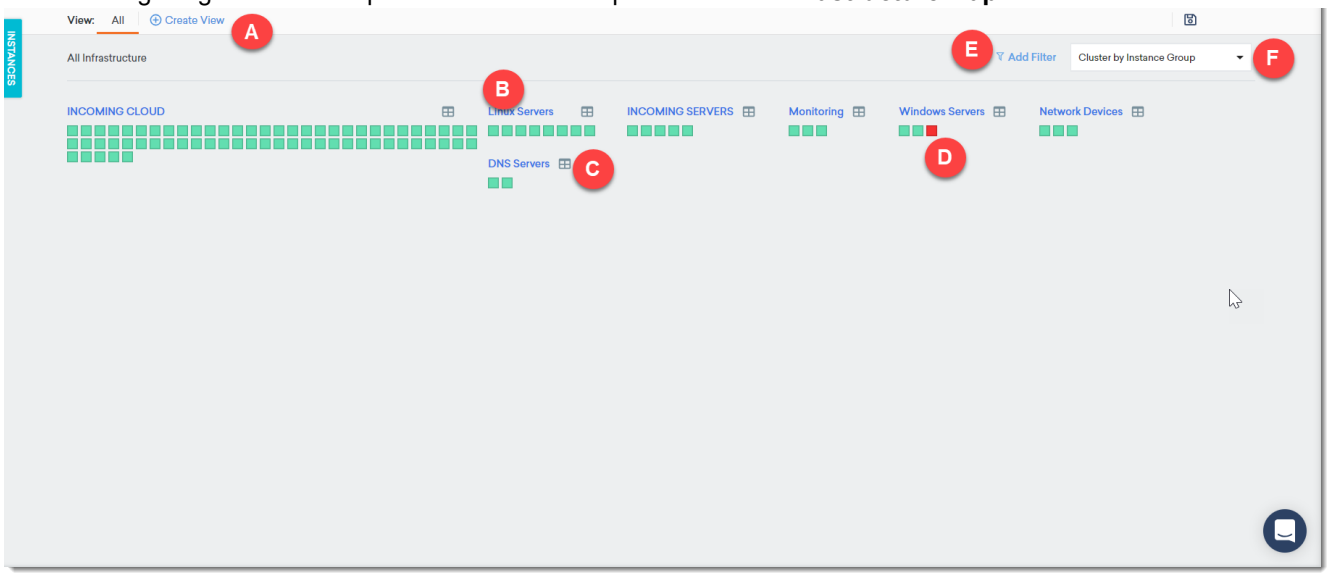
For more information on responding to an incident, see Respond to an outage and delay further notifications.

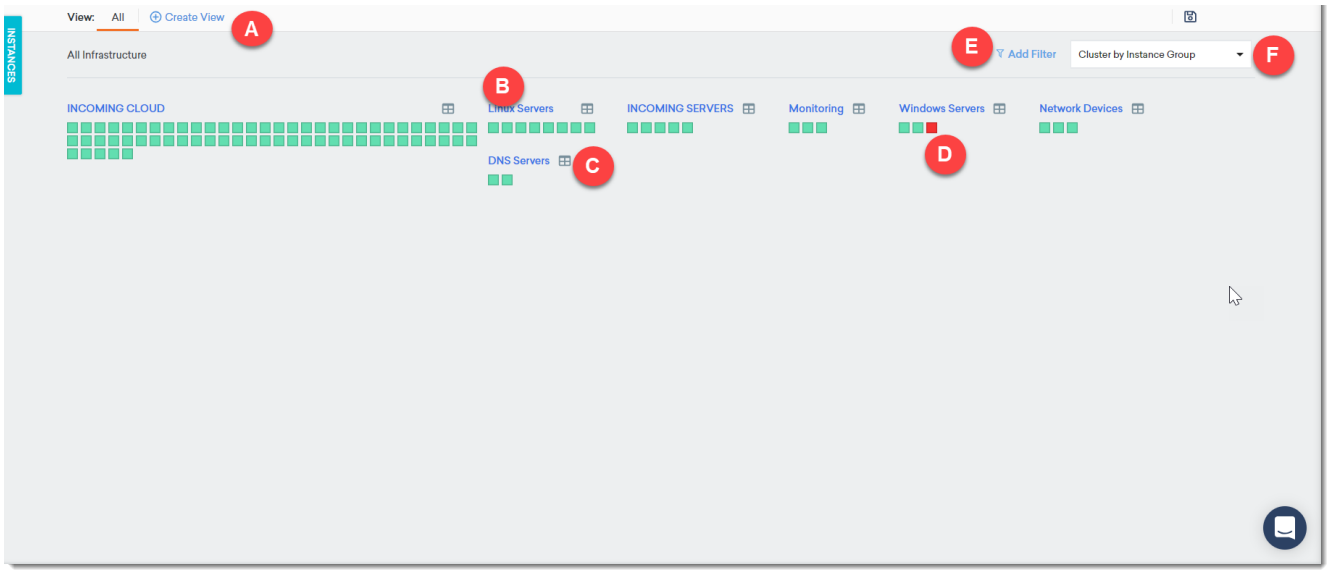
Part 4: Visualization

There are several ways to view and analyze your infrastructure in FortiMonitor. The **Infrastructure Map** provides a high-level view of your infrastructure and allows you to immediately identify if there is an incident, which instance is affected, and if related instances are also experiencing issues. **Dashboards**, on the other hand, can be utilized to analyze data and diagnose performance issues in your infrastructure. You can also use a **Status Page** to expose the current status and historical availability of your infrastructure to your customers and end-users. In addition to the **Infrastructure Map**, **Dashboards**, and **Status Page**, you can also view the status of your infrastructure on the instance level.

Infrastructure Map

Infrastructure Maps provide a high-level view of the health of your infrastructure in FortiMonitor. It allows you to cluster and explore the health of your nodes dynamically, helping you identify problematic hotspots in your infrastructure.
The **Infrastructure Map** is the default page after you have logged in to your FortiMonitor account unless you have configured a default dashboard. You can also access the Infrastructure Map by selecting **Monitoring** > **Infrastructure Map** in the header.
The following image and table explain the different components of the **Infrastructure Map**.





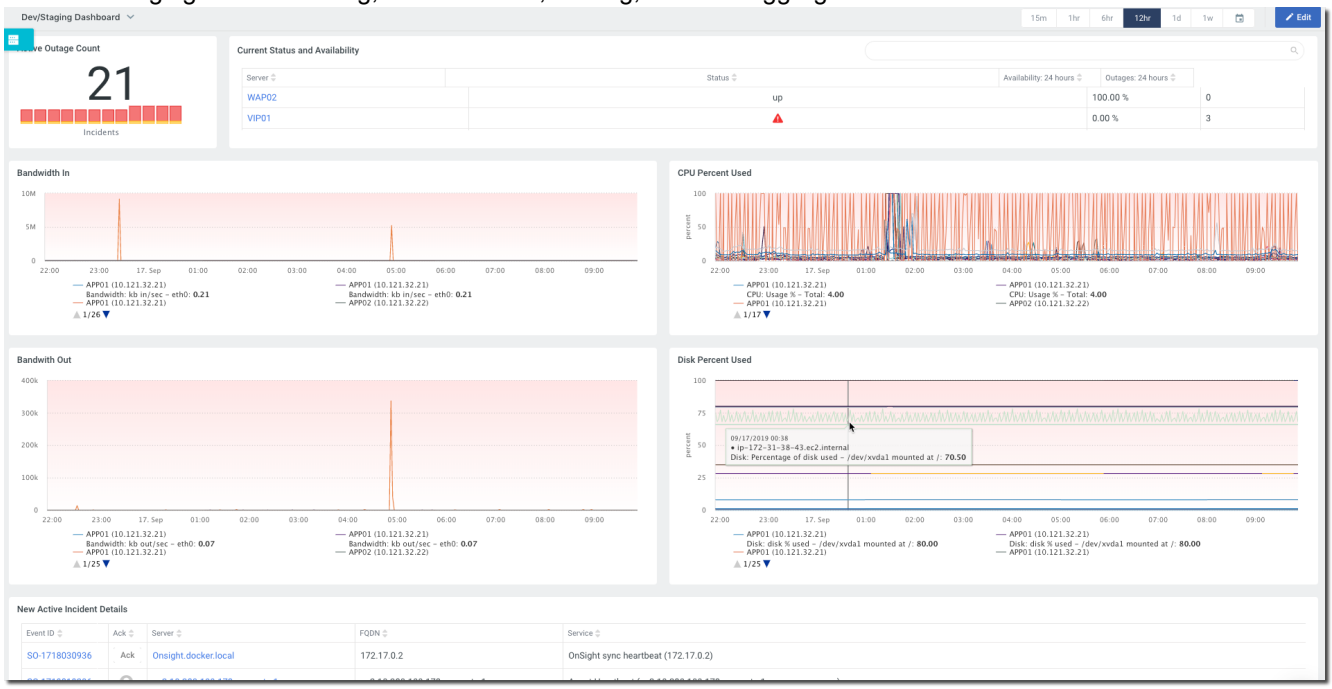
A	**Create view**
	Allows you to create an Infrastructure Map view. A view is a snapshot of your configured clustering and filtering settings.
B	**Clusters**
	Logical grouping of your instances. By default, instances are clustered by instance group.
C	**Table**
	Clicking the table icon displays a view of the configured metrics and status for all instances on the cluster.
D	**Nodes**
	On hover, displays a summary of the instance's configured metrics and health. When you click the node, the **Instance Drawer** slides out.
E	**Add filter**
	Filter the nodes by tag or server group.
F	**Filter by cluster**
	Cluster the nodes by tag, instance group, cloud region, or cloud instance type.

See Infrastructure Map for more information.

Dashboard

The Dashboard is a powerful mechanism for displaying and visualizing the various streams of data that are being captured by FortiMonitor. The Dashboard can be configured easily and uses a simple drag and drop builder for widgets

in order to achieve the layout you require. In addition, each of the widgets has several options for controlling different elements ranging from formatting, data selection, filtering, and data aggregation.



For more information on how to use Dashboards, see Dashboards.

Widgets

The Dashboard includes a catalog of widgets that you can choose from. Each widget provides a different suite of options for formatting, data selection, and data aggregation. See Dashboard widgets for a list of available widgets.

Time ranges

The time range selection determines what data appears on the **Dashboard** and widgets. When you select a time range, the aggregated data on the widgets change to reflect the selected time frame. The Dashboard includes preconfigured time ranges that you can choose from. You also have the option to create your own time range. For example, setting the start of the time range to the previous year can provide can help you analyze trends and the past performance of your infrastructure.

Tags

You can use **Tags to** segment and categorize your infrastructure and **Dashboards**. Every widget provides options to dynamically display the appropriate infrastructure and metrics by tag, rather than hard-coded selections. This allows your infrastructure to expand and contract as needed without requiring constant changes on what is being shown in your live dashboards. For more information on tagging, see Tags.

Public dashboard

With **Public Dashboards,** users who are not logged in to FortiMonitor will be able to see Dashboards that you share. To make a **Public Dashboard**, Select **Edit** on the **Dashboard** page then select the **Public** option. A public URL that you can share will become available.

Multi-tenant dashboard

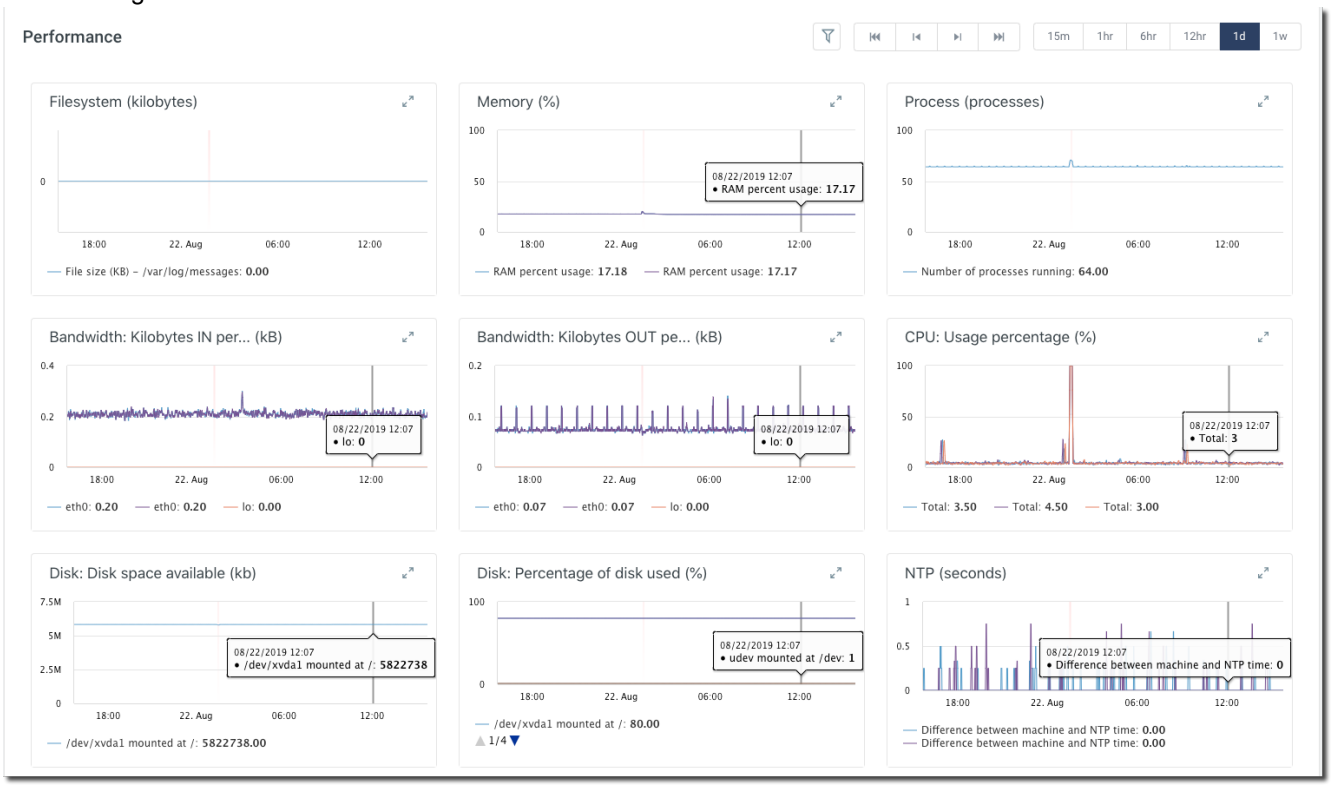
If you have a multi-tenant account or an MSP, you have the ability to query data and build dashboards across any of the subtenant accounts. This is a powerful method to maintain your single pane of glass view across your different customer segments.

In order to start, create a new dashboard and select the option for **Customer Data**. This option can only be set on new dashboards and upon initial creation.

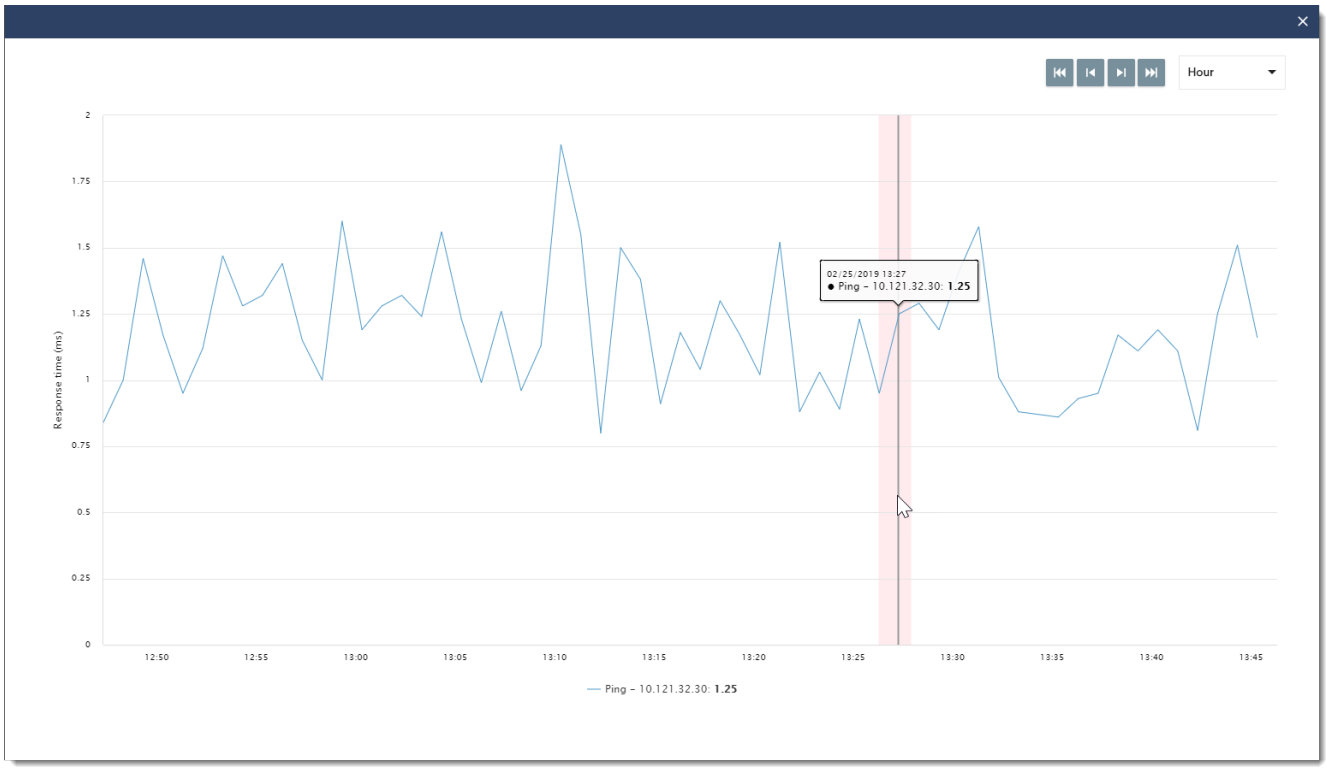
For more information see Multi-tenant dashboards.

Instance details page

The **Instance Details** page exposes all of the relevant information about the instance, including the instance performance, monitoring configuration, incidents, maintenance schedules, availability, and more. Performance-based graphs are displayed in the **Performance** section of the **Instance Details** page to help you visualize the status of each metric configured for the instance.



Clicking the expand arrows on the metric will show you a graphical representation of that metric's data and incidents. Any incident will be presented as a red bar in the graph.

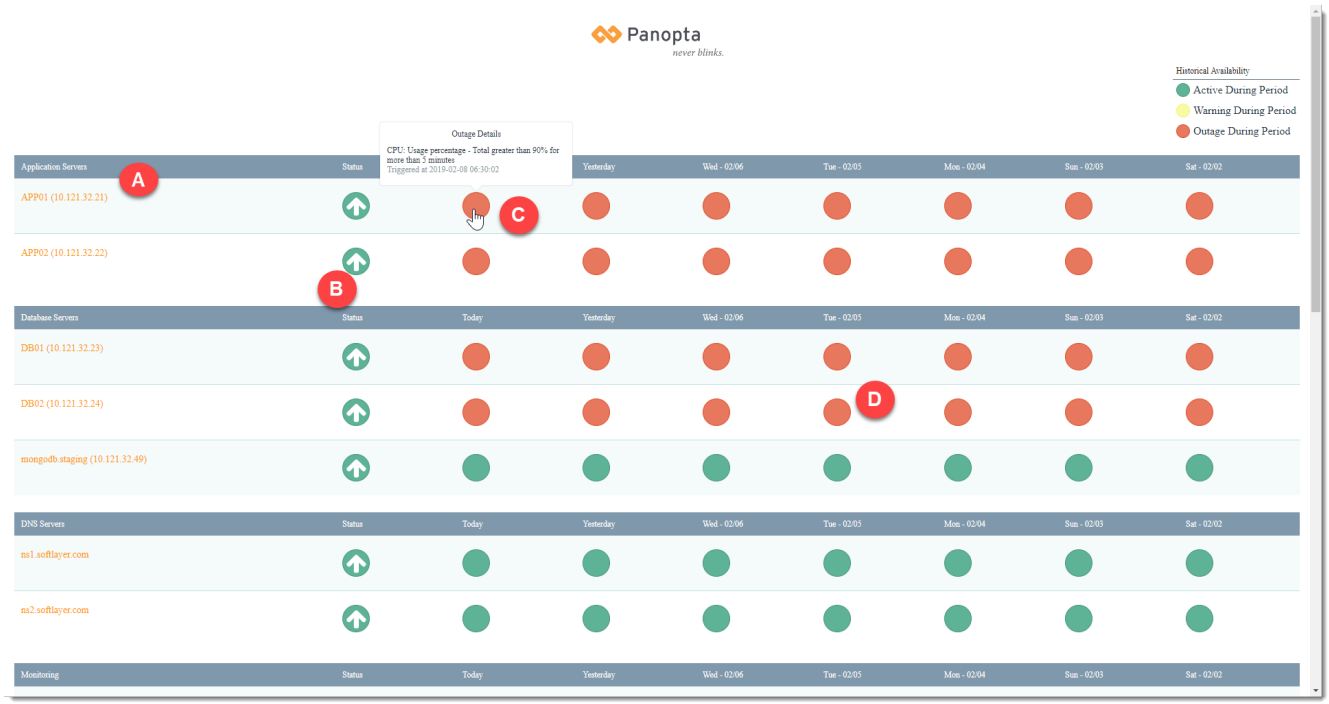


See the Instance Details page for more information.

Status page

A **Status Page** is a tool that you can use to expose your current status and historical availability to your customers and end-users. The Status Page can be used to update your customers about issues such as downtime or maintenance.

The following image and table explain the different components of the **Status Page**.



A	**Instances**
	Clicking an instance will display a detailed report for that instance
B	**Status for the day**
	The instance's current status, a green arrow means the instance is up and a means it's down
C	**Status on hover**
	On hover, the status of the instance for that day is displayed
D	**Status for the week**
	The performance of an instance over the past week. A green dot indicates that your server was up all of that day, a yellow dot indicates that there was at least one alert, and a red dot indicates that there was an outage.

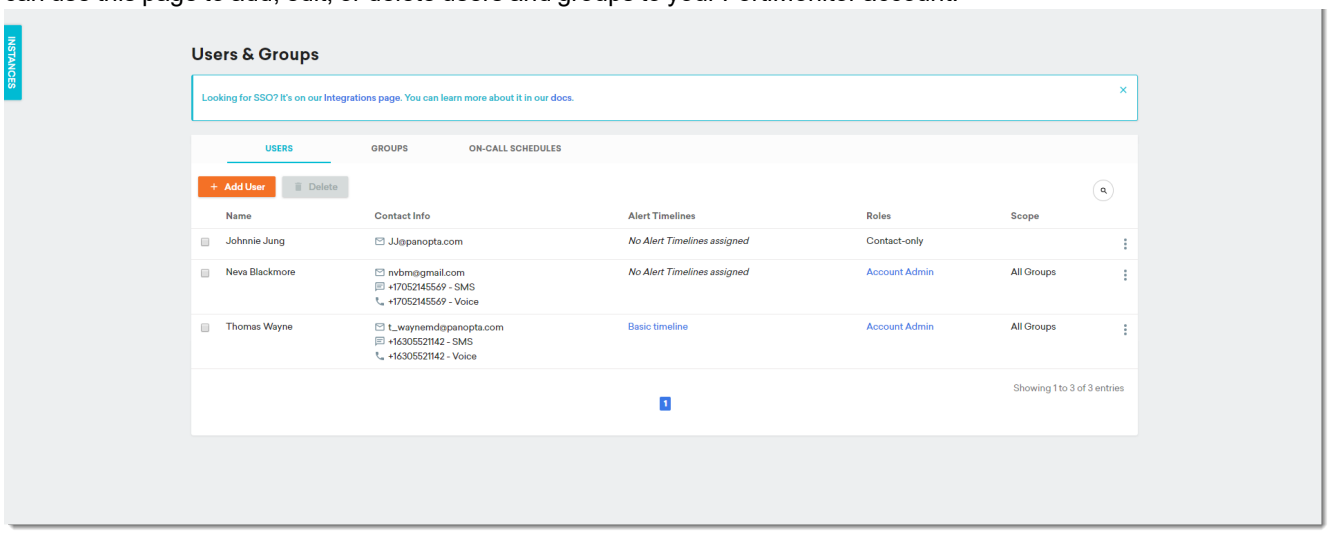
To view an example **Status Page**, go to http://status.panopta.com/example.

Part 5: Team Management

This article describes how to manage users and groups in FortiMonitor, including creating users and setting up access control levels and system roles. Select **Settings** > **Users, Groups & On-Call** from the navigation bar to access the page.

Users and groups

User accounts, groups, and on-call schedules in FortiMonitor are managed in the **Users, Groups & On-Call** page. You can use this page to add, edit, or delete users and groups to your FortiMonitor account.



The following table provides a description of the fields available in a user account.

Field	Description
Name	The user's name. The name defined here will be used in the control panel and in all communications.
Timezone	The user's timezone.
Account type	Indicates whether the user can log in and access the control panel (Control Panel Access) or merely receives notifications (Alerting Only). Further access refinements for those with Control Panel Access can be made in the **Access control** section below.
Tags	(Optional) User tags are used to restrict access to dashboards.
Default dashboard	(Optional) If a default dashboard is selected, the user will be redirected to this dashboard when they log in to FortiMonitor.
ACL roles	The scope of actions and infrastructure the user has access to within FortiMonitor. Learn more about access control in Access control.
Contact information	The `Contact Information` pane allows you to add various contact methods for the user, which are then later used in Alert Timelines. However, only email is required.

Note: By default, only accounts with the **Account Admin** role can add, delete, and modify users. Create or clone a custom role if you want to add these permissions to users but not necessarily have all the permissions of an Account Admin.

See Add users to your account for more information about adding users to your account.

Groups

To streamline management of users you assign to **Alert Timelines**, you can optionally add multiple users to alerting groups. Instead of assigning users one by one to a timeline, you can create a group and assign that group to the timeline. For example, if you have a group of Linux admins that you want to assign to an **Alert Timeline** for Linux servers, you can create a group called **Linux Admins** and assign that group to the timeline. If an incident occurs, all members of the group will be alerted.

To create and manage groups, see Create a Group.

Access control

FortiMonitor's access control (ACL) provides you with the ability to control which actions a team member can perform in FortiMonitor. The system is driven by roles, which are logical groupings of actions that a user can perform.

Note: A user can have multiple roles. These roles are layered on top of one another. For example, a user can both have the **Dashboard Admin** and **Server Admin** roles.

FortiMonitor provides the following system roles out of the box:

Role	Description
Account Admin	Able to perform any activity within FortiMonitor
Server Admin	Able to perform most activities, with the exception of user, integration, and API management, as well as a few other ancillary activities
Dashboard Admin	Full management of dashboards
Dashboard Viewer	Read-only access to dashboards
Incident Responder	Slightly more advanced than a read-only user. Allows the user to view instances, start maintenance, pause monitoring, and more
Billing Admin	Access to billing only
API Full Access	Full access to API functionality, including read and write operations
API Read-only Access	Read-only access to all API operations

See Access control for more information.

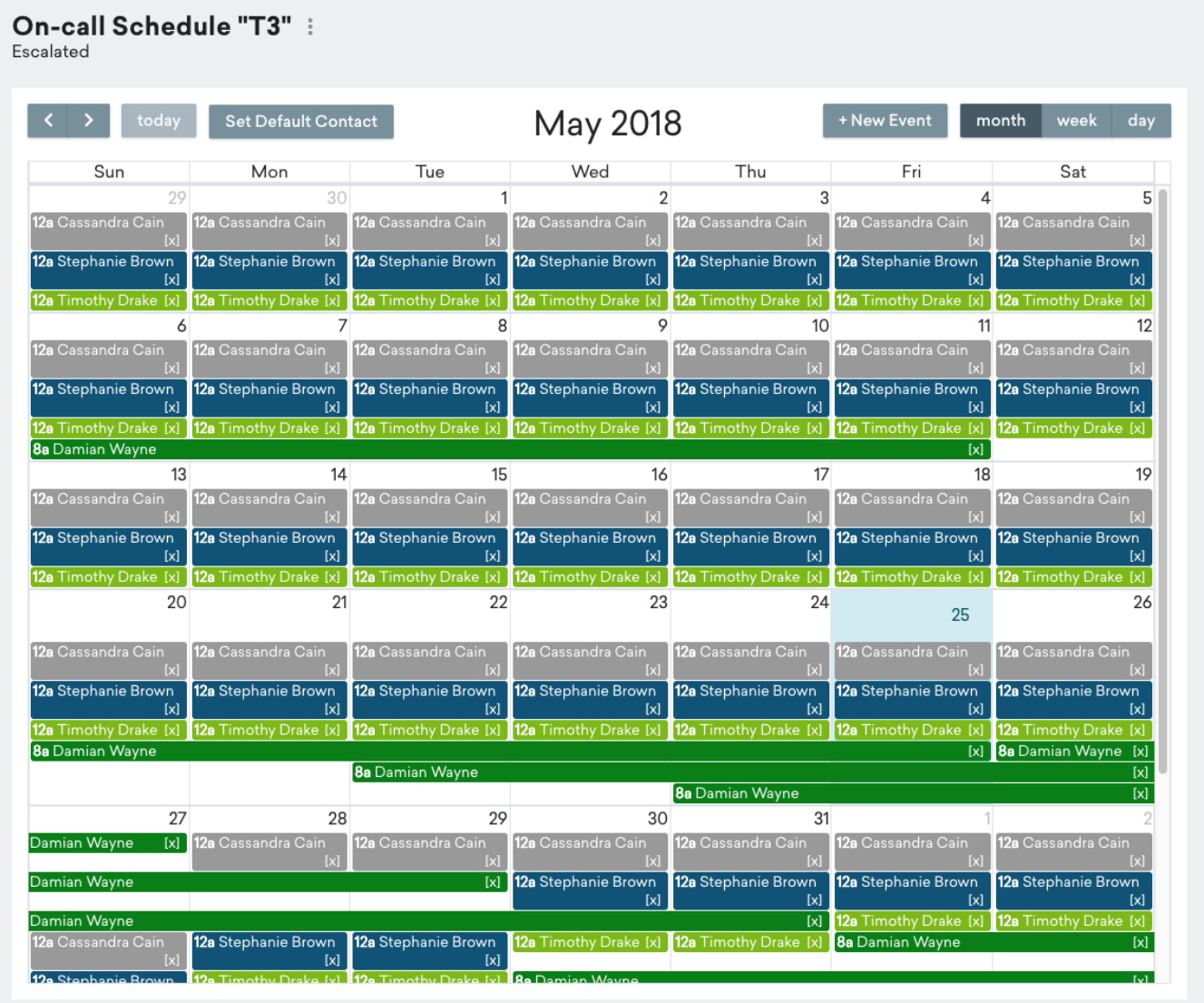
Custom roles

You can create a custom role if you want to fine-tune the roles you use in FortiMonitor. Custom roles can be comprised of any number of actions that are available in FortiMonitor. For instance, you may want a role that allows a user to view and edit instances, but read-only access for network devices. When creating a custom role, you may choose to either start from scratch or clone an existing role.

See Add Custom roles for more information.

On-Call schedules

You can create, modify, or view On-Call schedules in the **Users, Groups & On-Call** page.

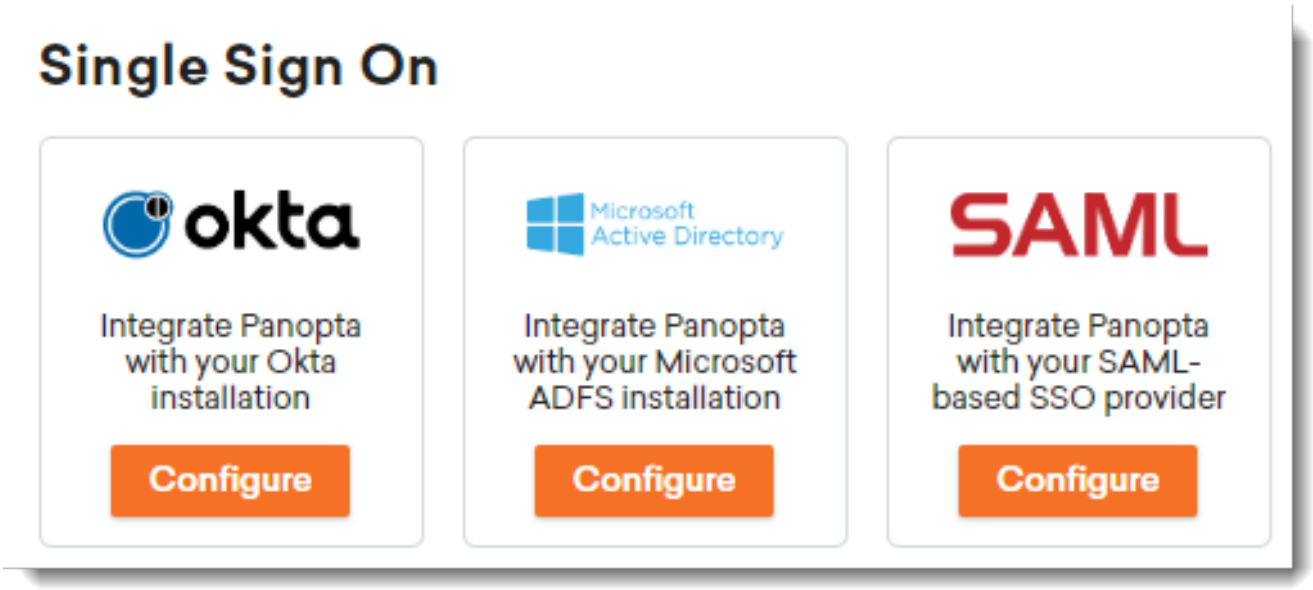


For more information, see Set up an On-Call schedule:

Single Sign-On (SSO)

Setting up SSO for your FortiMonitor account allows you to use credentials configured in your organization's internal authentication tool (Active Directory, SAML, Okta). When you use SSO, users can benefit from not specifying separate credentials when logging into FortiMonitor. Administrators can also save time by having a central management location for user accounts and authentication. To set up and use SSO, select **Settings** > **Integrations** then follow the steps provided in Single sign-on (SSO).

.

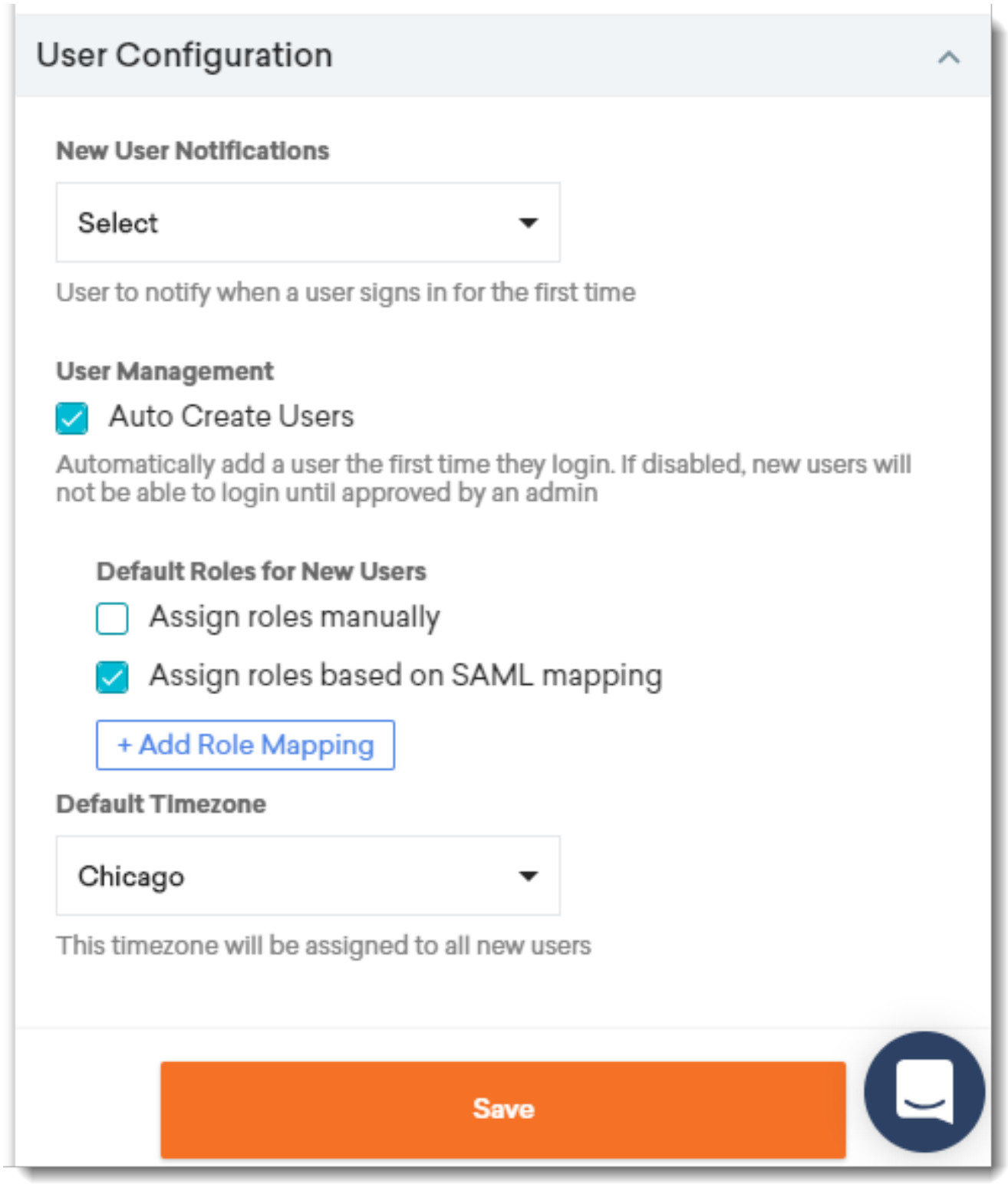


User configuration

During SSO configuration, you have several options to control what happens when new users with your SSO credentials log in for the first time to FortiMonitor. This includes:

- **New user notification** - The user to notify when a user logs in for the first time.
- **Auto-create users** - Select this option to automatically add users the first time they log in. If this option is not selected, new users will not be able to log in until approved by an administrator.
- **Default roles for new users**
  - **Assign roles manually** - Manually select a role to assign to new users. You can select more than one role. See Access Control.
  - **Assign roles based on SAML mapping** - Automatically assign roles to new users based on the configured SAML mapping. See SSO-based roles for more information.
- **Default timezone** - The timezone to assign to the new user.

These options are available under the **User Configuration** module. See Single sign-on (SSO) for more information.



Part 6: Reports

This article describes the different methods to pull data and create reports from FortiMonitor. The following options are available:

- Canned reports
- Scheduled reports
- Ad-hoc reports

In addition, you can also download any previously generated report from the **Report History** page.

Canned reports

Canned reports are pre-configured reports that can be generated on-demand or on a recurring basis. Once a report is generated, it will be sent to the email address you selected while creating the report. You also have the option to add other FortiMonitor or external users as recipients to the report. Similarly, several configuration options are available when generating a canned report. For more information, see Canned reports.

The following table describes the available canned reports in FortiMonitor.

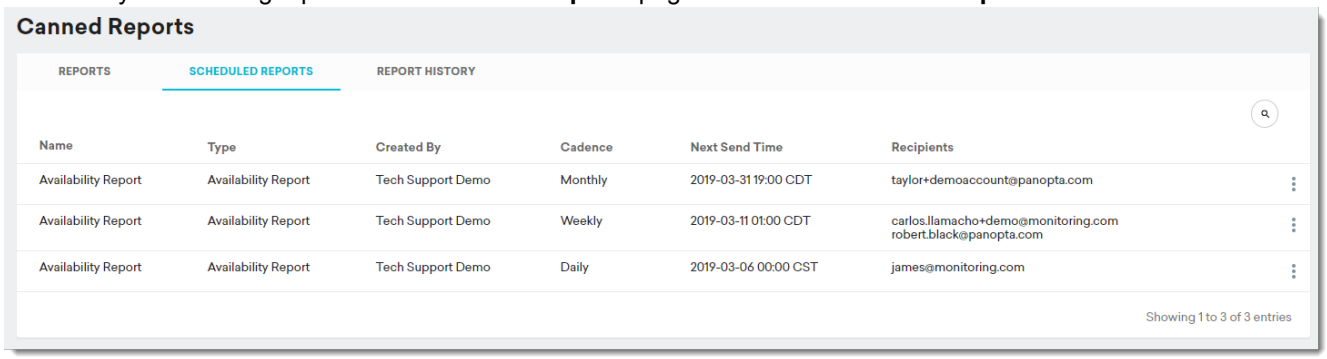
Report type	Description
Agent Status report	Provides the status and current version of all the Agents in your infrastructure. This can help you identify any inconsistencies as well as Agents that require updates.
Availability report	Provides a full rundown of the current status of your infrastructure in FortiMonitor for a given time range.
Capacity Planning report	Provides metric trends and usage to help you with capacity planning.
Incident report	Shows you a detailed list of all the incidents that occurred in your infrastructure over a selected period of time.
Tag Audit report	Shows which tags are in use and where they are applied.
User Audit report	Shows a detailed list of all users and what permissions they have. The report also shows you which users are active, the last time they logged in, or whether or not they have the mobile app installed.
Instance Configuration report	Allows you to export the current monitoring setup on a determined group of instances.
Metric Export report	Shows the performance of each metric in your FortiMonitor account.
Metric Configuration report	The Metric Configuration report is similar to the Instance configuration report but only reports the metrics. The configuration difference between the two is that under **Scope** you have an additional option to choose to filter the metrics by tag, metric type, report on only network service metrics or only agent metrics.
Service Performance report	The Service Performance report is similar to the Availability Report but allows you to view the Availability in a cross-tab format, with metrics on the y-axis and instances on the x-axis.

MTTR and MTTA Report	The Mean Time to Resolve (MTTR) and Mean Time to Acknowledge (MTTA) report provides information on how long incidents are resolved and acknowledged after their occurrence.
Historical SLA Performance Report	Provides a monthly or quarterly breakdown of your infrastructure's performance.
User Incident Activity report	Shows each user's incident activity

To create a canned report, see Create a report.

Scheduled reports

When creating any of the canned reports, you also have the option to create recurring or scheduled reports. You can view all of your recurring reports on the **Canned Reports** page under the **Scheduled Reports** tab.



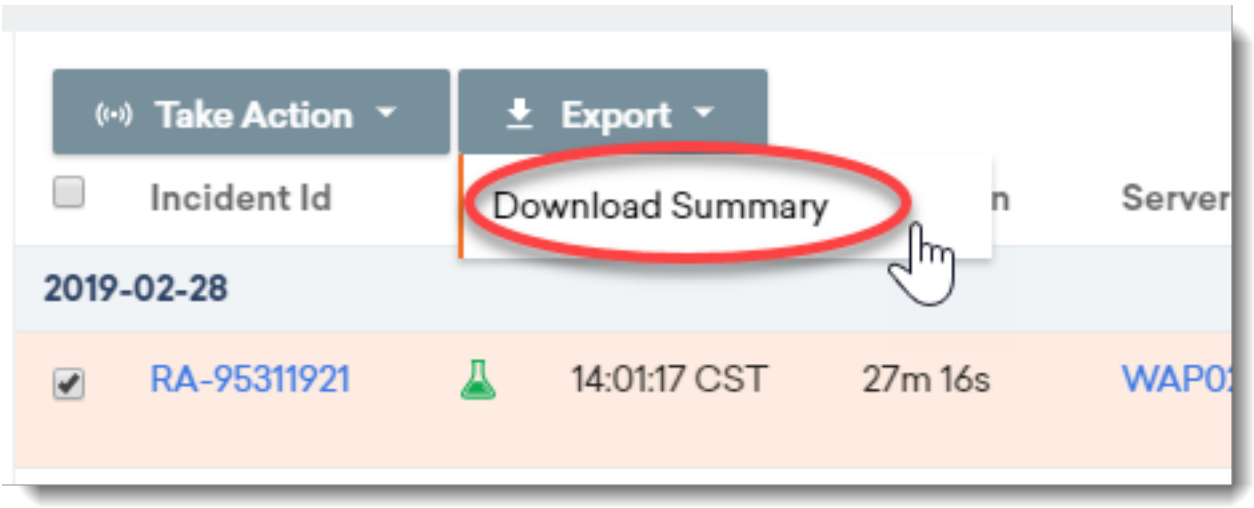
There are several options available when creating a scheduled report, including recipients of the report, and the cadence or how often a report is generated. Similar to canned reports, the configuration options vary depending on the type of report you choose. To schedule a report, see Create a report.

Report history

The **Report History** tab displays all the previously generated reports for your infrastructure. You can download a copy of any of the reports by selecting **Download**.

Ad-hoc reports

From the **Incident History** page, you can generate a report on any incident that occurred in your infrastructure. To do so, select one or more incidents then select **Export** > **Download Summary**. There are also several filters available to refine the list of incidents displayed on the page.



Part 7: CounterMeasures

This article describes the features and functionality of **CounterMeasures**. CounterMeasures are automated remediation and mitigation actions triggered by the FortiMonitor Agent whenever a metric threshold is crossed for more than a configured amount of time. Actions range from simple debugging commands (**top**,**netstat**, or **vmstat**) to complex actions such as making API calls or restarting a service when a threshold is breached. CounterMeasures can increase the level of automation within your operations workflow, either by using the out of the box FortiMonitor CounterMeasures or by creating your own.

CounterMeasures are represented throughout the control panel by a beaker icon. This icon indicating a configured CounterMeasure is available anywhere an incident is listed. You can also see a list of all the configured CounterMeasures in your account by going to Settings > CounterMeasures.

Note: To enable CounterMeasures, you need to have the latest version of the FortiMonitor Agent installed on your instance. See Update the FortiMonitor Agent to upgrade your Agent.

Slack integration

One of the biggest benefits of CounterMeasures is the ability to quickly arm yourself with information when an incident arises. You can configure CounterMeasures to send the information it retrieves directly in Slack, saving you the time to log into the offending instance. See CounterMeasures Slack Integration for more information.

Enable CounterMeasures

Go to the following sections to learn more about enabling CounterMeasures on your instances:

- Linux CounterMeasures
- Windows CounterMeasures

Example workflow

CounterMeasures have a simple workflow that is driven by the FortiMonitor Agent. Here is an example:

1. Enable **CounterMeasures** in your Agent configuration file or manifest file.
2. Add a CounterMeasure to a metric threshold, such as Disk % used.
3. If the threshold is crossed, the CounterMeasure will be run.

Any output provided by the CounterMeasure will be attached to the incident record and will be available in the FortiMonitor control panel, incident log, or Slack.

CounterMeasure actions

Standard Linux CounterMeasure actions

Out of the box, the FortiMonitor Agent comes with standard CounterMeasure actions that can be utilized.

Action	Description
Top	Displays information about CPU and memory utilization by process
dmesg	Displays information from the kernel
Netstat	Displays network connections for the TCP, routing tables, and a number of network interface and network protocol statistics
Reboot Server	Reboots the server
Vmstat	Retrieves the following virtual statistics: processes, memory, paging, block IO, traps, and CPU activity

Standard Windows CounterMeasure actions

The following CounterMeasures are available if you have the Windows Agent installed.

Action	Description
Top	Displays information about CPU and memory utilization by process
Application Error Logs	Retrieves the last 50 Application Error logs
Get Service Status	Retrieves the status of all running services
Netstat	Displays network connections for the TCP, routing tables, and a number of network interface and network protocol statistics
Reboot	Reboots the server
Restart IIS	Restarts the IIS web server service

System Error Logs	Retrieves the last 50 System Error logs
Vmstat	Retrieves the following virtual statistics: processes, memory, paging, block IO, traps, and CPU activity

Custom actions

In addition to the standard CounterMeasure actions, you can also create your own by following the steps detailed in:

- Custom Linux CounterMeasure actions
- Custom Windows CounterMeasures actions

Remote actions

You can centrally manage your CounterMeasure plugins by adding the `countermeasures_remote_plugins` parameter to your Agent manifest file that points to an archive of Agent plugins. When the Agent is installed, it will download and extract the directory in the proper location. Zip, tar, and python file types are supported.

If you would like updates to your CounterMeasures plugins to automatically be applied, you can optionally set `countermeasures_refresh_plugins` to the number of hours for the refresh interval. This will help with keeping all your servers up-to-date.

See Utilizing Remote Actions for more information.

CounterMeasure metric threshold

CounterMeasures are configured together with metric thresholds. See Configure a CounterMeasure to configure CounterMeasures and thresholds. An orange beaker icon indicates that an approval is required. Selecting the icon displays a modal where you can approve the appropriate CounterMeasure.

Approval

An **Approval** option is available for situations where you don't want a CounterMeasure to execute without a final authorization. To require approval for CounterMeasure actions, select the **Require Approval** option when configuring the CounterMeasure.

CounterMeasure status

The CounterMeasureslife cycle is detailed in the following list:

- **Pending:** state of an action that is scheduled to execute in the future and does not require approval or has already been approved. The beaker icon will appear grey.
- **Pending Approval:** state of an action that is scheduled to execute in the future but will not do so until approved. The beaker icon will glow orange.
- **In Process:** state of an action that has been synced down to the Agent to execute. The beaker icon will appear gray.

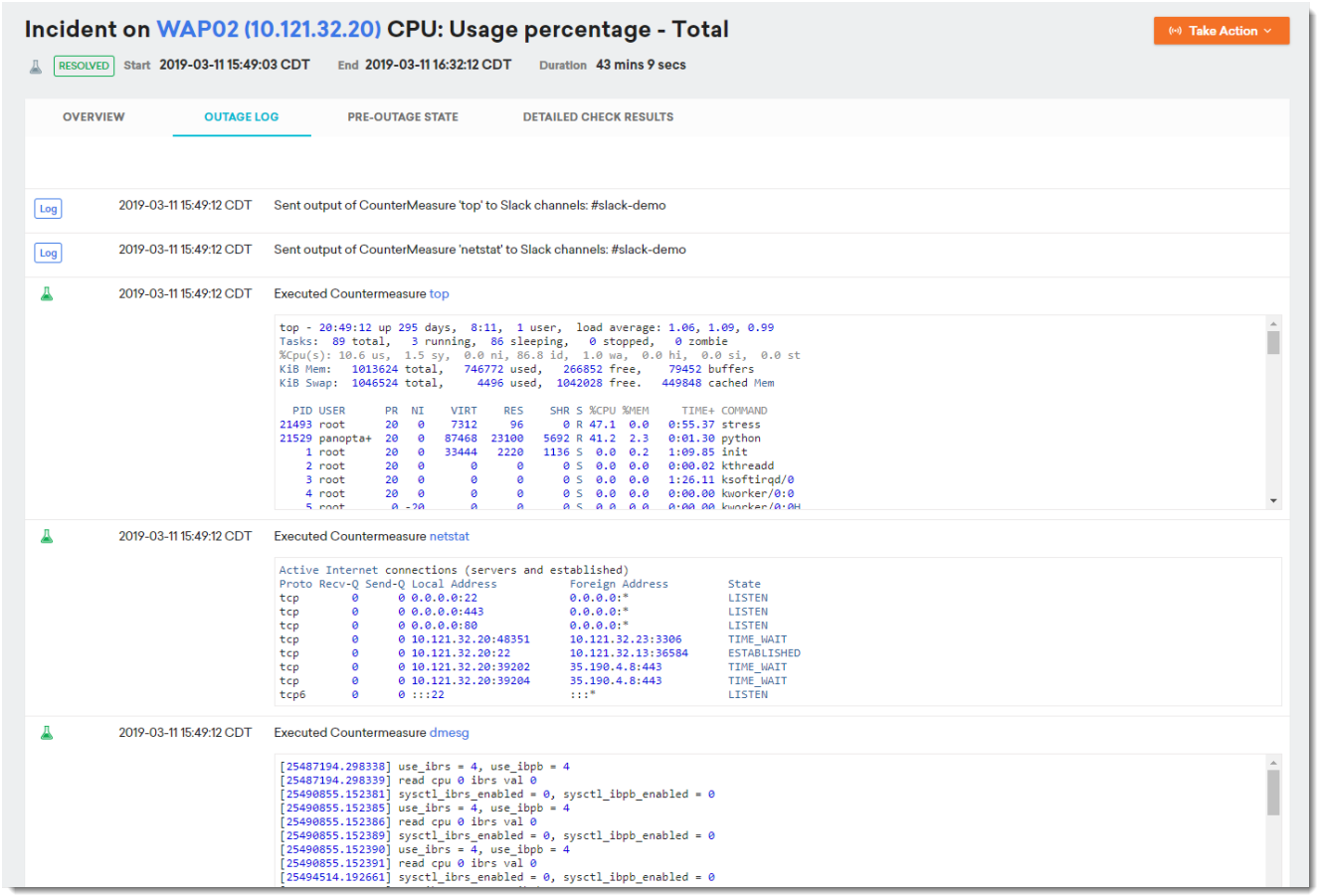
- **Complete:** state of an action that has executed successfully. The beaker icon will appear green.
- **Error:** state of an action that attempted to execute but returned an error while executing. The beaker icon will appear red.
- **Skipped:** state of an action that was at one time pending but whose underlying incident was resolved before executing. The beaker icon will appear grey.

CounterMeasure output

You can view a CounterMeasure's output from the following:

- CounterMeasures details panel
- Incident details page

The following figure shows the CounterMeasure output from the outage log.



See Viewing CounterMeasure output for more information.

Billing

Your account is only charged for CounterMeasures when they are actively configured on an instance. That is, only instances that have CounterMeasures configured on Alert Thresholds.

Supported browsers

Although it relies on on-premises software components for data collection, FortiMonitor is principally a cloud-based Software as a Service (SaaS) product. Access to that information analyzed in the cloud is via a web based user interface (WebUI) using a browser. This document addresses compatibility and known issues with the more popular browsers.

Major browsers such as Chrome, Firefox, Safari, Internet Explorer, Edge, Opera, and Android are updated at a fast pace, and as a result, it is not feasible to test all features of the FortiMonitor interface against all versions of the major browsers.

FortiMonitor development has focused on ensuring a positive experience on large high-resolution screens with the most popular browsers, particularly Chrome and Firefox. These two browsers are available for most platforms.

Chrome

Chrome uses Google's Blink browser engine. As of Chrome version 88.0 as deployed on MacOSX and Windows, there are no known issues running the FortiMonitor WebUI with Chrome. Other browsers which use the Blink engine and which should therefore offer a similar rendering of web content include Microsoft Edge and Opera.

Firefox

Firefox uses Mozilla's Gecko browser engine. As of Firefox version 85.0 as deployed on MacOSX and Windows, there are no known issues running FortiMonitor WebUI with Firefox.

Safari

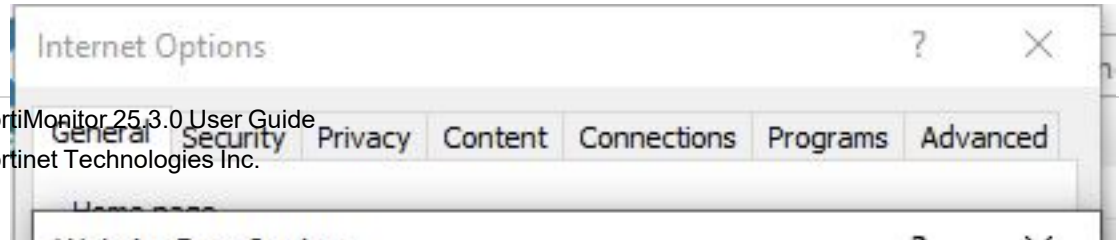
Safari uses Apple's WebKit browser engine. As of Safari version 14.0 as deployed on MacOSX, there are no known issues running FortiMonitor WebUI with Safari. However, the engine and browser are not extensively tested for compatibility with FortiMonitor.

Internet Explorer

Internet Explorer uses Microsoft's Trident browser engine. Development of this engine has been discontinued, and Microsoft is promoting Edge in place of IE. However, IE is still found on many versions of Windows.

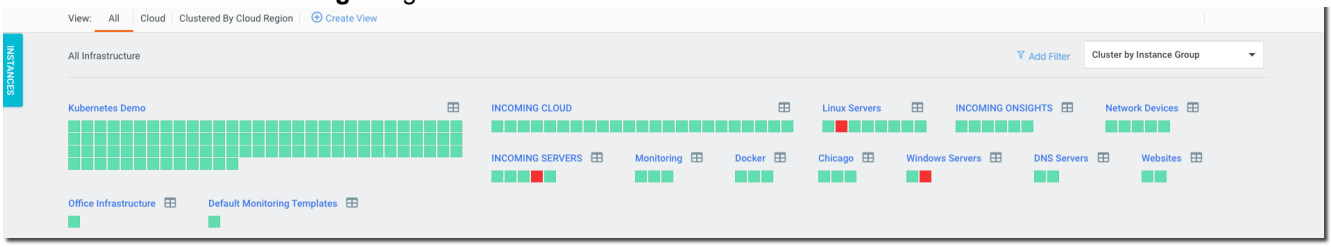
As of IE version 11 as deployed on Windows 10 and Windows 2019 Server, there is a known issue with the handling of HTML5 local storage. This affects some FortiMonitor dashboards and visualizations, causing them to fail to load. If this happens, enable the **Allow website caches and databases** option. To do this, perform the following steps:

1. Click the gear icon then click **Internet options**.
2. Click **Settings** under **Browsing history**.
3. In the **Website Data Settings** window, click the **Caches and databases** tab.
4. Enable **Allow website caches and databases**
5. Click **OK**.



Infrastructure Map

The Infrastructure Map is a high-level view of the health of your account that lets you cluster and explore the health of your nodes dynamically, helping you identify problematic hotspots in your infrastructure. The Infrastructure Map is the default page you will land on when visiting FortiMonitor unless you have a default dashboard set. Otherwise, it's available under the **Monitoring** navigation item.



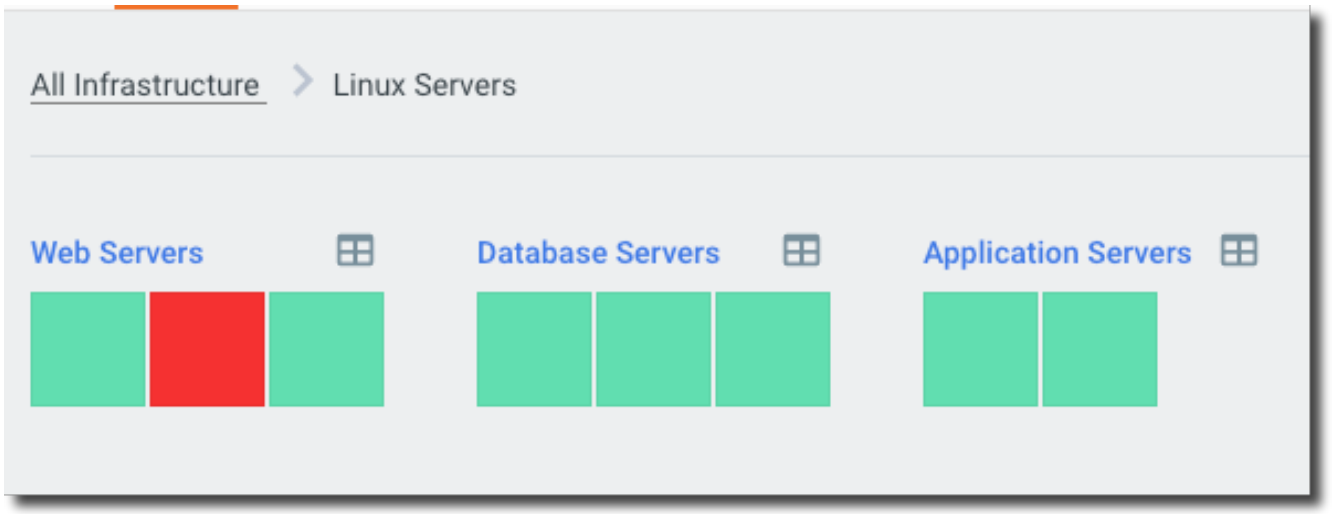
Clustering

Clusters are logical groupings of your infrastructure. By default, your instances will be clustered by instance group. Using the Clustering selector in the top right corner, you can cluster infrastructure using a number of options:

- **No Clustering**: show all of your infrastructure in a flat group
- **Cluster by Tag**: each tag you select will become its own cluster
- **Cluster by Instance Group**: the default setting - each of your top-level instance groups will be used as a cluster, including all the instances recursively contained within it
- **Cluster by Cloud Region**: if you're utilizing our native cloud connections with AWS or Azure, you can create clusters by their given regions
- **Cluster by Cloud Instance Type**: if you're utilizing our native cloud connections with AWS or Azure, create clusters by specific instance types (EC2, RDS, etc.)

Diving

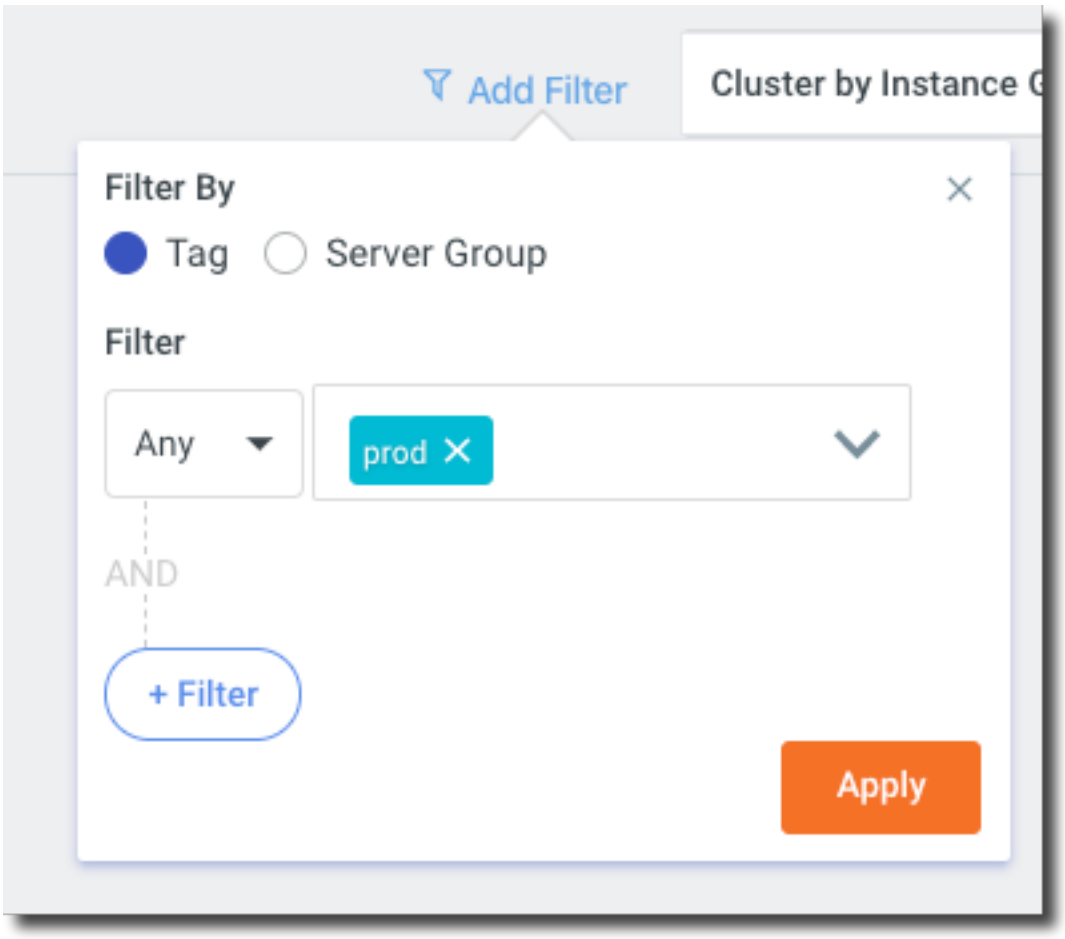
Diving is the ability to move up and down through the layers of your infrastructure, regardless of your clustering method. To dive into the next layer, simply click the name of a cluster. For instance, when using the default cluster method of by-instance group, you can click the name of any instance group to "dive" into it, viewing the groups within it, if any.



Each level you dive into, you can cluster and filter as needed, including mixing clustering types.

Filtering

Filters allow you to control the available set of instances that are clustered in your Infrastructure Map. For instance, if you only care about exploring your **prod** instances, you can add a tag filter of **prod**, assuming they're tagged accordingly. Only instances with the **prod** tag would then be shown - even when you dive down through your infrastructure.

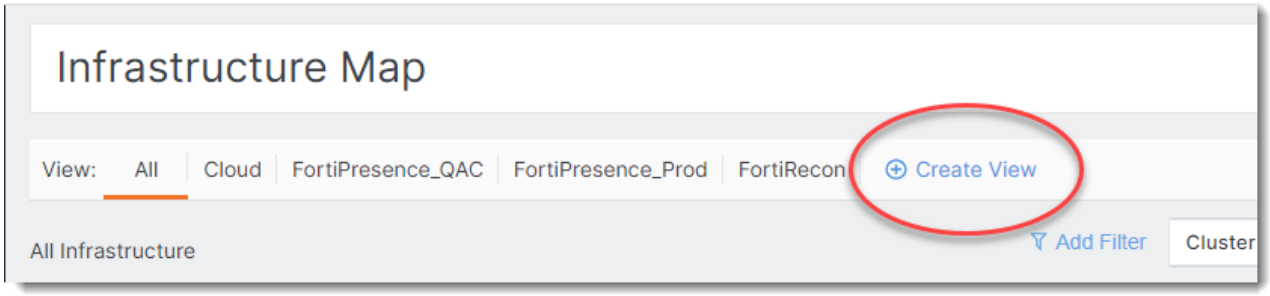


Views

Views are snapshots of your configured clustering and filtering settings - similar to a bookmark. Each view you create (using the **Create View** button) will be created as a new tab, allow you to quickly switch between frequently-used views.

To create a view, perform the following steps:

1. Click **Create View**.



Enter a name for the view.



3. Choose the options to enable:
   a. Default
   b. Viewable by teammates
4. Click **Save**.
5. Configure the new view by adding filters or choosing a clustering option.
   To save the view, click the **Save** icon.
6. 

Users, Groups, and Authentication

- Add users to your account
- Single sign-on (SSO)

Add users to your account via FortiMonitor

There are three ways to add users to FortiMonitor depending on when your FortiMonitor account was created. Accounts created prior to September 15, 2024 are treated as legacy accounts.

The following table describes the different methods:

Method	Account type	Description
Create FortiCloud users from FortiMonitor on page 60	New accounts	Add FortiCloud users directly from FortiMonitor. For new customers, FortiCloud users can be added from the FortiMonitor control panel.
Add users using FortiMonitor on page 57	Legacy accounts	Add users using the FortiMonitor control panel. These users will only exist in the region where they are created.
Sync users from FortiCloud on page 59	Legacy accounts	Add users created via FortiCloud to FortiMonitor. A discovery process is ran nightly to sync users created in FortiCloud to FortiMonitor. You can also manually sync users to create the users immediately.

Add users using FortiMonitor

To add a new user to your account, perform the following steps:

1. From the navigation menu, select **Teams & Activity> Users, Groups & On-Call**.
2. Click **Add** then **Add User** .
3. Enter the user's email address.
4. Select the User Type.
   a. Alerting Only - User only receives notifications.
   b. Control Panel Access - User can log in and access the control panel. Further access refinements for those with Control Panel Access can be made in the **ACL Roles** section below.
5. Enter the required information.

Field	Description
General	
Name	The user's name, as it will appear throughout the control panel and in communications.
Avatar	Upload a photo or avatar for the user.
Timezone	The user's timezone - All displayed and computed times are adjusted for the user (this does not affect other users on the account).
Dashboard Access	Configure which dashboards this user has access to. You can configure this user to have access to just the dashboards you selected or all of the dashboards in your account.
Default Dashboard	(Optional) If a default dashboard is selected, the user will be redirected to this dashboard when they login to FortiMonitor.
Tags	(Optional) User tags are used to restrict access to dashboards.
Single Sign-on	To allow certain users to still login via email and password, check the **Allow Non-SSO Login** checkbox.
Password	Select how you want to setup up passwords for the new user. You can manually create a password for the user by selecting **Specify Password**. Selecting **Send Invite Email** sends an email to the email address provided with a link that allows the user to set a password. Passwords must be at least 8 characters long and contain at least one letter, number, and special character.
Permissions	
Access Control	The scope of actions and infrastructure the user has access to within FortiMonitor. Learn more about access control here.
Instance Group Access	Select which groups do the roles selected in **ACL Roles** apply. You have the option to apply the roles to all groups by selecting **All Groups**. Selecting **Selected Groups** allows you to specify which groups the **ACL Roles** apply to.
Alerting	
The **Alerting** pane allows you to add various contact methods for the user, which are then later used in Alert Timelines. However, only email is required. You can also add labels (optional) to each contact information.	

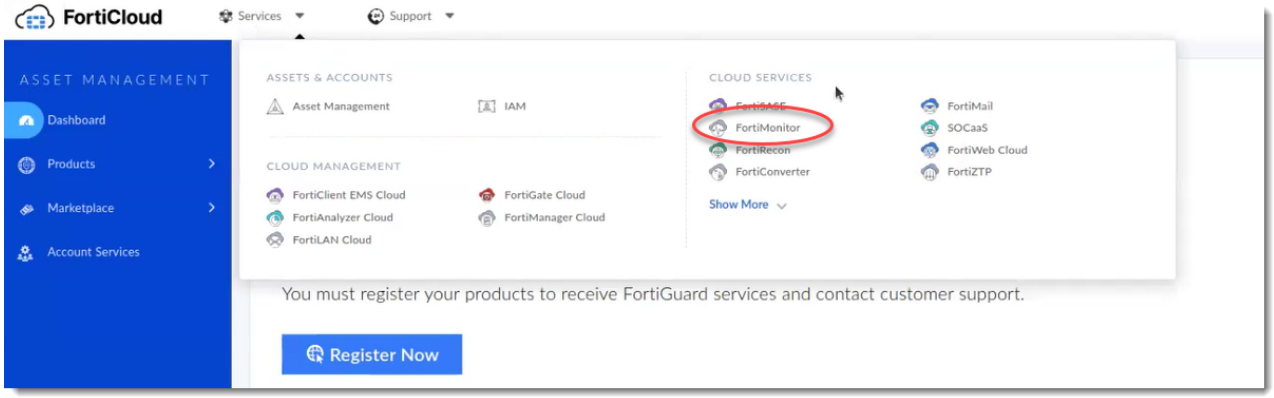
For each email address and phone number provider, you can set which alert types that specific contact method receives.

Alert Type	Description
Clear (All Clear)	All Clear messages are sent by default when an incident resolves
Ack (Acknowledgment)	Acknowledgment alerts are sent to all users who have previously been alerted on the incident when a member of their team acknowledges an alert
Broadcast	Messages can be broadcast to other team members in order to communicate about an incident.

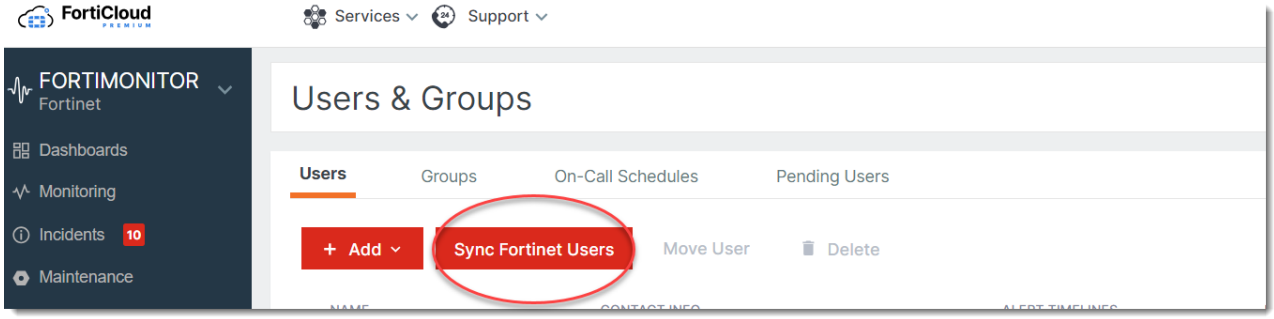
6. Select **Save**. The new user is added to your FortiMonitor account. Note that this user will only exist in the region where it was created.

Sync users from FortiCloud

1. Create a FortiCloud user by following the steps detailed in this page.
2. After creating a FortiCloud user, go back to the main FortiCloud page.
3. Go back to FortiMonitor by clicking **FortiMonitor** from the **FortiCloud Services** menu.

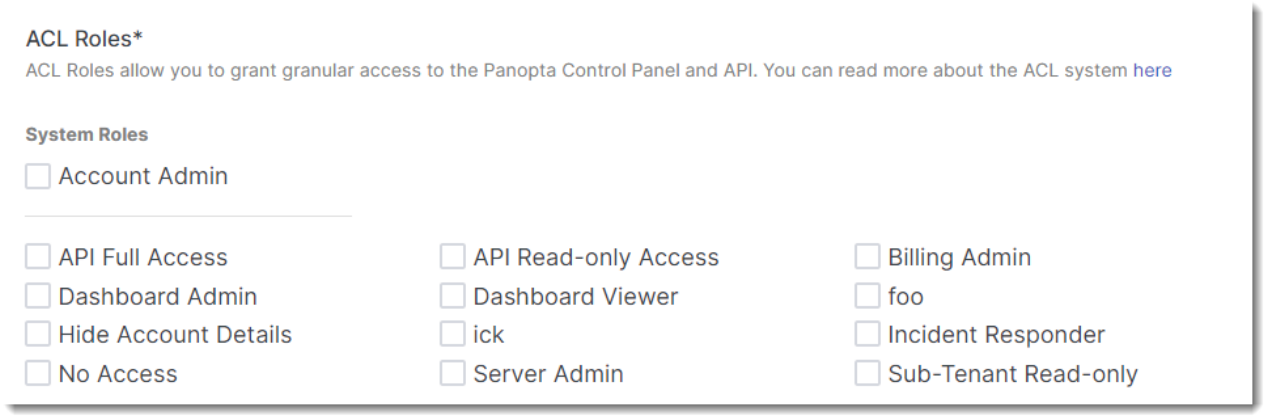


4. From the FortiMonitor control panel, go to **Teams & Activity** > **Users, Groups & On-Call**.
5. Click the **Sync Fortinet Users** button. This will add the created users to FortiMonitor.
Note: A discovery process is ran nightly to sync users created in FortiCloud to FortiMonitor. Use the **Sync Fortinet Users** button if you want to sync the created users immediately.



6. Users created in FortiCloud will have the **No Access** ACL role enabled by default. An Administrator must set the

permissions in FortiMonitor. For more information, see Access control.



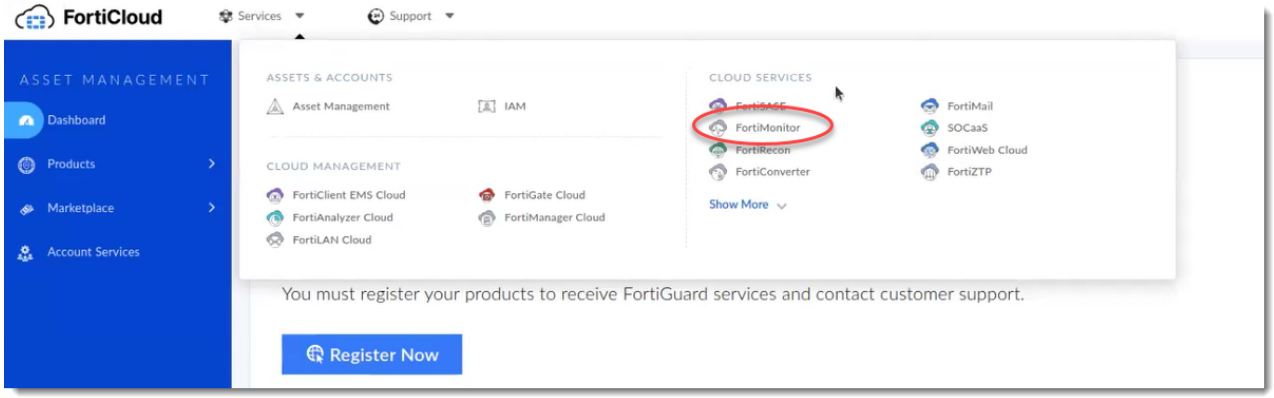
Create FortiCloud users from FortiMonitor

You must be logged into FortiMonitor using FortiCloud to be able to create FortiCloud users from FortiMonitor.

Note: This method is only for FortiMonitor accounts created on September 15, 2024 or later. For accounts created prior to September 15, 2024, see Add users to your account.

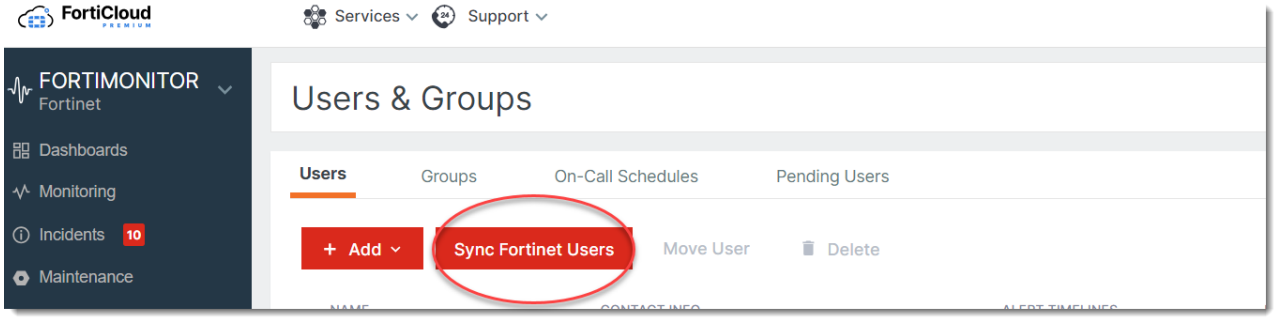
To add a new user to your account, perform the following steps:

1. From the navigation menu, select **Teams & Activity> Users, Groups & On-Call**.
2. Click **Add** then **Add User** .
3. Enter the user's email address.
4. Select the User Type.
   a. Alerting Only - User only receives notifications. Proceed to Add an Alert Only user.
   b. Control Panel Access - User can log in and access the control panel. Further access refinements for those with **Control Panel Access** can be made in the **ACL Roles** section below.
5. After selecting **Control Panel Access**, click **Continue to FortiCloud**.
6. Create a FortiCloud user by following the steps detailed in this page .
7. After creating a FortiCloud user, go back to the main FortiCloud page.
8. Go back to FortiMonitor by clicking **FortiMonitor** from the **FortiCloud Services** menu.

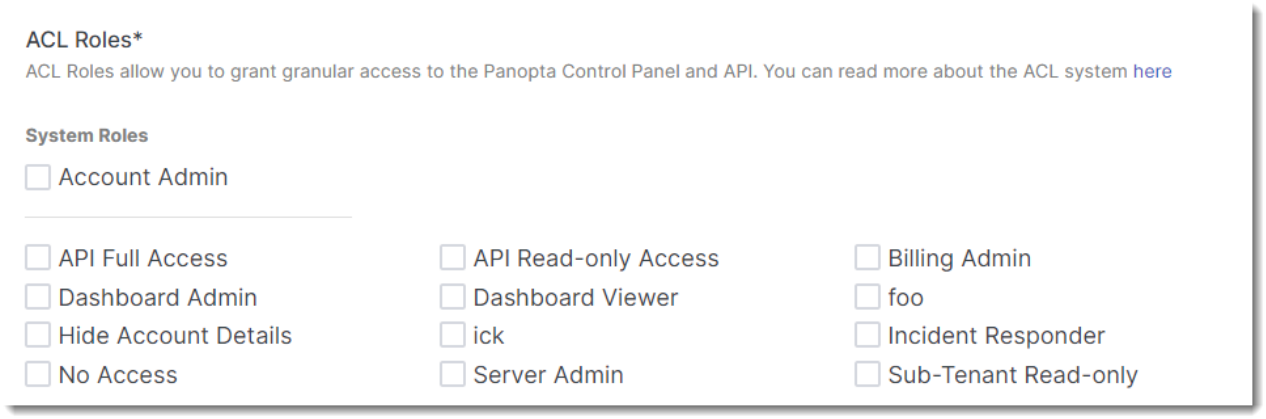


9. From the FortiMonitor control panel, go to **Teams & Activity** > **Users, Groups & On-Call**.

10. Click the **Sync Fortinet Users** button. This will add the created users to FortiMonitor.
Note: A discovery process is ran nightly to sync users created in FortiCloud to FortiMonitor. Use the **Sync Fortinet Users** button if you want to sync the created users immediately.



11. Users created in FortiCloud will have the **No Access** ACL role enabled by default. An Administrator must set the permissions in FortiMonitor. For more information, see Access control.



Add an Alert Only user

Alert Only users are restricted users that are only allowed to receive alerts.

To create an Alert Only user, perform the following steps:

1. From the navigation menu, select **Teams & Activity> Users, Groups & On-Call**.
2. Click **Add** then **Add User** .
3. Enter the user's email address.
4. Select Alerting Only.
5. Enter the required information.

6.

Field	Description
Name	The user's name, as it will appear throughout the control panel and in communications.
Timezone	The user's timezone. All displayed and computed times are adjusted for the user (this does not affect other users on the account).
Avatar	The user's avatar.

Field	Description
Alerting	
	The **Alerting** pane allows you to add various contact methods for the user, which are then later used in Alert Timelines. However, only email is required. You can also add labels (optional) to each contact information.

7. For each email address and phone number provider, you can set which alert types that specific contact method receives.
8. Select **Save**. The new user is added to your FortiMonitor account. Note that this user will only exist in the region where it was created.

Access control

FortiMonitor's access control (ACL) functionality gives teams fine-grained controls over the infrastructure team members can view (scope) and specific actions they can perform. The system is driven by **roles**, which are logical groupings of actions that a user can perform. A user can have one-to-many roles assigned to them - they're designed to be layered on top of one another.

Roles

System Roles

Out of the box, FortiMonitor provides a number of default roles that will be more than sufficient for a lot of teams. They are not editable, but you can easily clone and customize them to meet your needs.

Role	Description
Account Admin	Able to perform any activity within FortiMonitor
Server Admin	Able to perform most activities, with the exception of user, integration, and API management, as well as a few other ancillary activities
Dashboard Admin	Full management of dashboards
Dashboard Viewer	Read-only access to dashboards
Incident Responder	Slightly more advanced than a read-only user. Allows the user to view instances, start maintenance, pause monitoring, and a bit more.
Billing Admin	Access to billing only

Add Custom Roles

Teams looking for more fine tuning can leverage custom roles. Custom roles can be comprised of any number of actions that are available in FortiMonitor. For instance, you may want a role that allows a user to view and edit instances, but read-only access for network devices. When creating a custom role, you may choose to either start from scratch or clone an existing role.

Starting from scratch

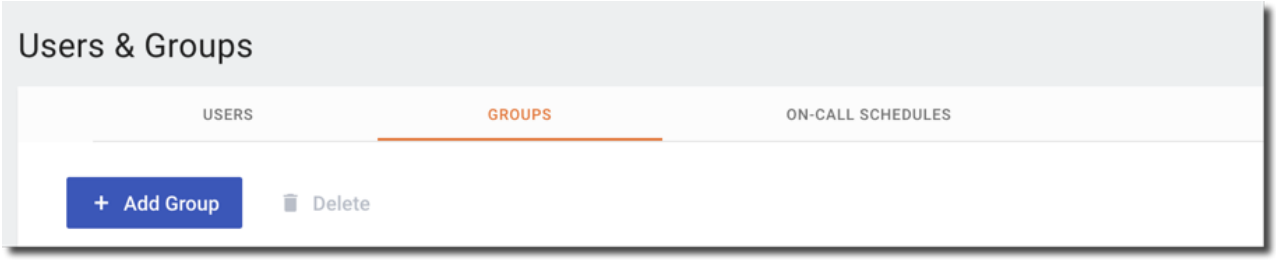
1. Navigate to **Settings > Access Control**.
2. Click **Add Role**.
3. Name your role and give it a description. Providing a description is not required but is highly recommended
4. Select any number of actions which your role can perform. Keep in mind, some actions are depended on each other - for example, it doesn't make sense to give someone edit access on an instance but not view access.
5. Click **Create**. Your role is now ready to use.

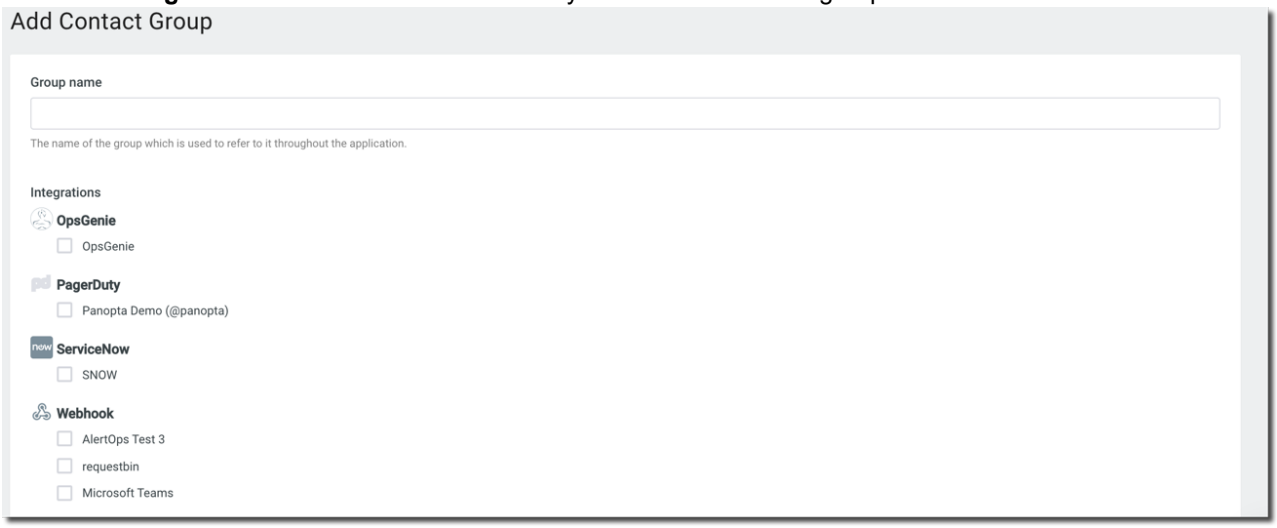
Cloning an existing role

1. Navigate to **Settings > Access Control**.
2. Locate the role you'd like to clone. On the right side of the table, click the corresponding **3-dot menu**.
3. Click **Clone**.
4. Give your new role a name. If you're a reseller, provide a role scope.
5. Customize your role to your liking.
6. Click **Create**. Your role is now ready to use.

Create a Group

To create a Group, perform the following steps:

1. From the navigation menu, select **Teams & Activity> Users, Groups & On-Call**.
   Click the **Groups** tab.
2. 

3. Click**Add Group**.
   Select the **Integrations** and user contact channels you'd like added to the group.
4. 

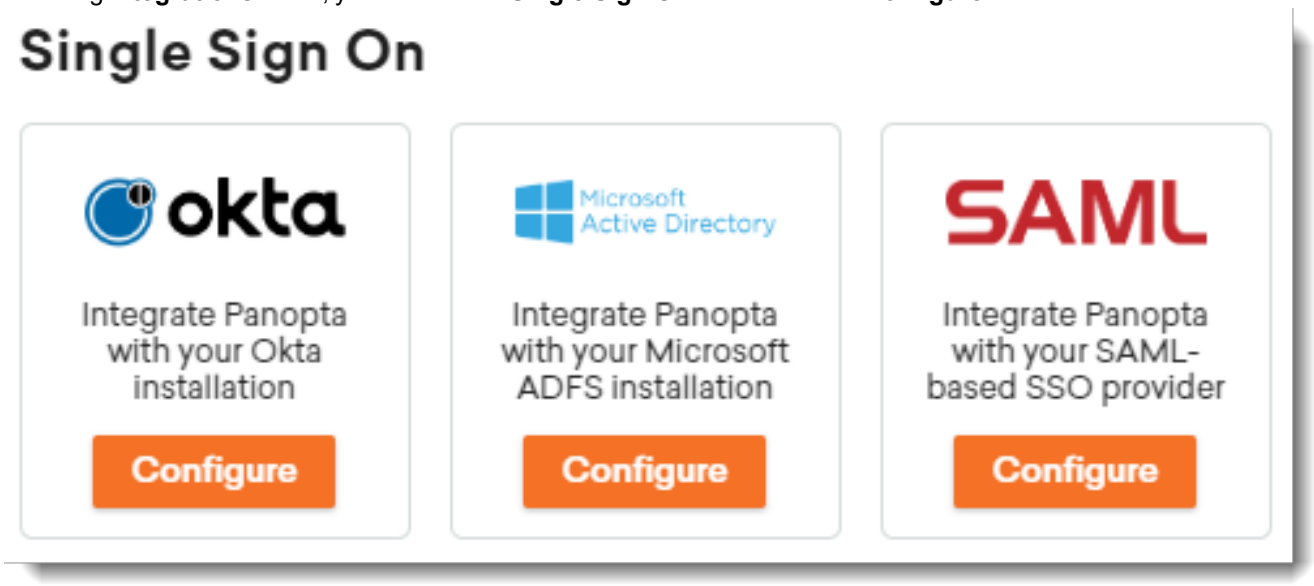
5. Click **Create** to create the group.

Single sign-on (SSO)

SSO allows your organization to utilize it's internal authentication tool to authenticate with and log in to FortiMonitor. This guide will walk you through integrating with a generic SSO provider, such as Simple SAML. You can also find docs specifically for ADFS and Okta.

Control Panel Configuration

Navigate to the **Integrations** page by selecting **Teams & Activity** from the global navigation bar, followed by selecting **Integrations**. Here, you will find the **Single Sign On** section. Select **Configure** on the SAML card.



General

Field	Description
URL Fragment	Customer-provided string that will determine the login URL for your FortiMonitor account, the format of `my.{region}.fortimonitor.com/sso/{url fragment}` For example, if you were to enter `fortimonitor` and you are in the US01 region, your login URL would be `my.us01.fortimonitor.com/sso/fortimonitor`.
Domains	Register domains tied to an SSO configuration to allow first time users to log in.
Username	Field in your SAML payload that matches a user's FortiMonitor login email. This is an email for most customers.
Entity ID	URL that provides your IDP metadata.
Login URL	The URL we redirect the user to when the user arrives at your FortiMonitor SSO login.

	For instance, if a user visits your FortiMonitor SSO URL which is built in the `URL Fragment` configured in step 1, the `Login URL` is the address we would then redirect them to where they authenticate with your SSO tool
Login Binding	This is a colon separated sequence of strings. For example, `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect`. It is generated by your SSO tool.
Logout URL (optional)	URL to redirect the user upon logout request.
Logout Binding	This is a colon separated sequence of strings. For example, `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect`. It is generated by your SSO tool.

Certificate

A valid x509 certificate. Ensure it is copied with out extra whitespace.

User Configuration

Field	Description
Prevent non-SSO logins	Enabling this option prevents users from logging in to FortiMonitor with their email address and password. This setting can be overwritten at the account level. To override the setting, see Mixing login types. **Note**: Leave this box unchecked until you have confirmed that your SSO configuration is working properly.
New User Notifications	Select the user account to notify when an SSO user signs on for the first time to FortiMonitor.
Auto Create Users	Leave this box unchecked if you want to require admin approval before a user can use FortiMonitor. Once a user logs in for the first time, those selected in the `User Emails` select list will receive a notification email. They can follow the link in the email to grant the new user access. Until they do this, the user will not be able to access FortiMonitor - they will merely see a splash screen when they log in. You can view all users who are waiting for approval under the "Pending" users tab on the Users and Groups page. If you prefer to let users immediately begin using FortiMonitor upon login, check this box.
Default Roles for New Users	There are two options when the **Auto Create Users** option is enabled: • **Assign roles manually:** If you're automatically creating the user the first time they log in via your SSO integration, they can optionally be assigned any number of roles by default. • **SSO-based roles (Assign roles based on SAML mapping):** If you're sending your internal roles in your SAML payload, you can map

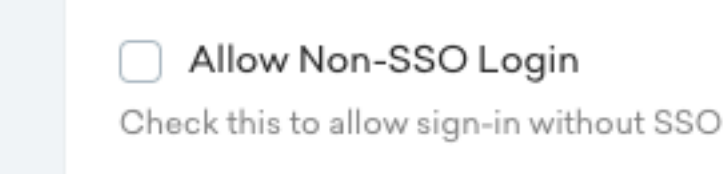
those to specific roles in FortiMonitor.

- In the `SAML Role Field` provide the payload key that corresponds to your internal roles in your SAML payload.
- In the `SAML Role`, enter the internal role you'd like to target. Please only enter one role
- In the `FortiMonitor roles to assign` dropdown, select the roles you'd like the user to have in FortiMonitor

You can create as many mappings as needed.

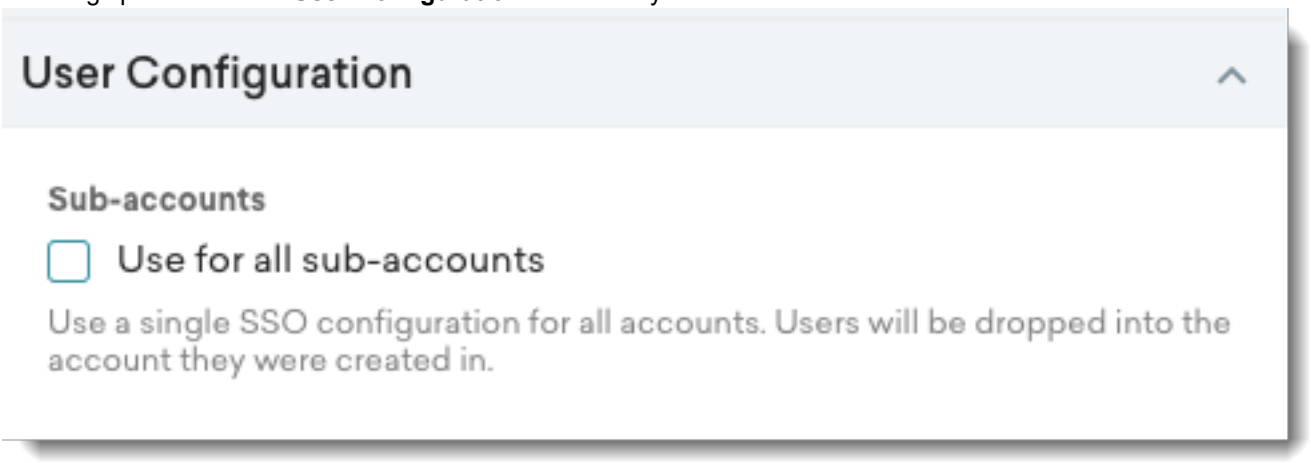
Mixing login types

Sometimes it is valuable to allow non-SSO users to still log in to your company's FortiMonitor account - especially if you leverage outside resources. To allow certain users to still login via email and password, check the "Allow Non-SSO Login" checkbox. This option can be found by editing the user - go to `Settings > Users, Groups & On-Call`, then edit the desired user. The option is on the first pane.



SSO configuration for Multi-tenant users

If you are a multi-tenant user, you have the option to apply your SSO configuration to all sub-accounts. You can find the following option under the **User Configuration** module on your master tenant account:



Select the **Use for all sub-accounts** option to use a single SSO configuration for all your sub-accounts/tenants. Settings configured in the master tenant account such as **New User notification**, **Default Timezone**, and **Auto Create Users** are also applied to all other tenants. All created users are automatically added to the correct tenant.

Set up SSO Configuration for SAML + ADFS 3.0

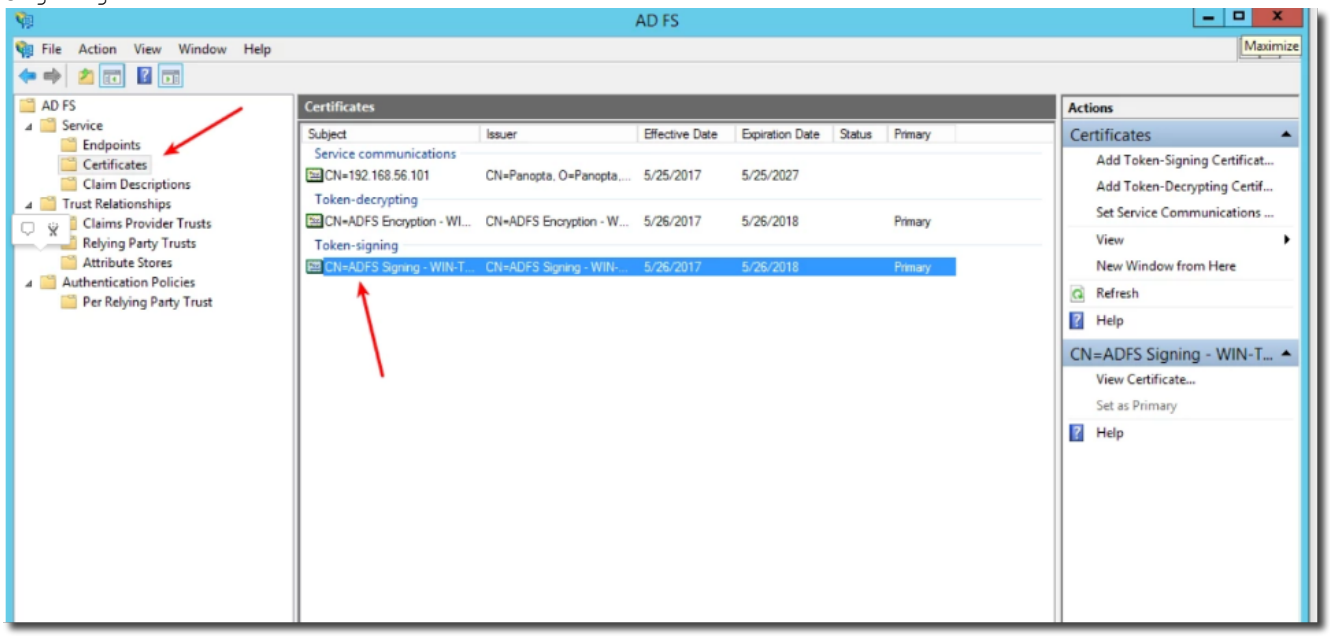
ADFS Configuration for SSO involves a couple extra steps compared to our regular SAML configuration. This article will guide you through the process.

Versions:

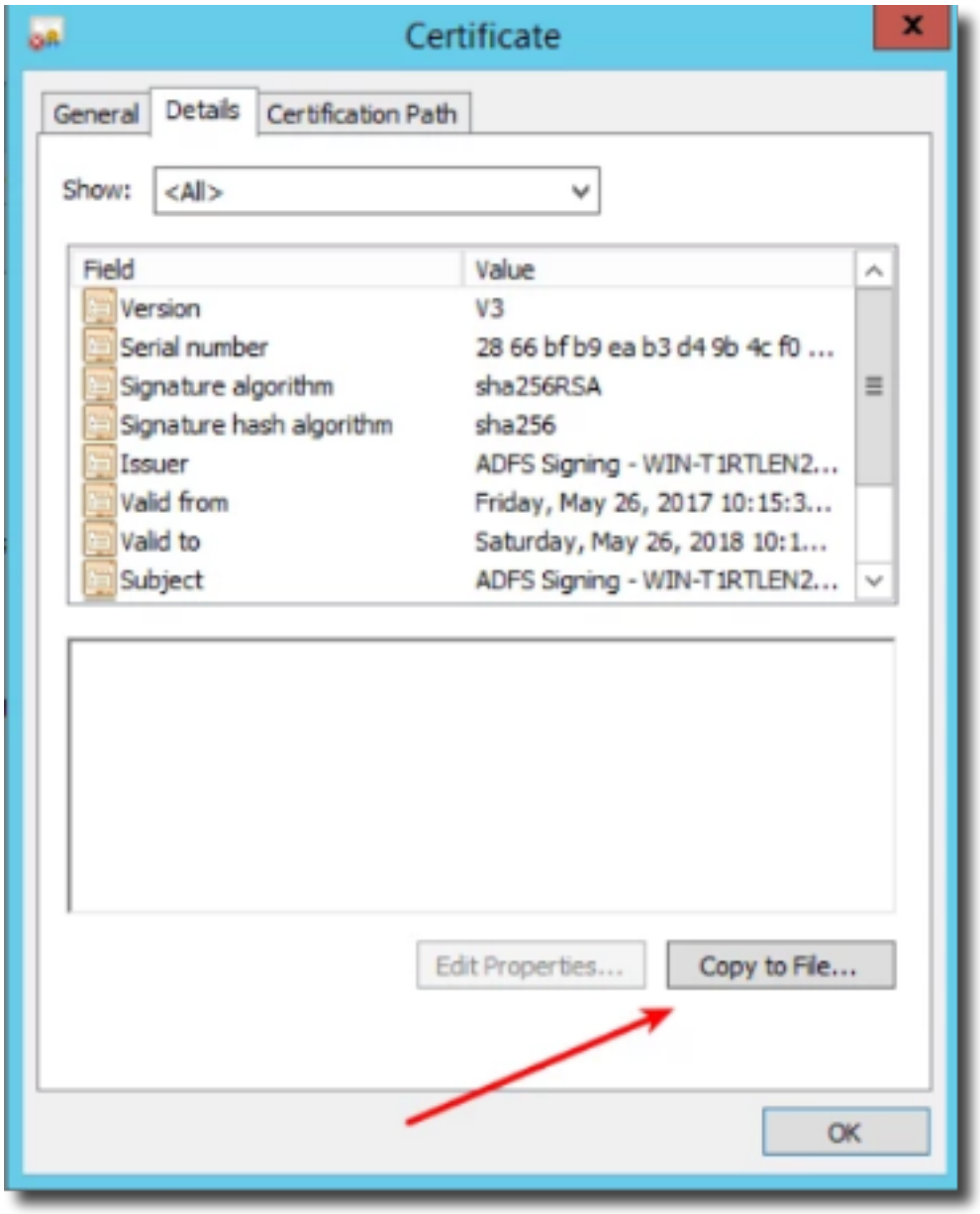
- Windows Server 2012
- ADFS 3.0

Obtaining the cert

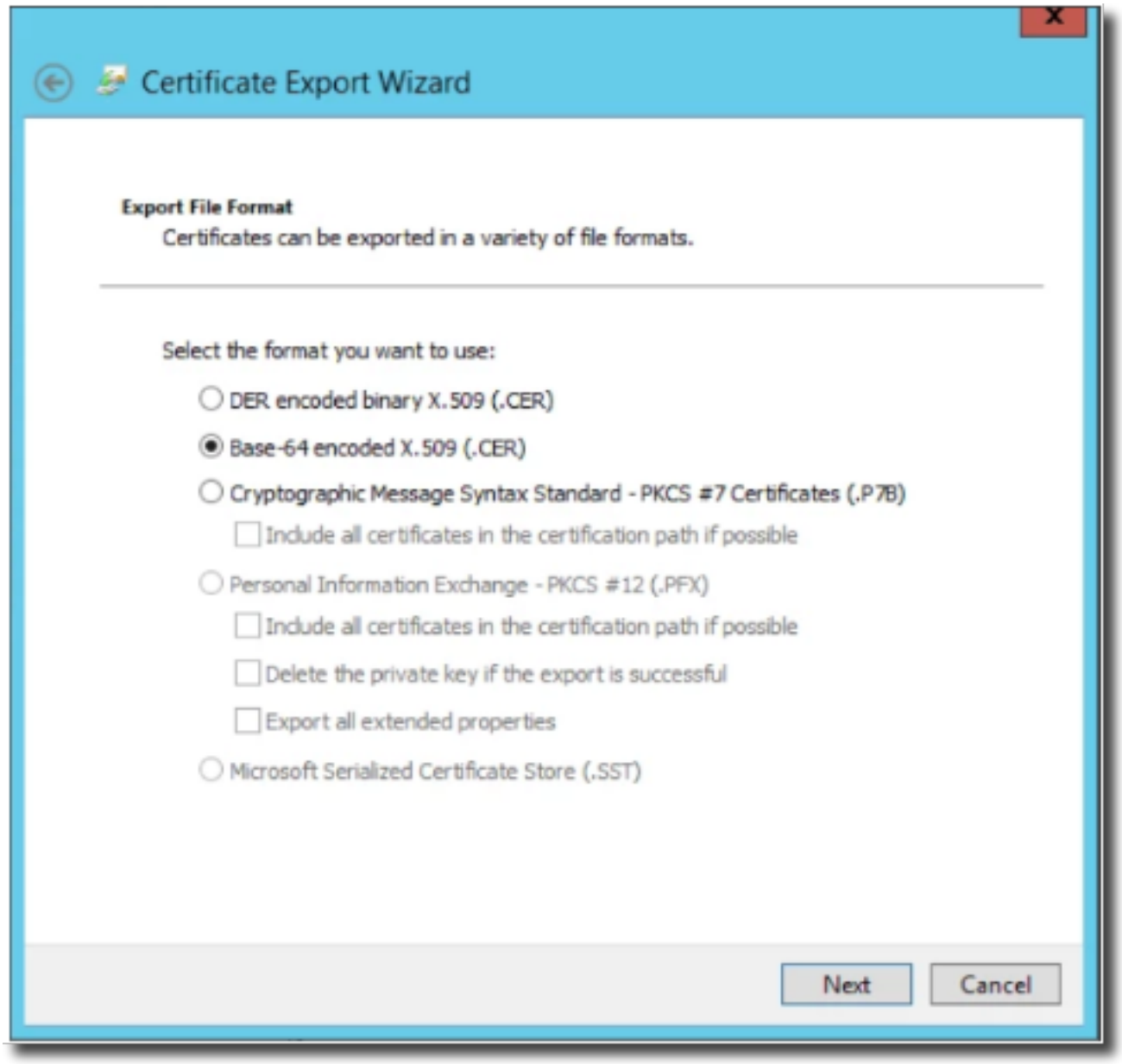
Open the ADFS Management tool. On the left, expand `AD FS` and `Service`; then, double click on the `Token-signing` cert.



Under the detail tab, click `Copy to File`.



Export the cert in the Base-64 Encoded X.509 (.CER)



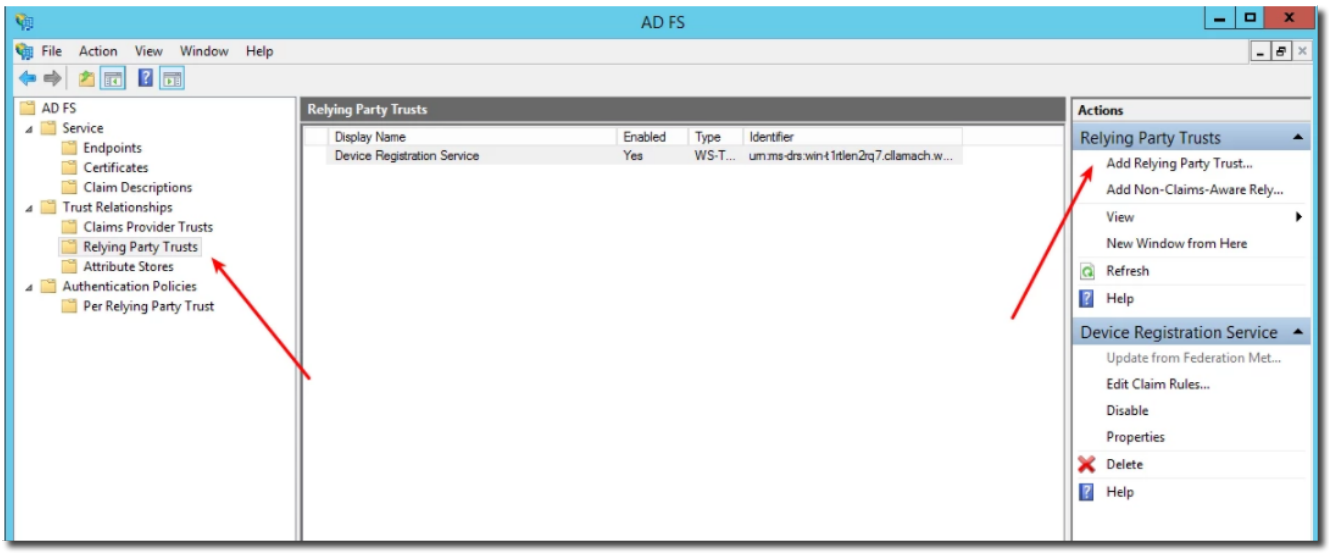
Open the FortiMonitor control panel and add the EntityID, Login URL and Login Binding to your SSO configuration - you can learn more about that in Single sign-on (SSO). Take note of the `username` field value and the `URL fragment`, as we are going to use those fields to setup our ADFS configuration.



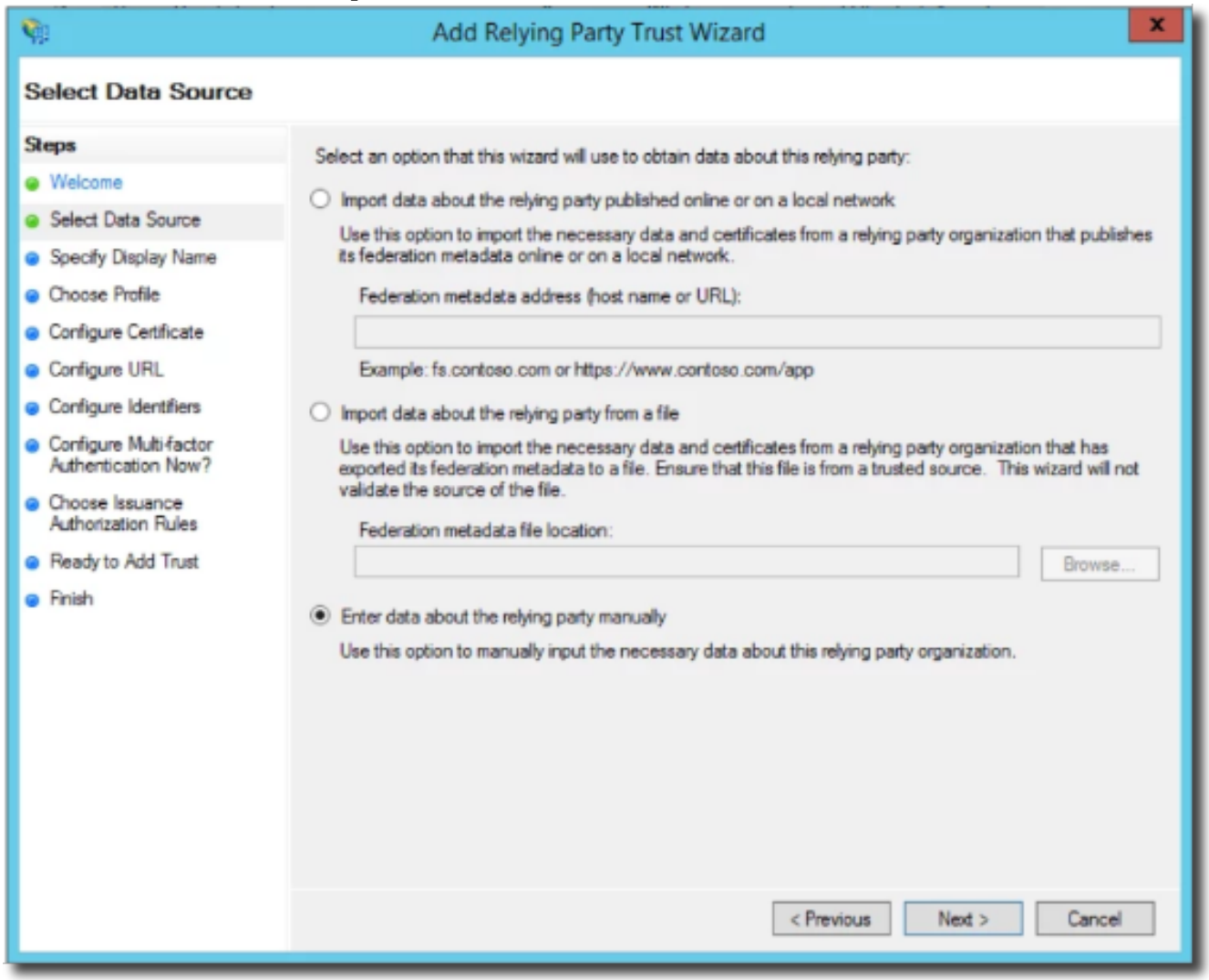
Paste the certificate we just exported into the `certificate` input, within the `Certificate` tab of the slide-out pane.

FortiMonitor as a Service Provider

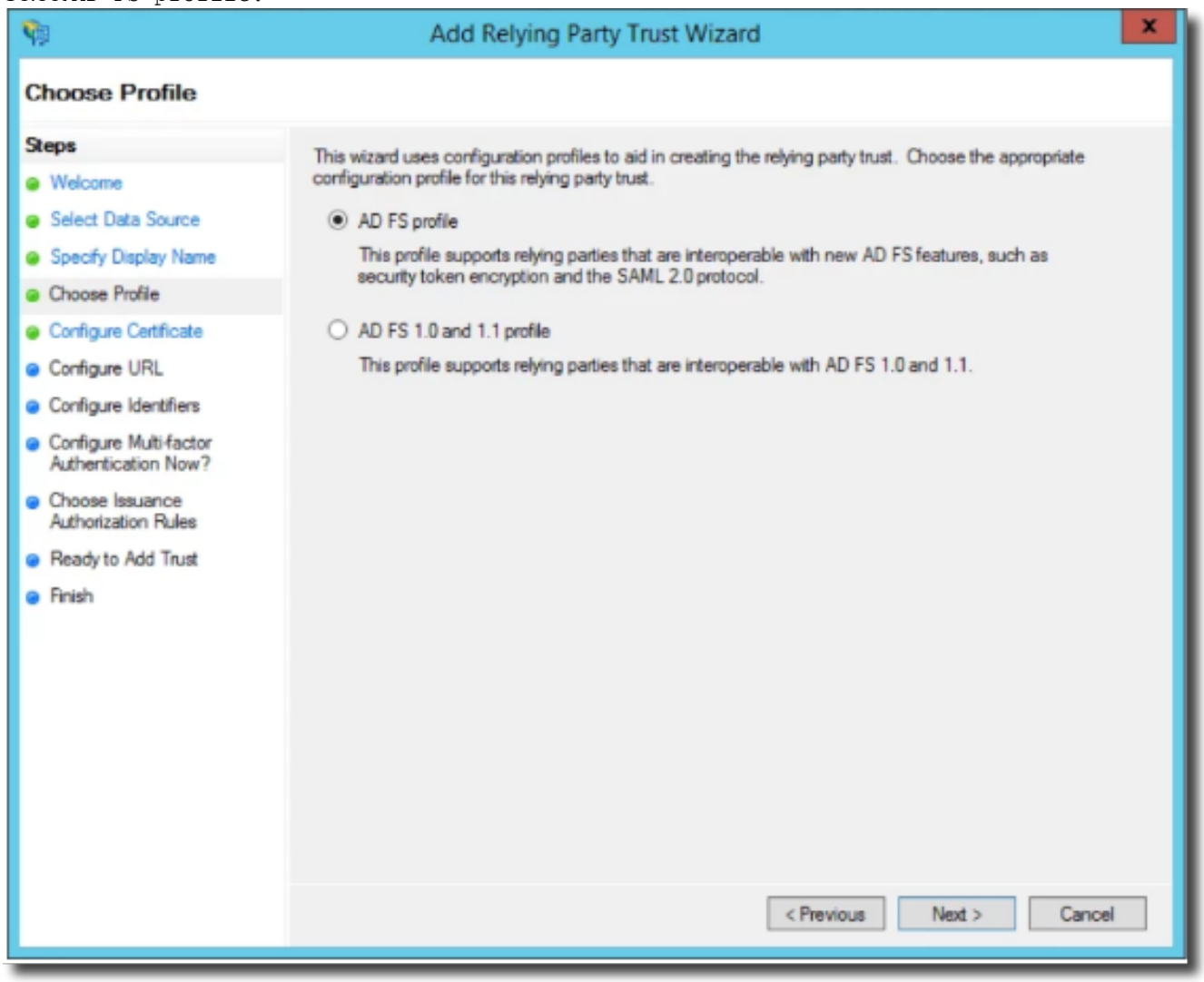
To configure FortiMonitor as a Service Provider for ADFS, first go to `Relaying Party Trusts in the Trust Relationships of the ADFS Management console and add a new Party Trust.`



Select `Enter data.. manually`.

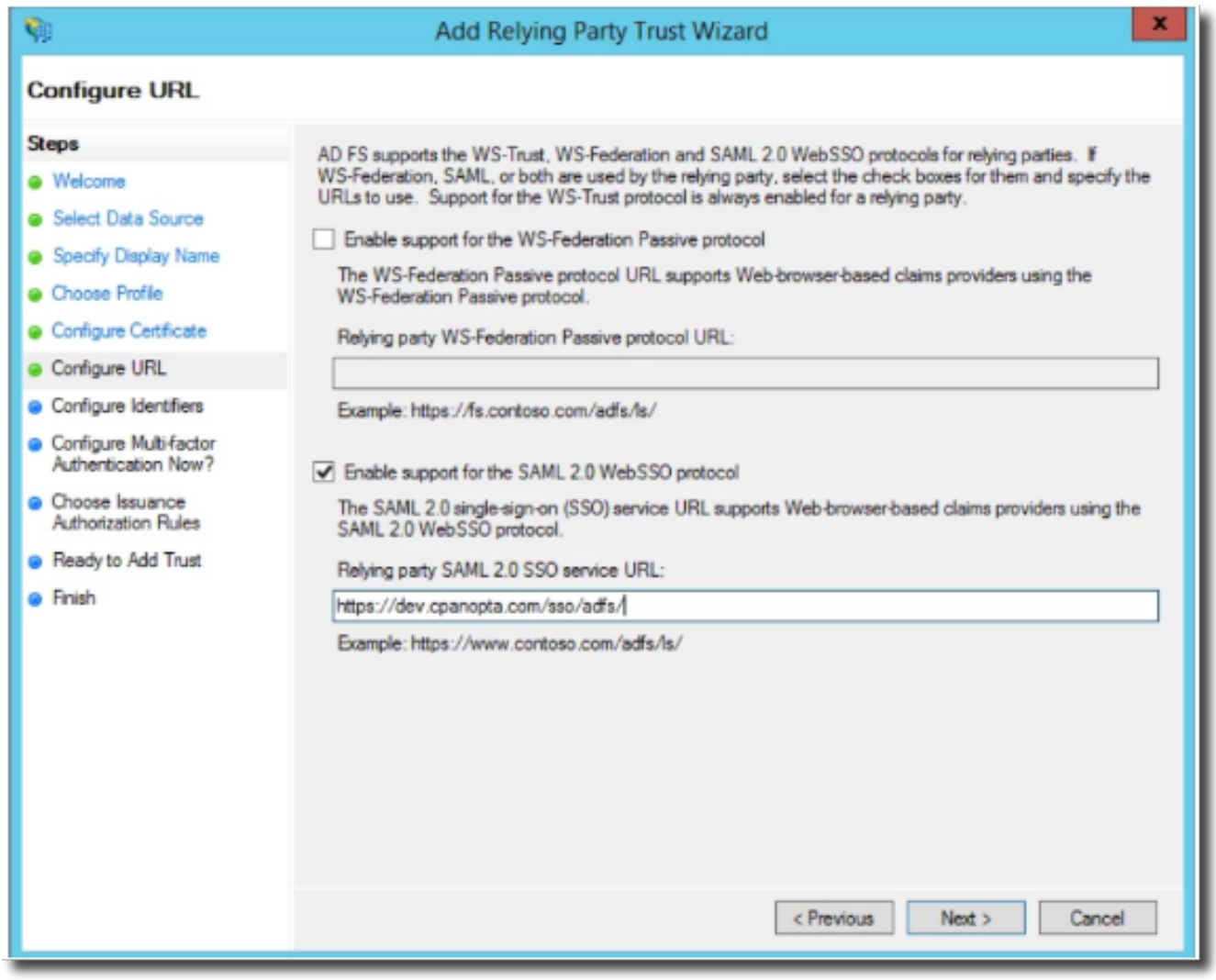


Select `AD FS profile`.



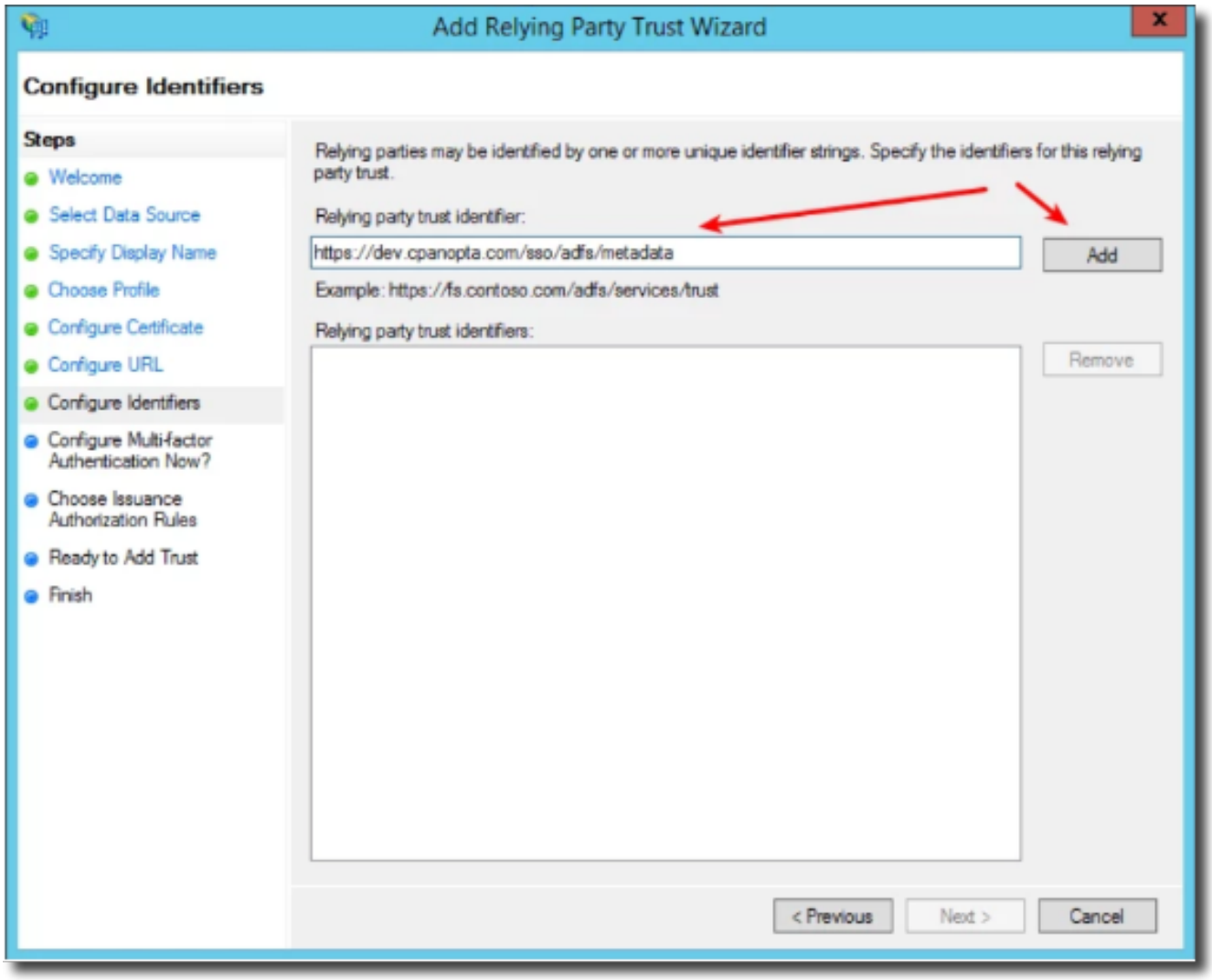
Select `Enable support for SAML 2.0 protocol`. The URL is `http://my.panopta.com/sso/{url fragment}`, where the URL fragment is the one you entered in your SSO Configuration in the FortiMonitor Control Panel.

New users should use fortimonitor.forticloud.com for the SSO URL. For long time customers of Panopta, continue using my.panopta.com.



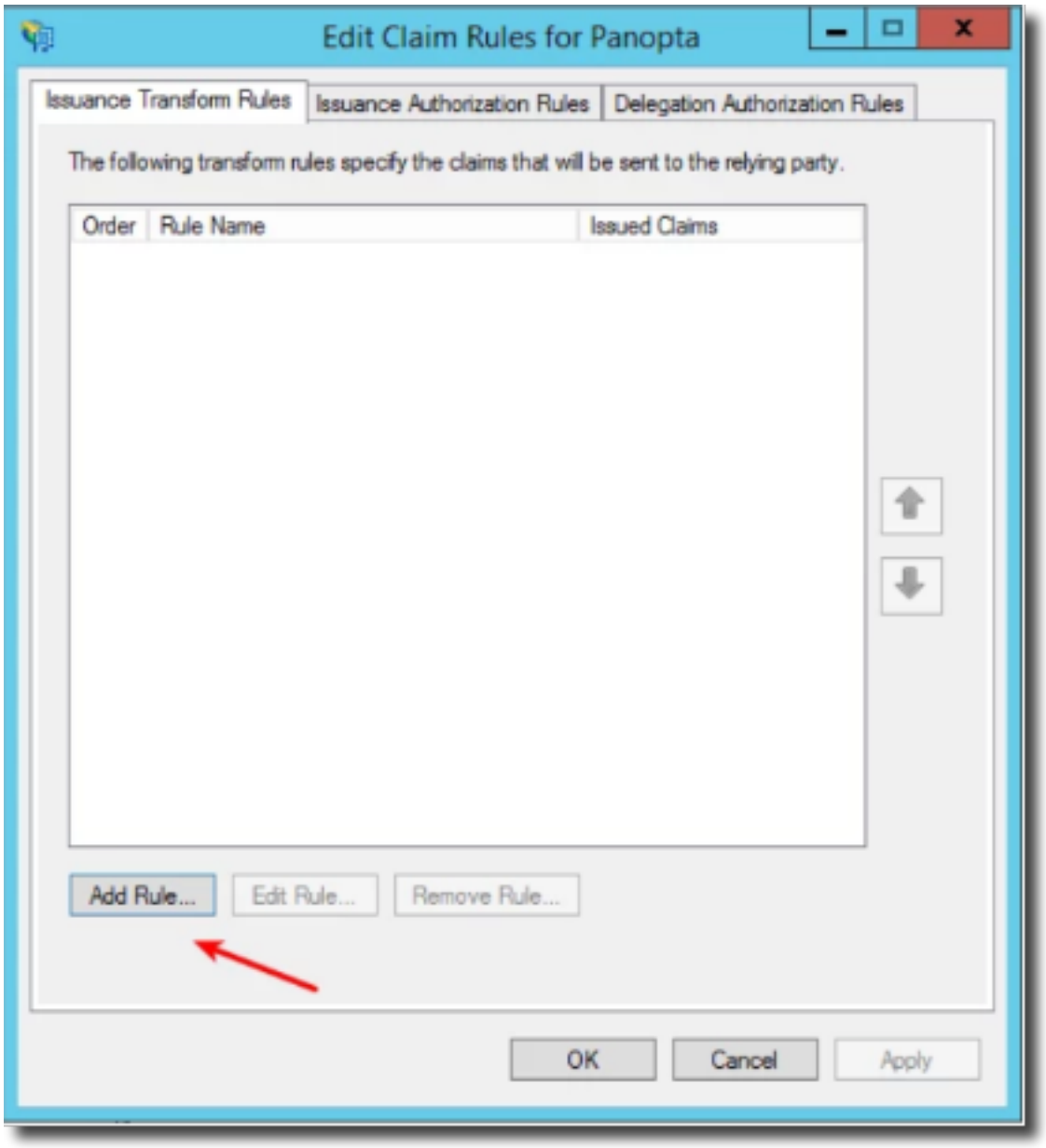
Add the `Relaying Party Trust Identifier`. The URL will be the URL you entered the previous step, with `/metadata` at the end of it, in the form of `http://my.panopta.com/sso/{url fragment}/metadata`.

New users should use fortimonitor.forticloud.com for the SSO URL. For long time customers of Panopta, continue using my.panopta.com.

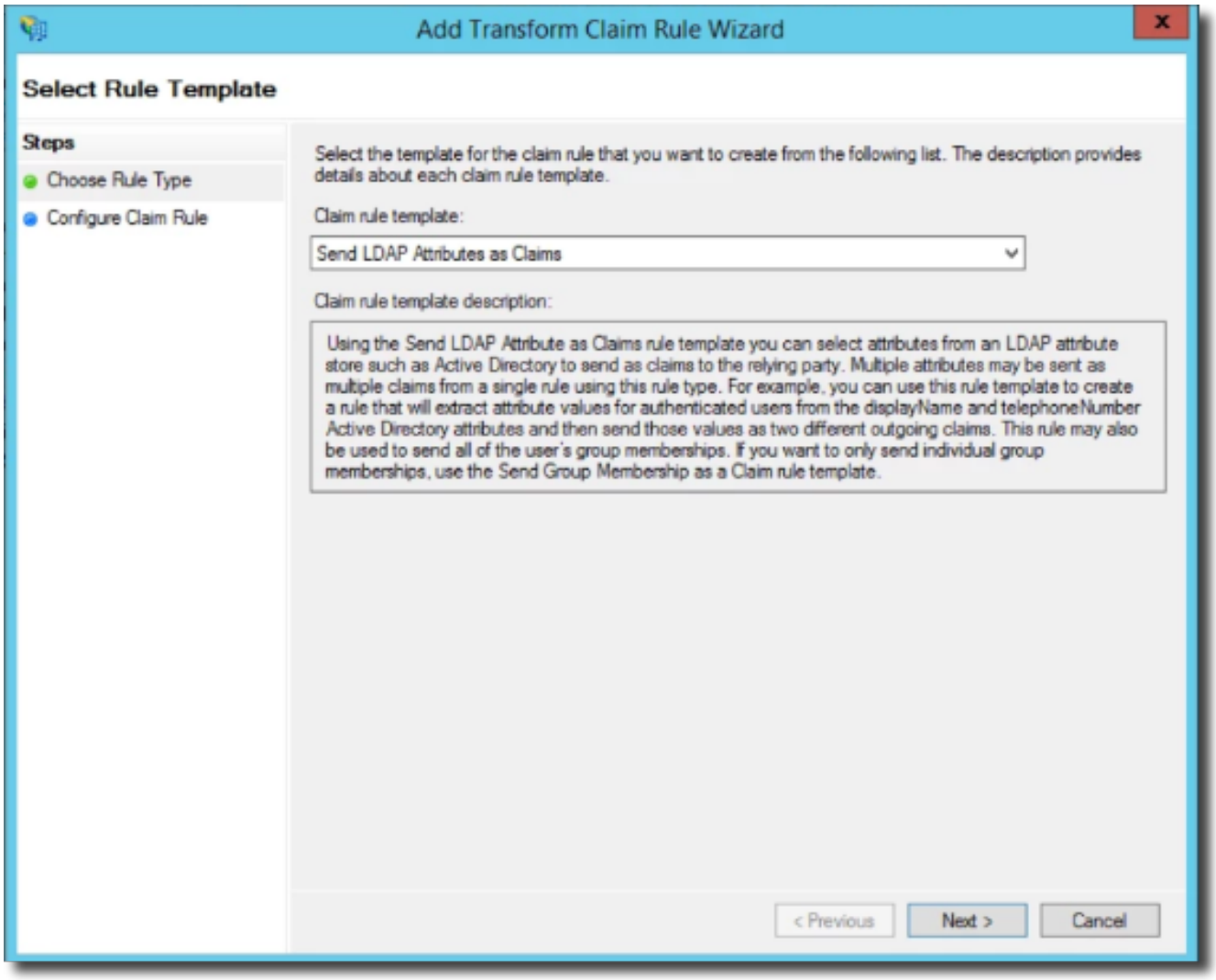


Leave multi-factor authorization off and allow all users access to the relaying party. After you finish, the modal is going to ask you to open the Claim Rules settings. Say yes. We need to configure 5 rules to send the required information to the Panopta Service Provider.

Click on `Add Rule`.

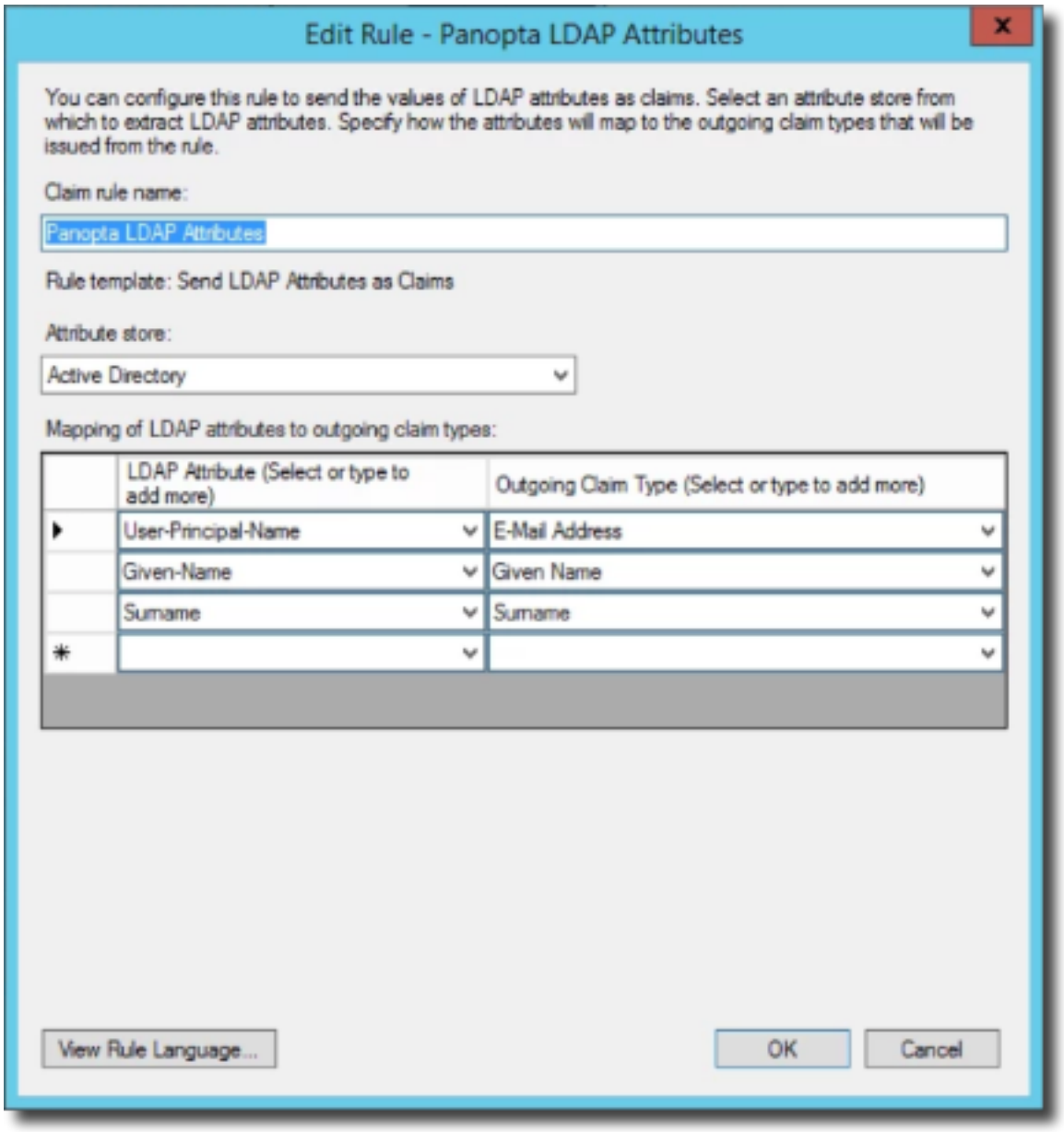


Select `Send LDAP Attributes as Claims`.

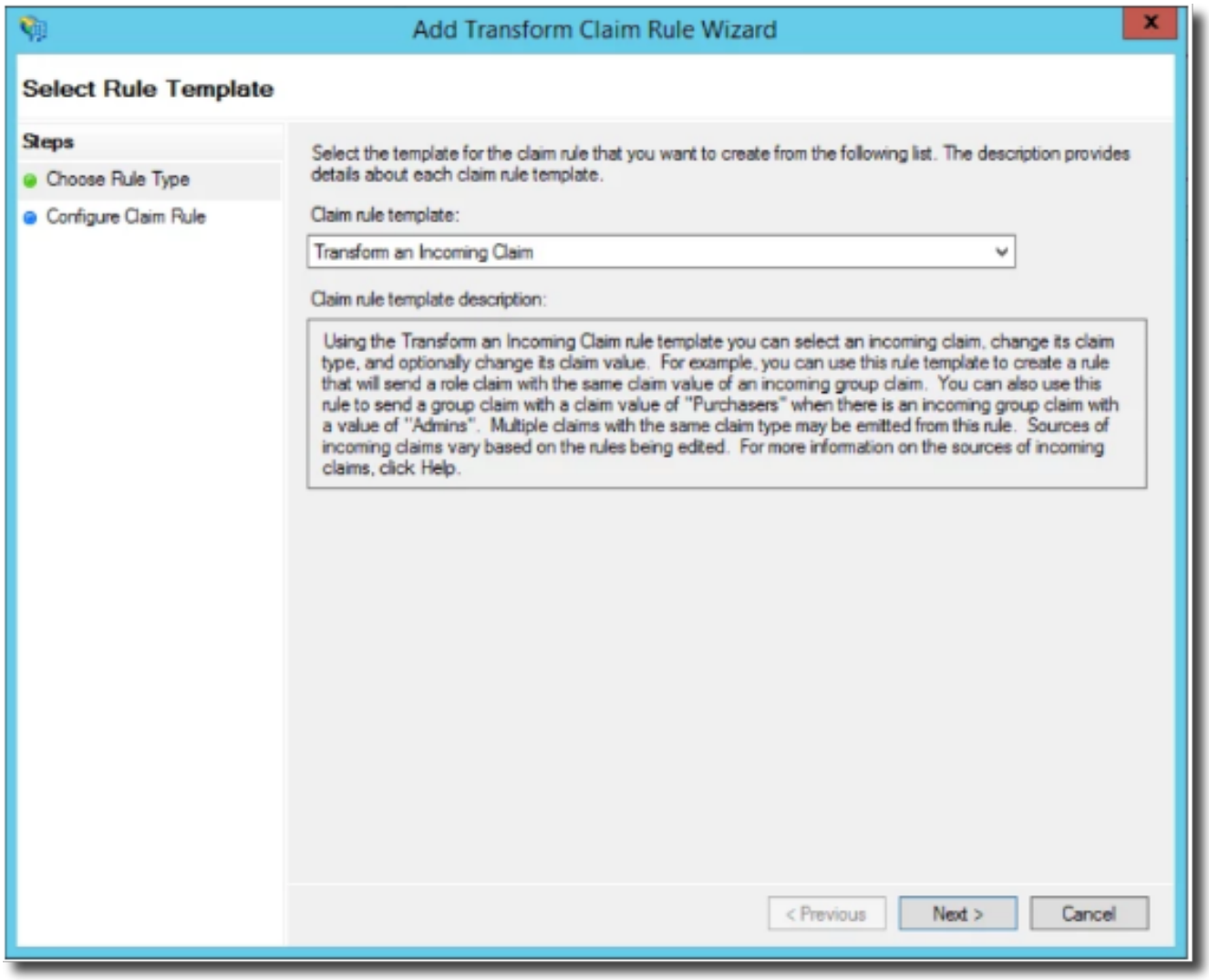


Choose a name for your rule - anything is fine. Select `Active Directory` as the `Attribute store` and ensure your rule is configured as seen in the below image.

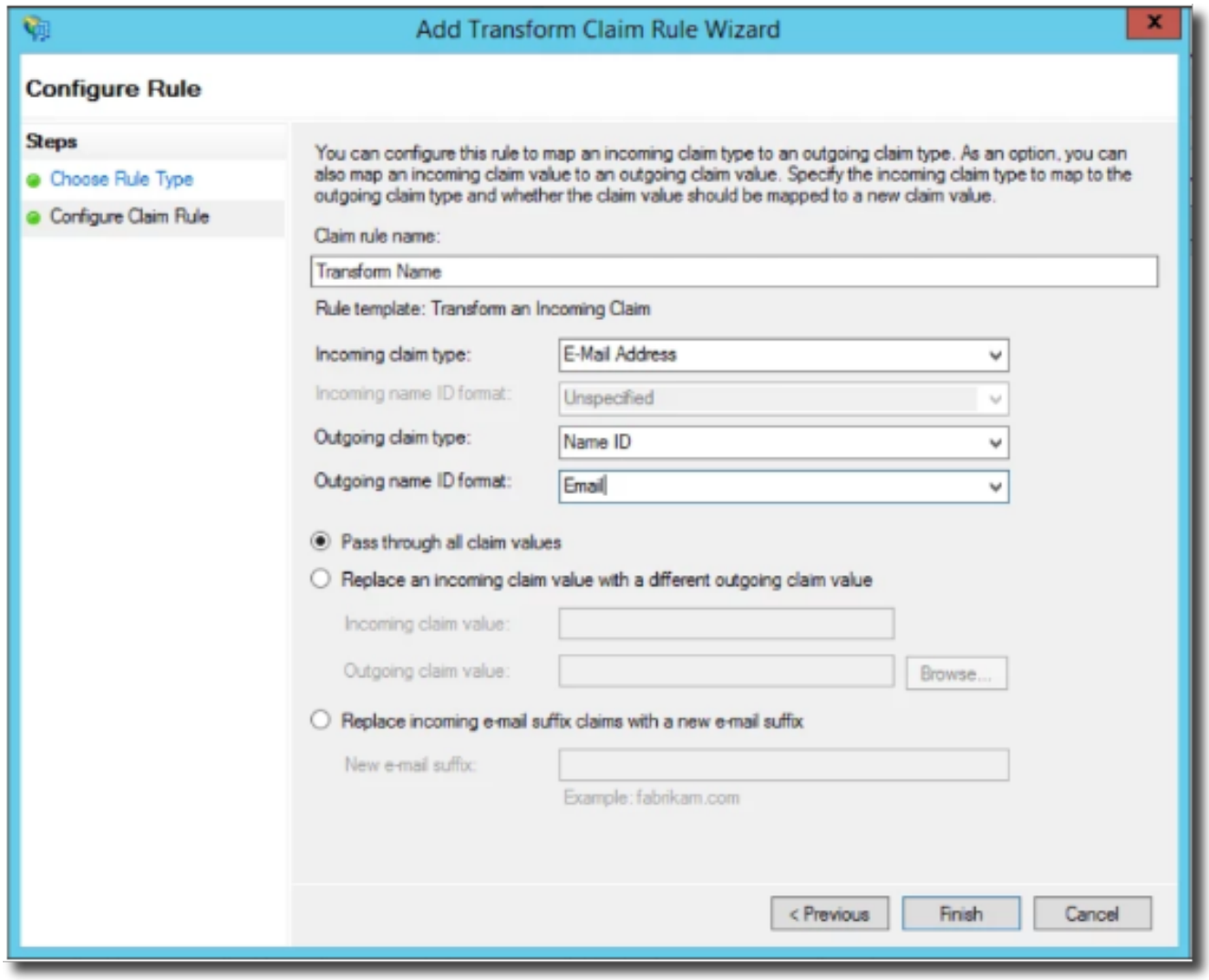
Ensure `Email Address` is correctly matched to your Active Directory.



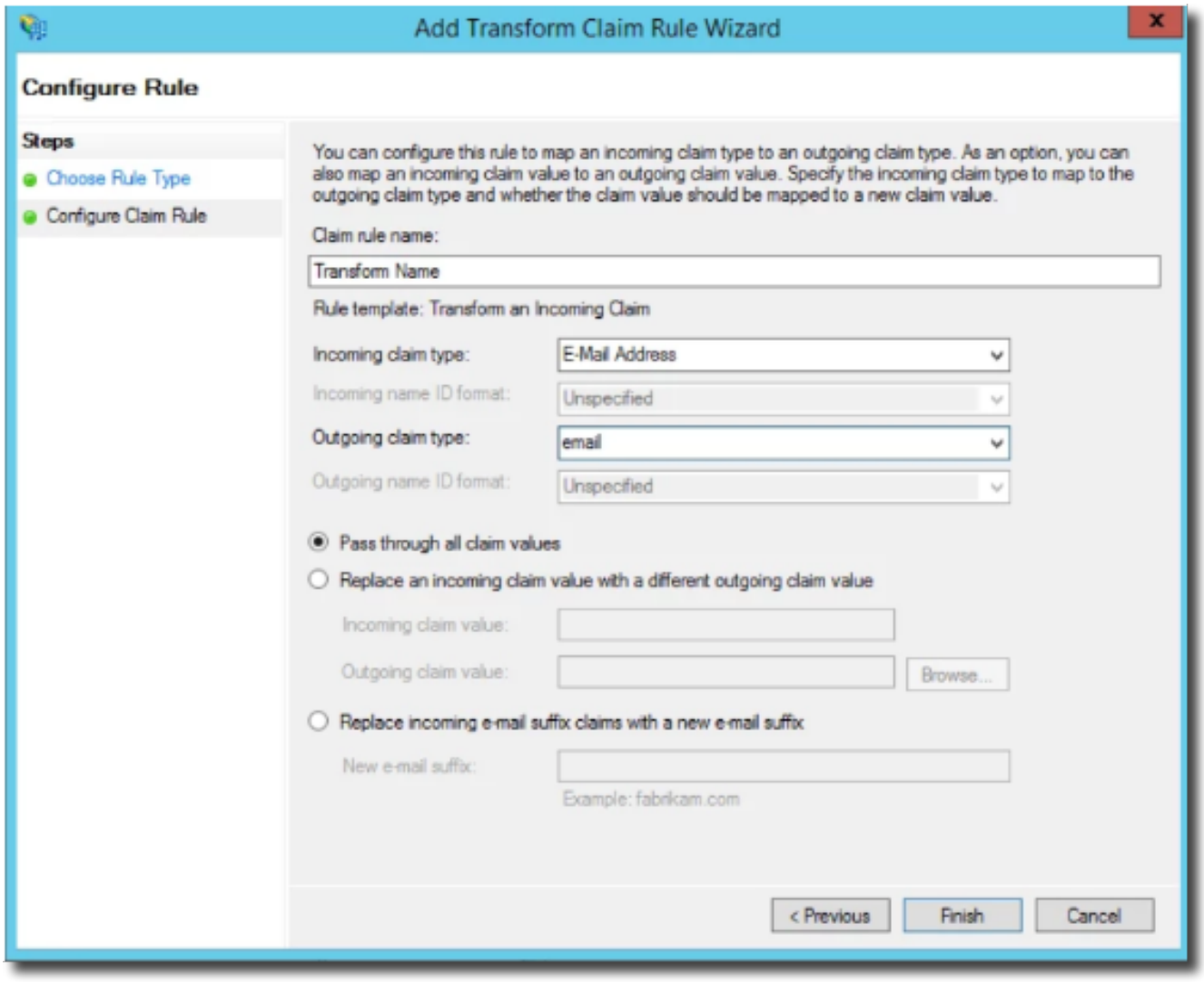
Add a second rule but this time select `Transform an Incoming Claim`.



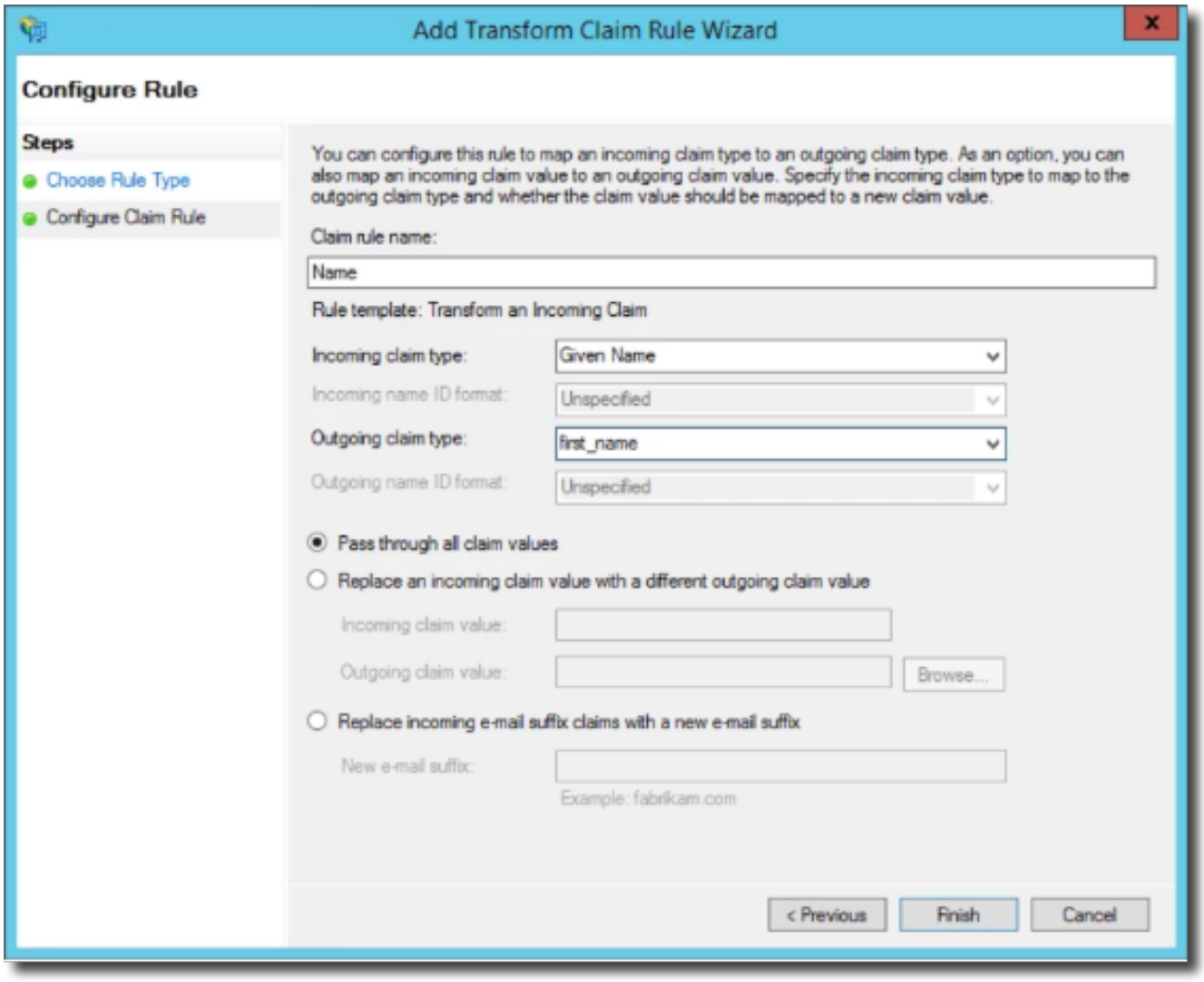
Select `Email Address as the Incoming claim type, Name ID as the outgoing claim type, and enter Email in the Outgoing Name ID format input.`



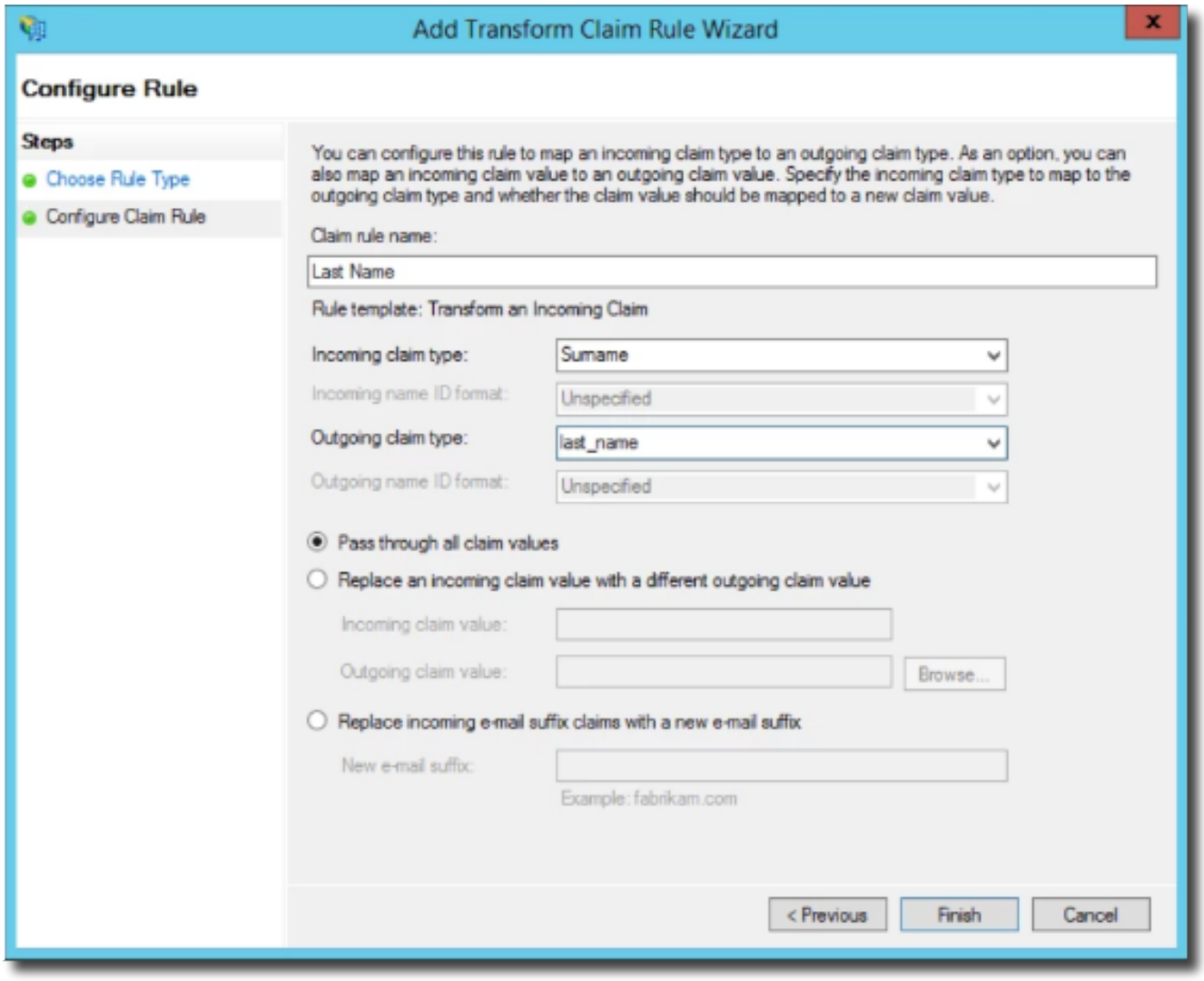
Add a second transform Rule, this one setting `Email Address as email. This email must match the username field value in our Panopta configuration.`



Add a third transform rule, setting the `Given Name as first_name`.

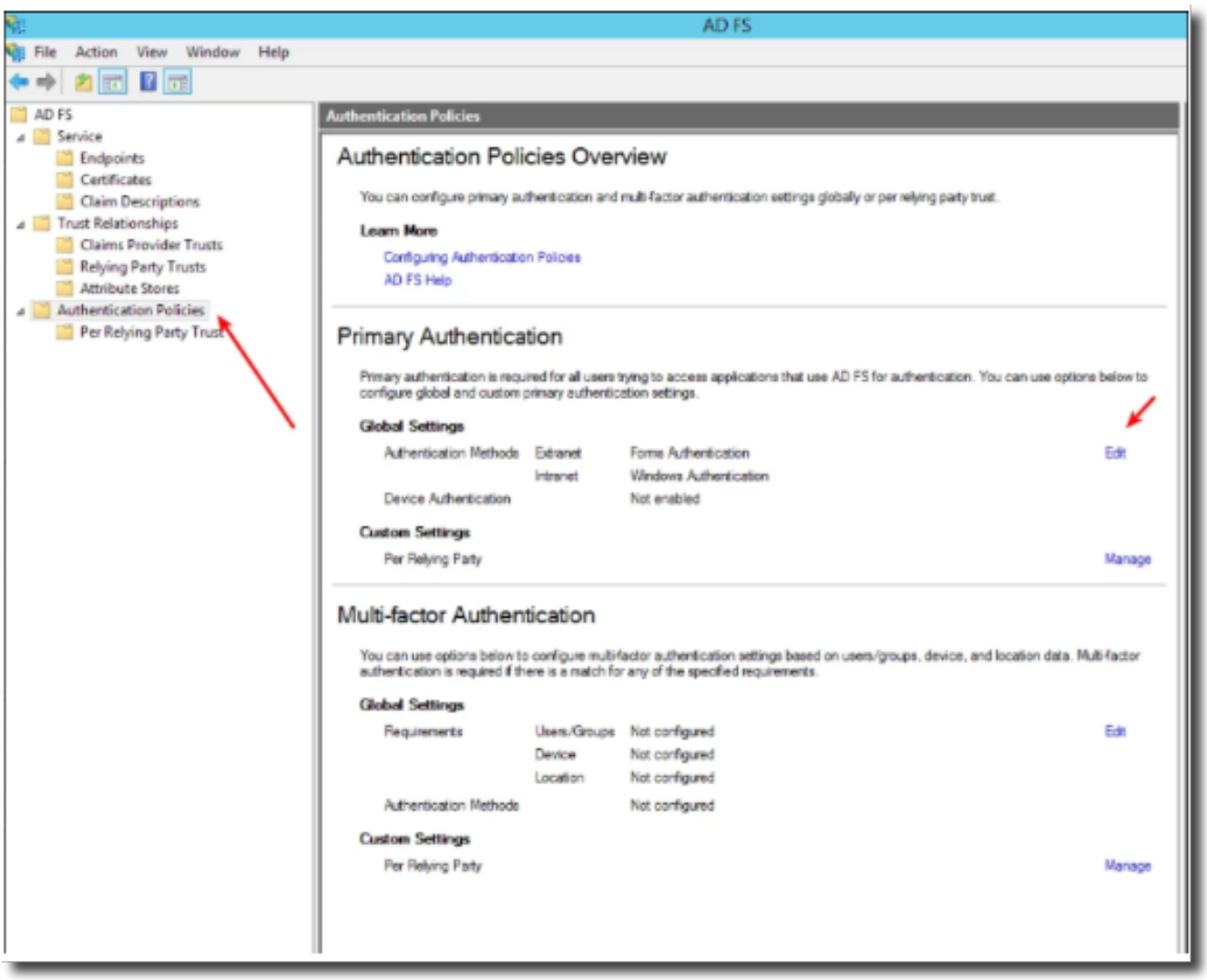


Add our final rule, setting the `Surname as last_name`.

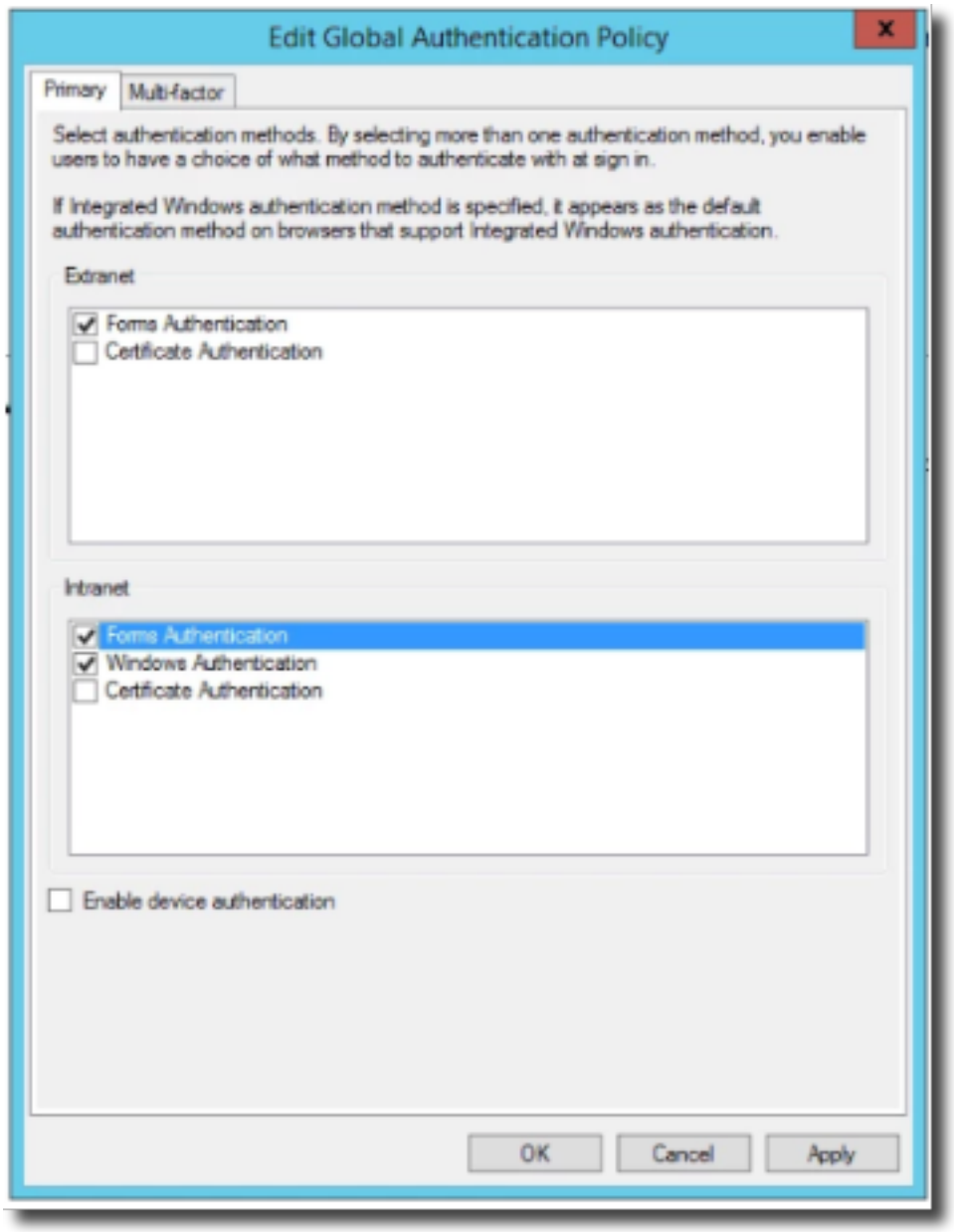


Next, ensure you have `Form Authentication set in the Authentication Policies.`

Go to Authentication Policies in ADFS Management and click **edit**.



If you are using testing or an intranet, enable `Form Authentication`.



Restart your ADFS Service after all the changes are made.

Go to your Panopta SSO login URL and you should be greeted with ADFS login screen.

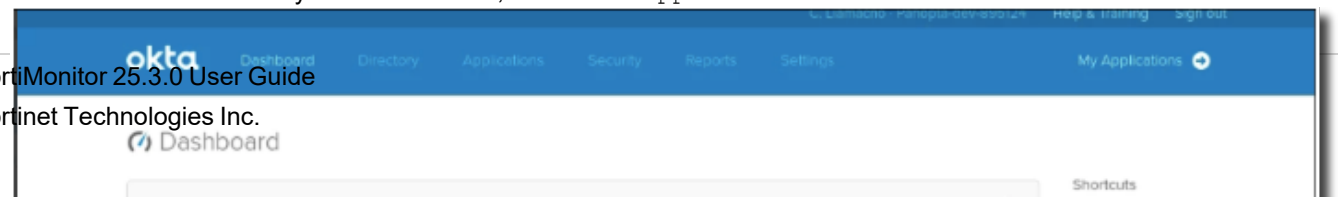
Enter your credentials and sign in - you should then be redirected to Panopta.

Set up SSO with Okta

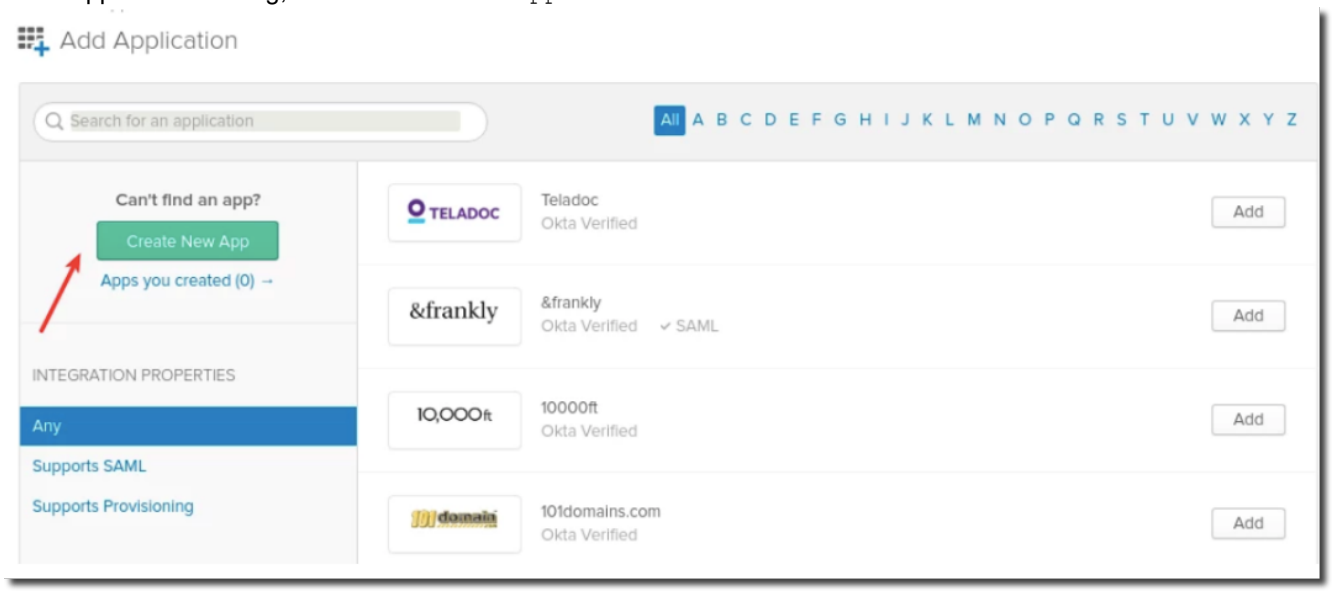
Before getting started, see SSO Configuration to learn more about our general SSO settings.

Okta Configuration

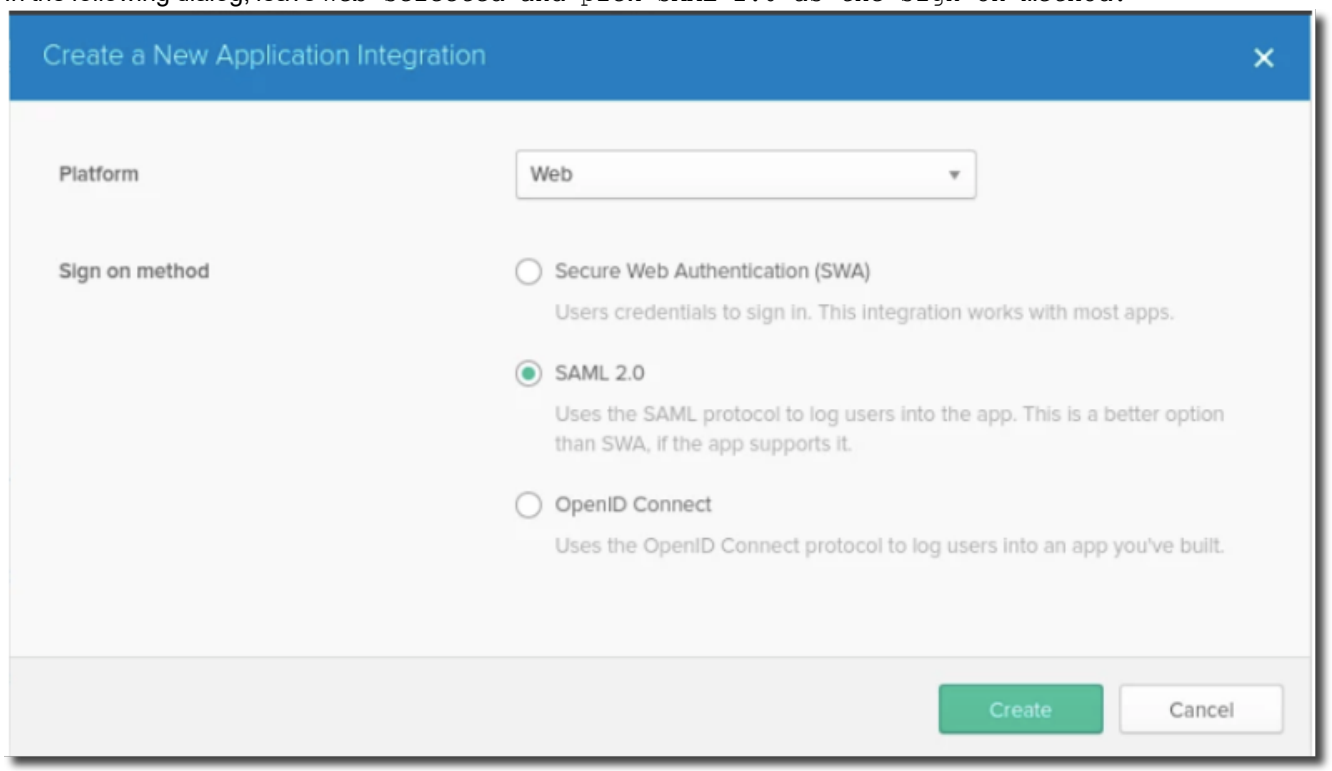
In the main Dashboard of your Okta account, select `Add Applications`.



In the application catalog, click `Create New App`.

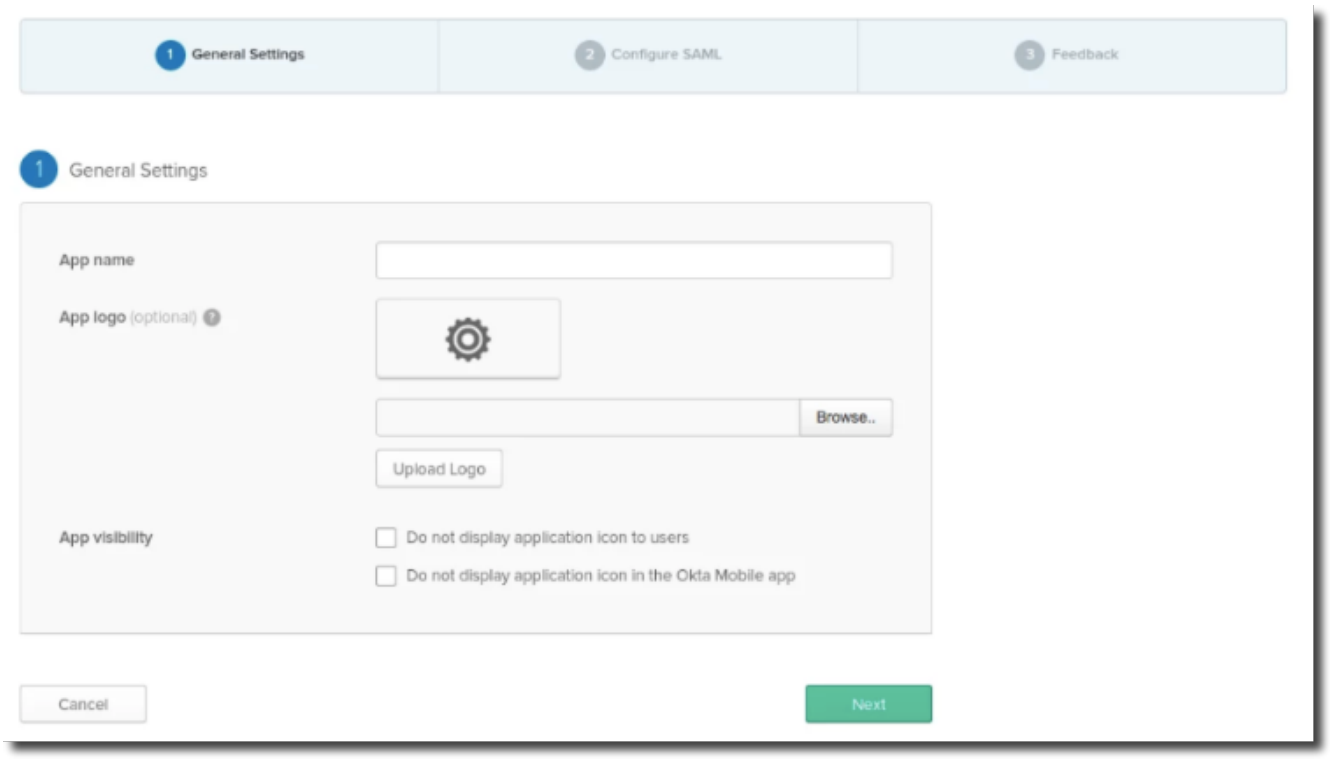


In the following dialog, leave `Web selected and pick SAML 2.0 as the sign on method.`



General Settings

In `General Settings`, provide, at minimum, a name.



SAML Configuration

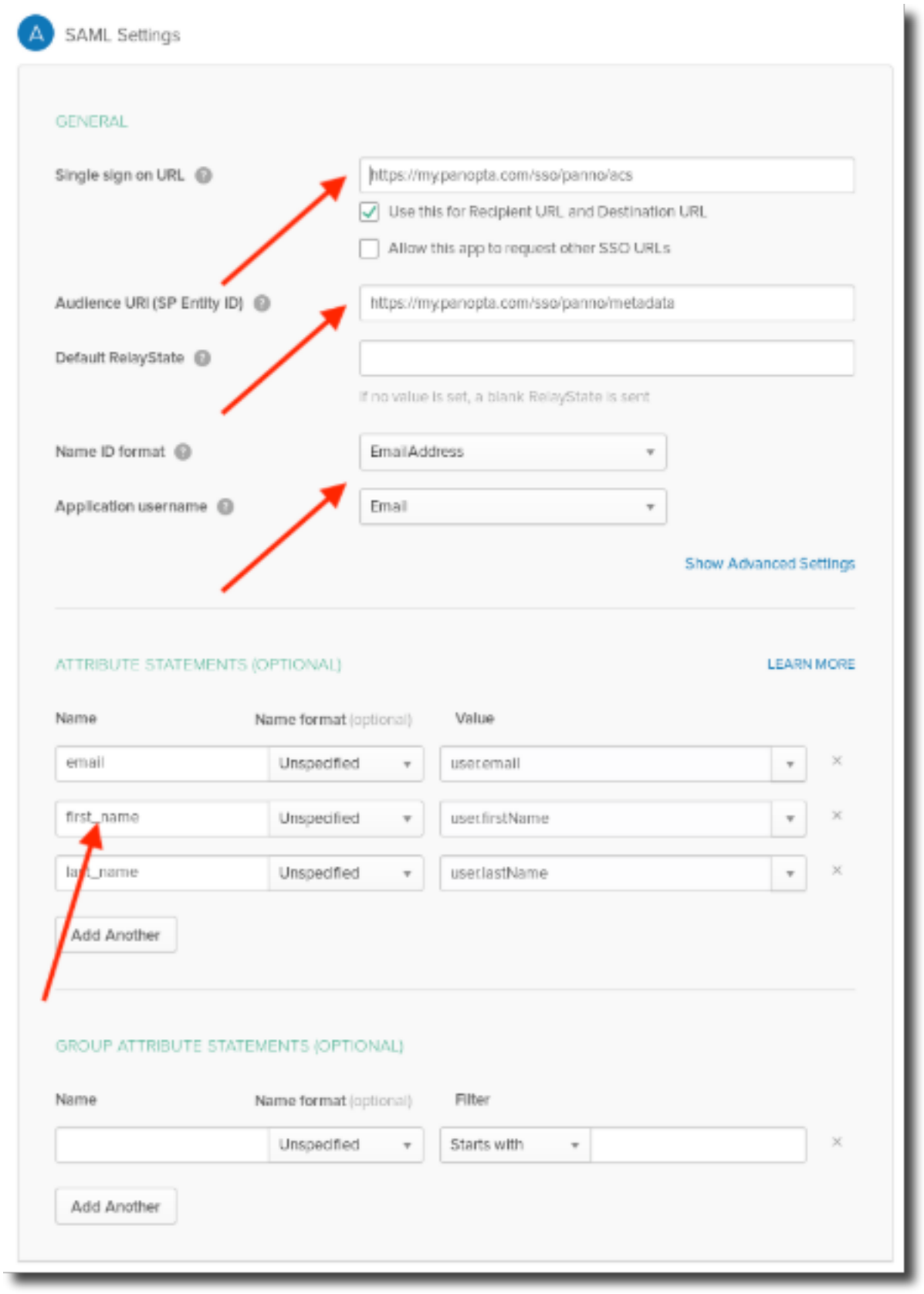
The **Single Sign On URL** is constructed with the **URL Fragment** configured in your SSO configuration. The single sign-on URL would follow the format: **https://fortimonitor.forticloud.com/sso/<url fragment>/acs**

The **Audience URI** follows the same format as the **Single Sign On URL**, but changing **acs** for metadata, such as **https://fortimonitor.forticloud.com/sso/<url fragment>/metadata**.

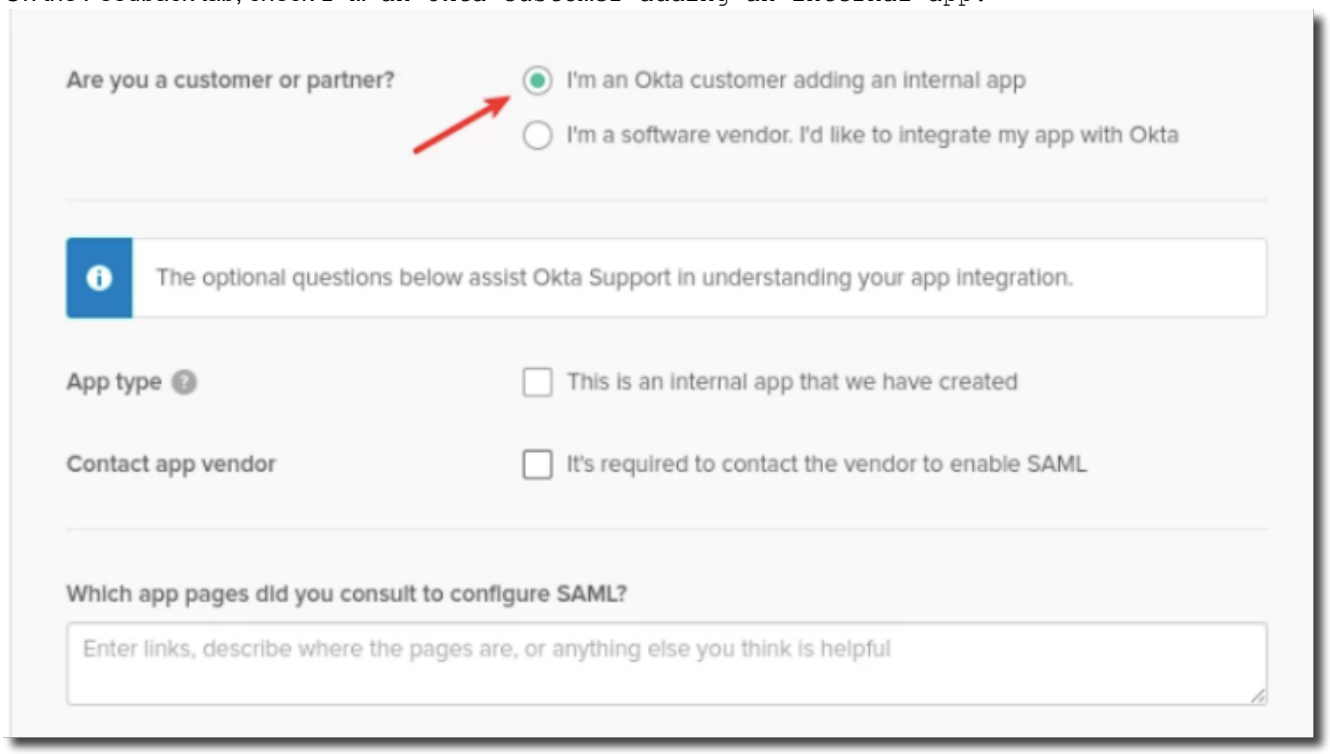
Lastly as FortiMonitor uses the email address as the base for identifying your user. Set the **Name ID** format as **EmailAddress** and **Application username** as **email**.

In the **optional attribute statements**, set the following three fields:

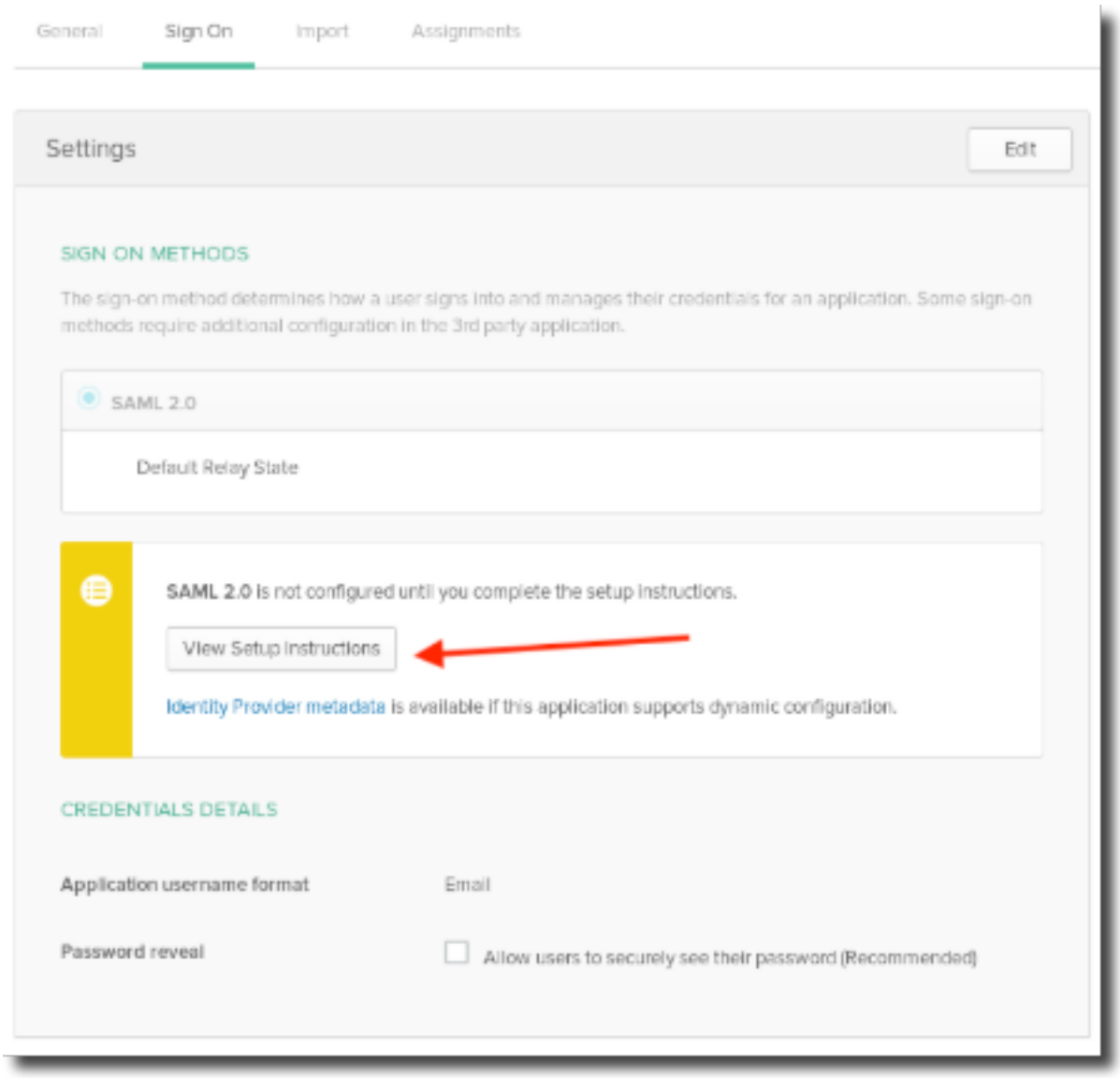
- **email** as **user.email**
- **first_name** as **user.firstName**
- **last_name** as **user.lastName**


A
SAML Settings
GENERAL
Single sign on URL
https://mypenopta.com/sso/panno/acs
Use this for Recipient URL and Destination URL
Allow this app to request other SSO URLs
Audience URI (SP Entity ID)
https://mypenopta.com/sso/panno/metadata
Default RelayState
If no value is set, a blank RelayState is sent
Name ID format
EmailAddress
Application username
Email
Show Advanced Settings
ATTRIBUTE STATEMENTS (OPTIONAL)
LEARN MORE
Name
Name format (optional)
Value
email
Unspecified
user.email
first_name
Unspecified
user.firstName
last_name
Unspecified
user.lastName
Add Another
GROUP ATTRIBUTE STATEMENTS (OPTIONAL)
Name
Name format (optional)
Filter
Unspecified
Starts with
Add Another

On the Feedback tab, check `I'm an Okta customer adding an internal app.`

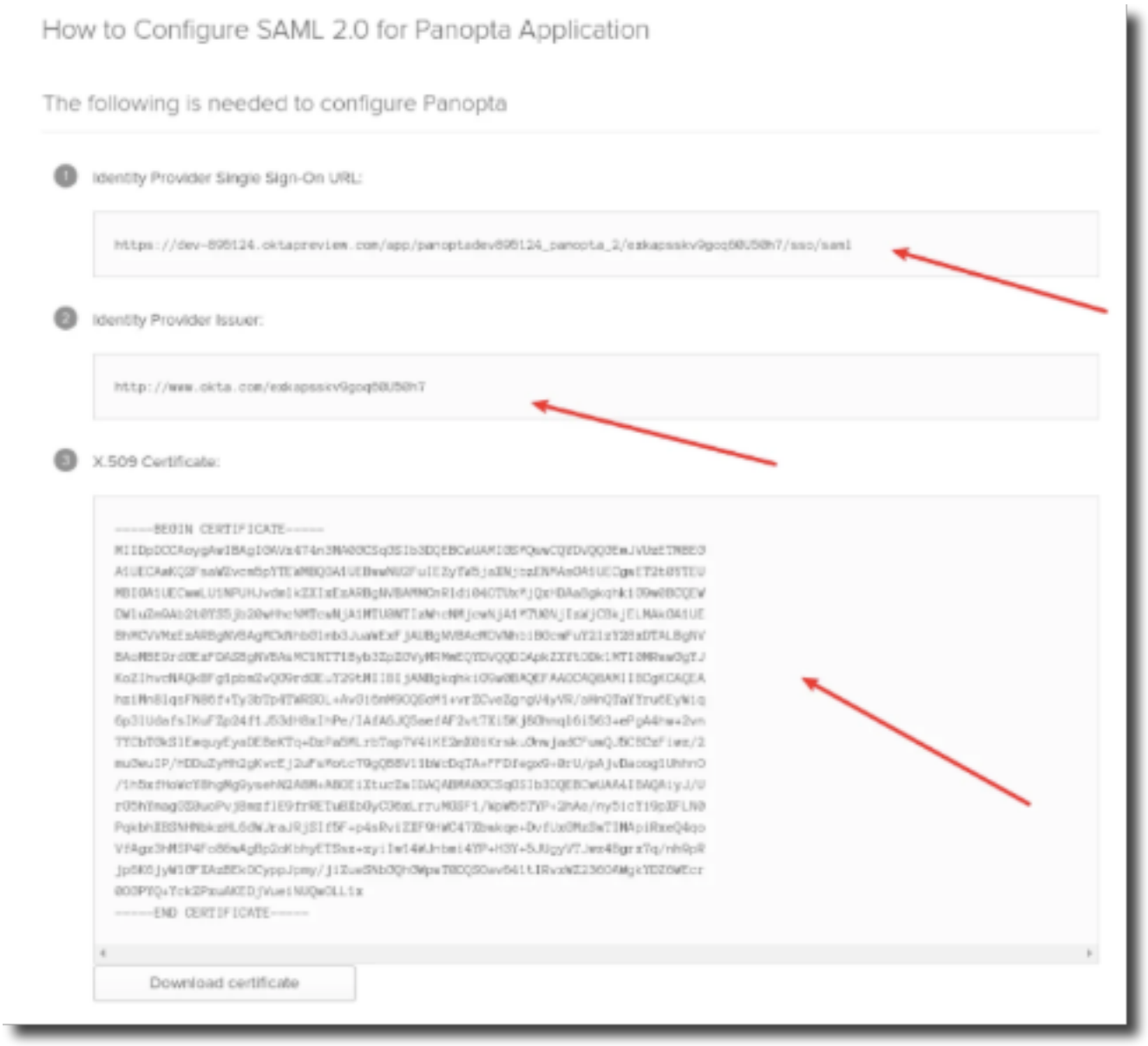


You should be taken to an page detailing the FortiMonitor application we just created. Click `View Setup instructions.`

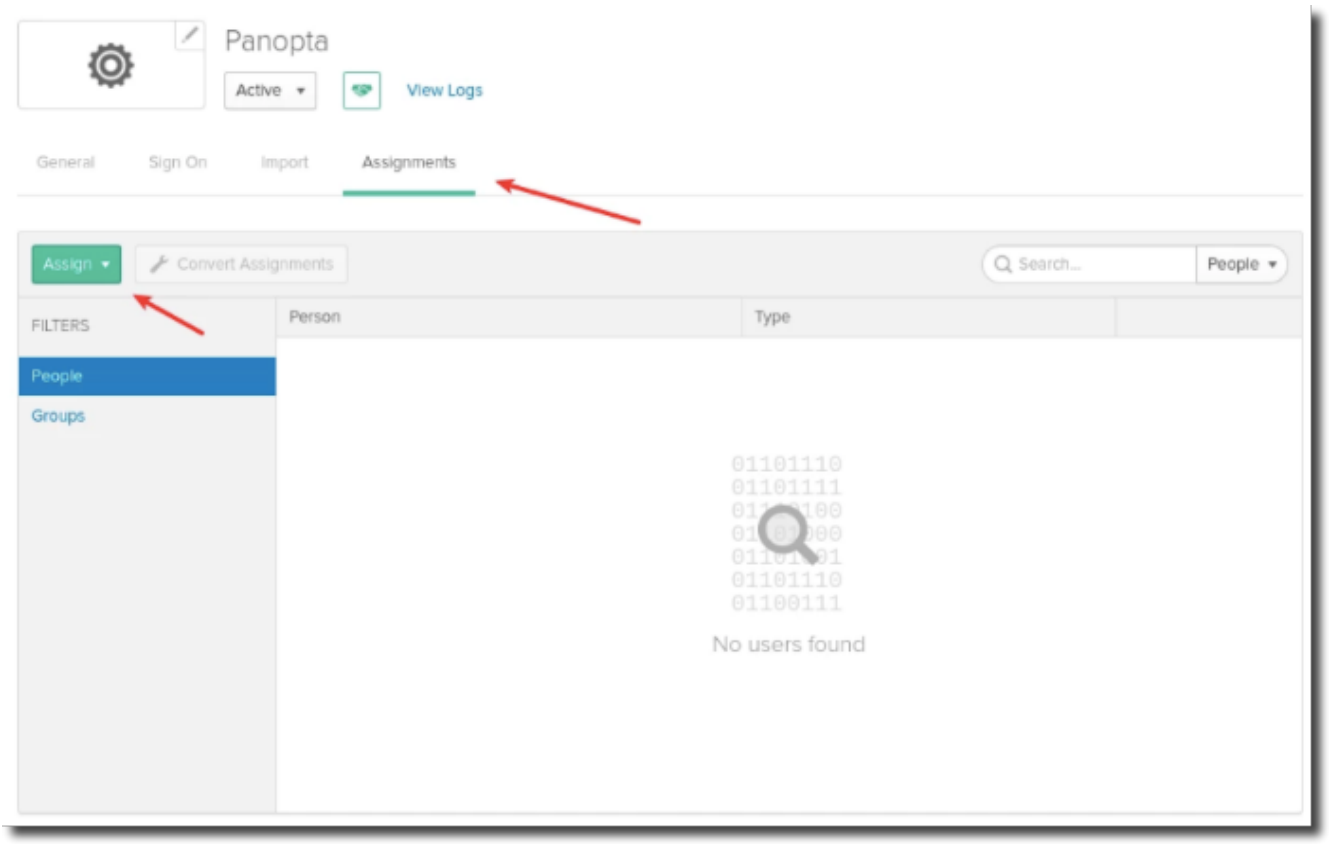


Utilize the data in this screen to complete the **General** section of your FortiMonitor SSO Configuration.

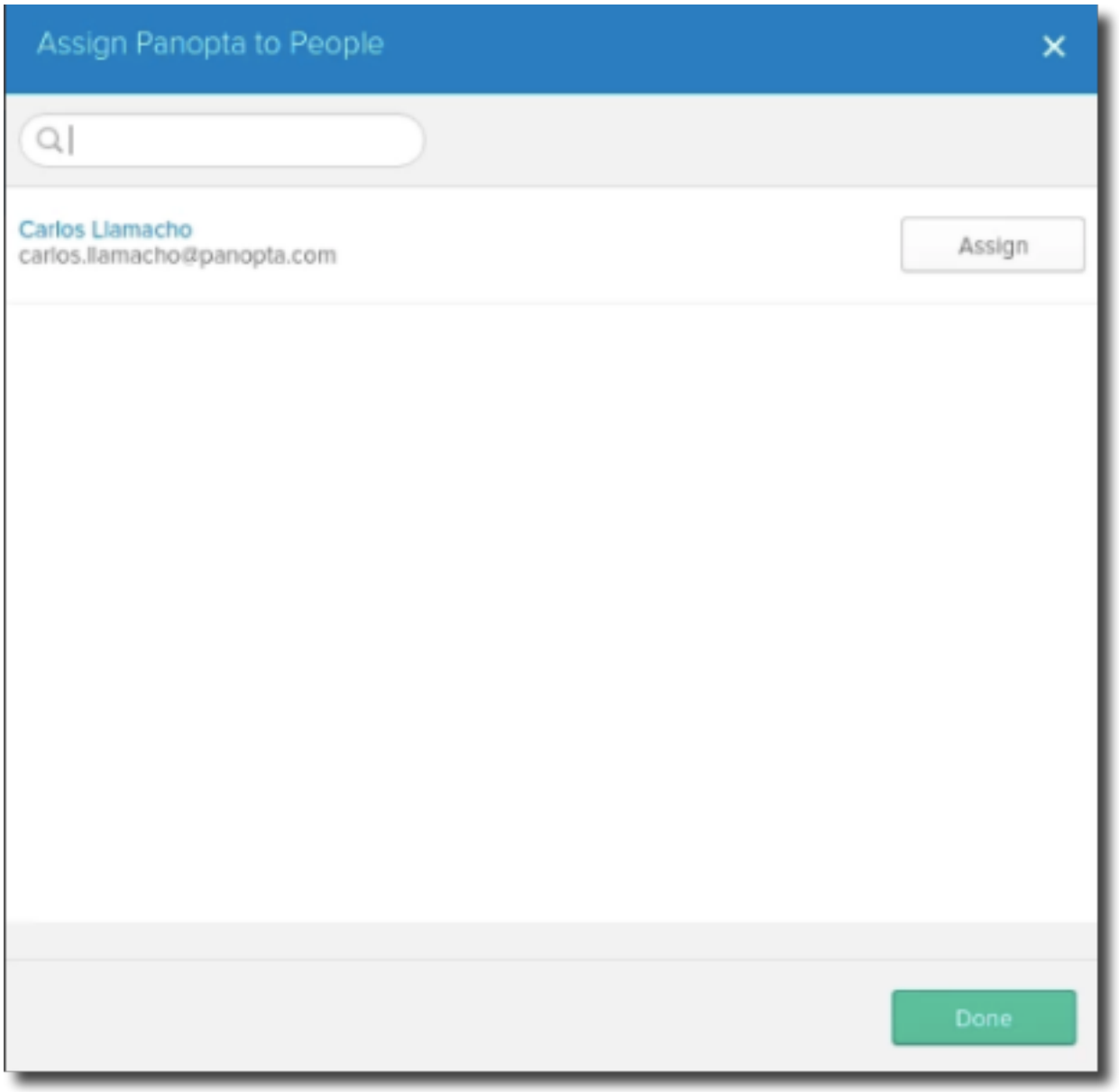
- Identity Provider Single Sign-On URL is the Login URL
- Identity Provider Issuer is the Entity ID
- The certificate must be copied in Certificate input, under the Certificates section
- Login binding must be filled with urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST



Lastly, assign any user you want to access the FortiMonitor app in Okta. Go to the assignment tab in the Application page and click Assign.



Search for people you want to be able to login into FortiMonitor.



Set Up SSO with Azure AD

Before getting started, see SSO Configuration to learn more about our general SSO settings.

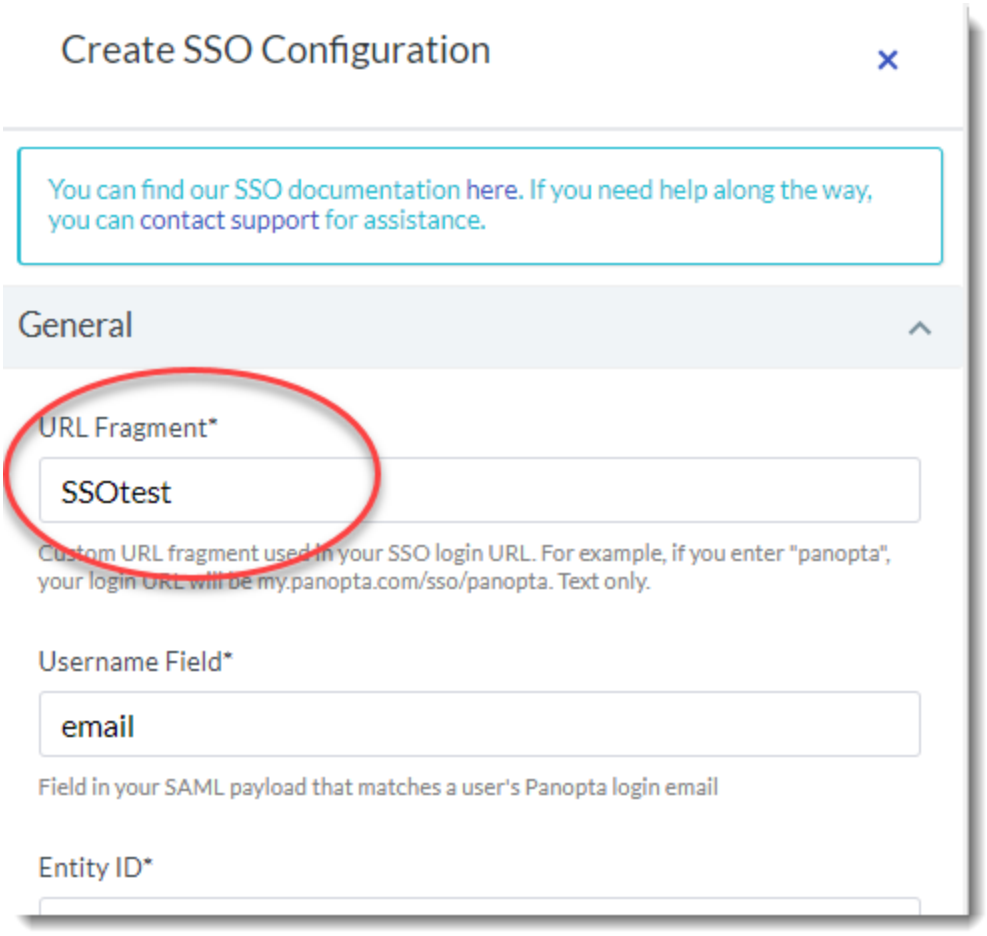
Note: See Quickstart: Enable single sign-on for an enterprise application - Azure AD for the prerequisites.

To set up SSO with Azure AD, perform the following procedures:

1. Create an SSO integration in FortiMonitor
2. Enable and configure Azure Active Directory SSO in Azure
3. Create a SAML signing certificate
4. Continue the SSO configuration in FortiMonitor
5. Upload the **metadata.xml** file from FortiMonitor to Azure

Create an SSO integration in FortiMonitor

1. Log in to FortiMonitor.
2. Create an SSO integration by clicking **Teams & Activity** > **Integrations** > **Microsoft Active Directory.**
3. Enter a **URL Fragment**. The URL Fragment is any custom word (text only) that will be attached to the FortiMonitor login URL. For example, if you set testing your FortiMonitor login will be **my.us01.fortimonitor.com/sso/testing**. Note that the URL fragment must be unique across all FortiMonitor customers. If you get an error saying the fragment is in use, please use something relevant to your organization. The URL Fragment is also based on your region. If you are in the EU region, **my.eu01.fortimonitor.com** will be used. For more information, see Multi-region support.



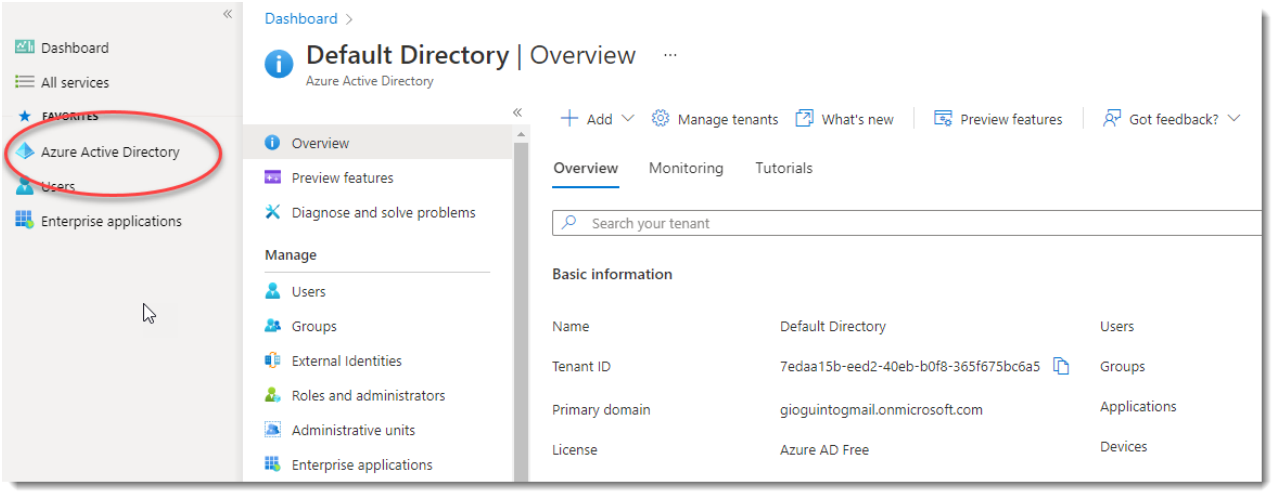
Note: New users should use **my.us01.fortimonitor.com** or **my.eu01.fortimonitor.com** for the SSO URL depending on your region. For long time customers of Panopta, continue using **my.panopta.com**.

4. Open a new tab then go to Azure Active Directory Admin Center using one of the roles mentioned in the Prerequisites (Global Administrator, Cloud Application Administrator, Application Administrator, or owner of the service principal).

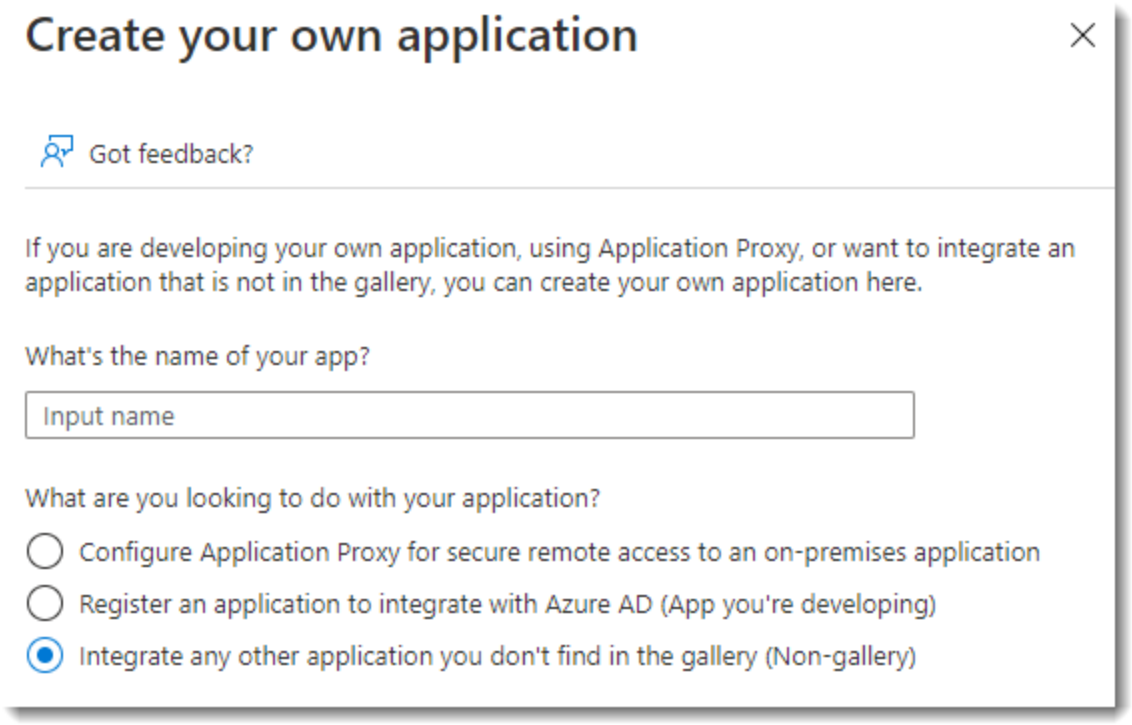
Note: Do not close the FortiMonitor tab.

Enable and configure Azure Active Directory SSO in Azure

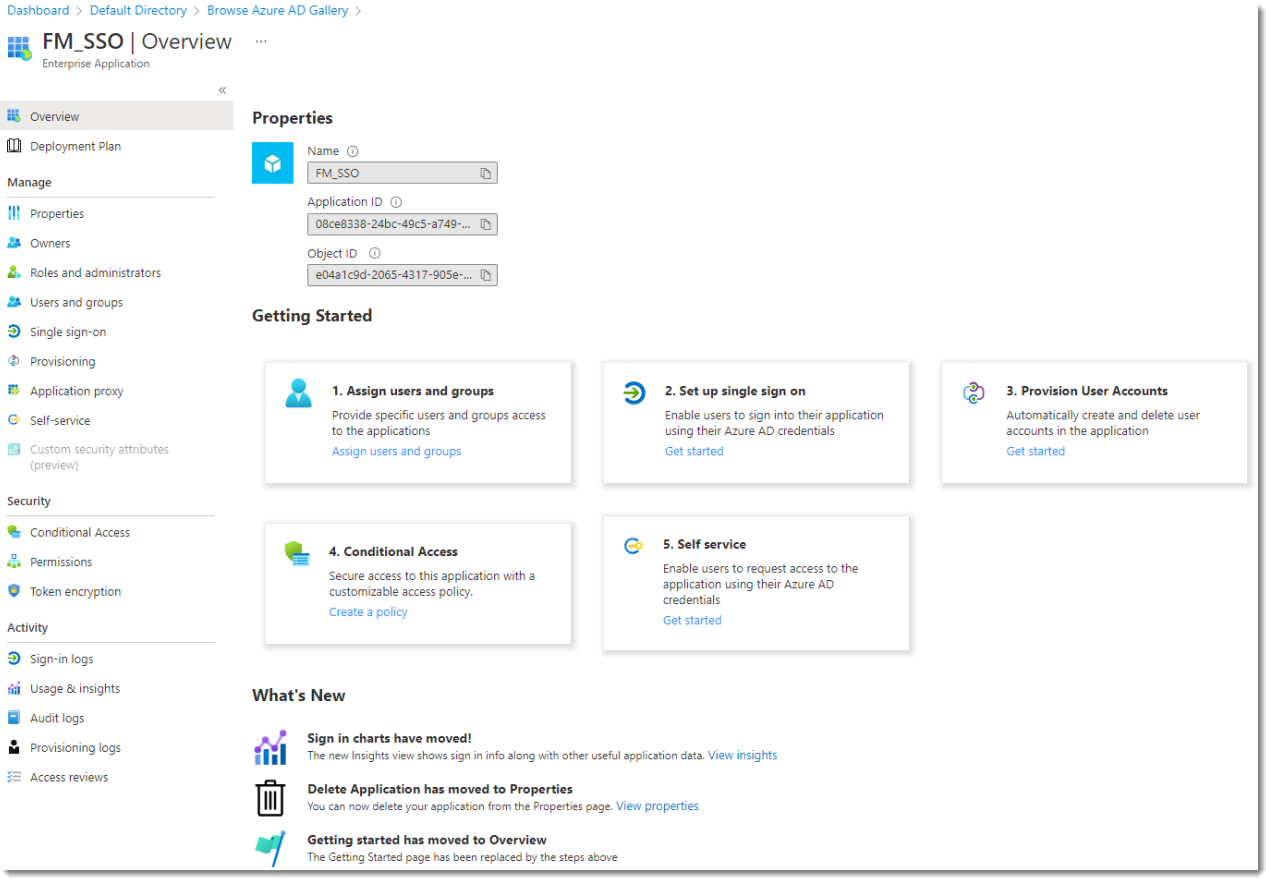
1. Form the Azure Active Directory Admin Center, click **Azure Active Directory**.



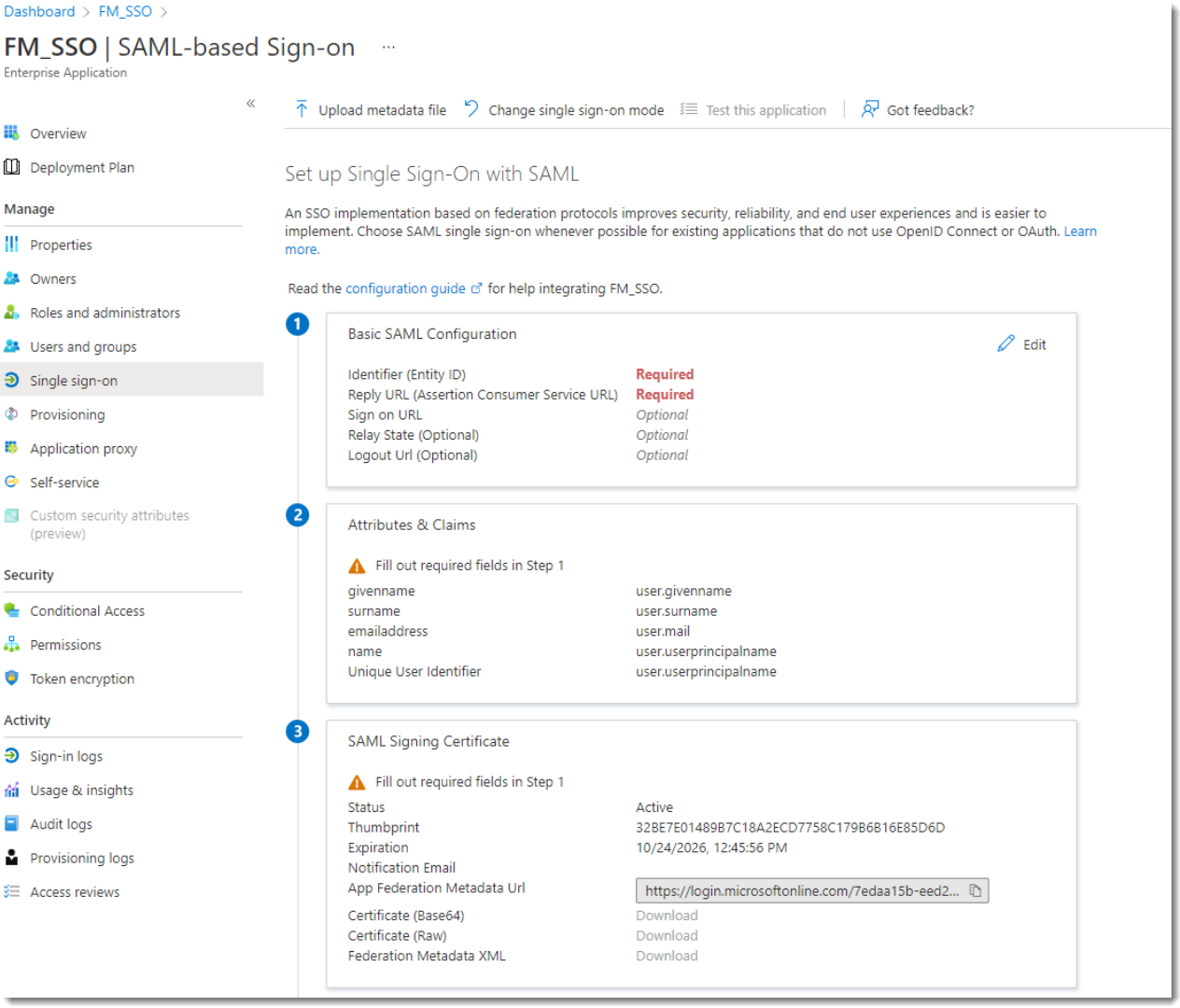
2. From this page, create an Application for the SSO by clicking **Add** > **Enterprise Application**.
3. Click **Create your own application**.
4. Enter a name for the application then select **Integrate any other application you don't find in the gallery (Non-gallery)**.



5. Click **Create**. The dashboard for the newly created application will be displayed.



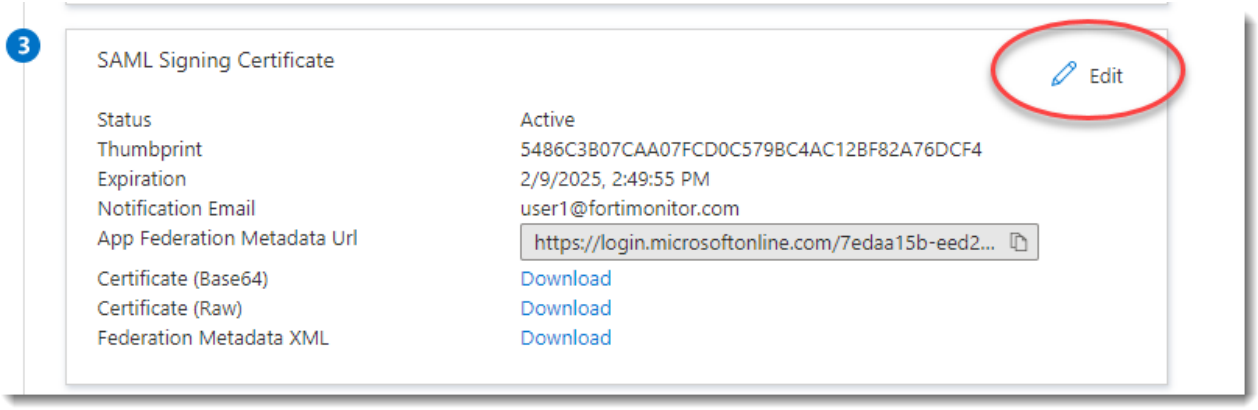
6. Click **Set up single sign on** > **SAML**. The **Set up Single Sign-On with SAML** configuration page is displayed.



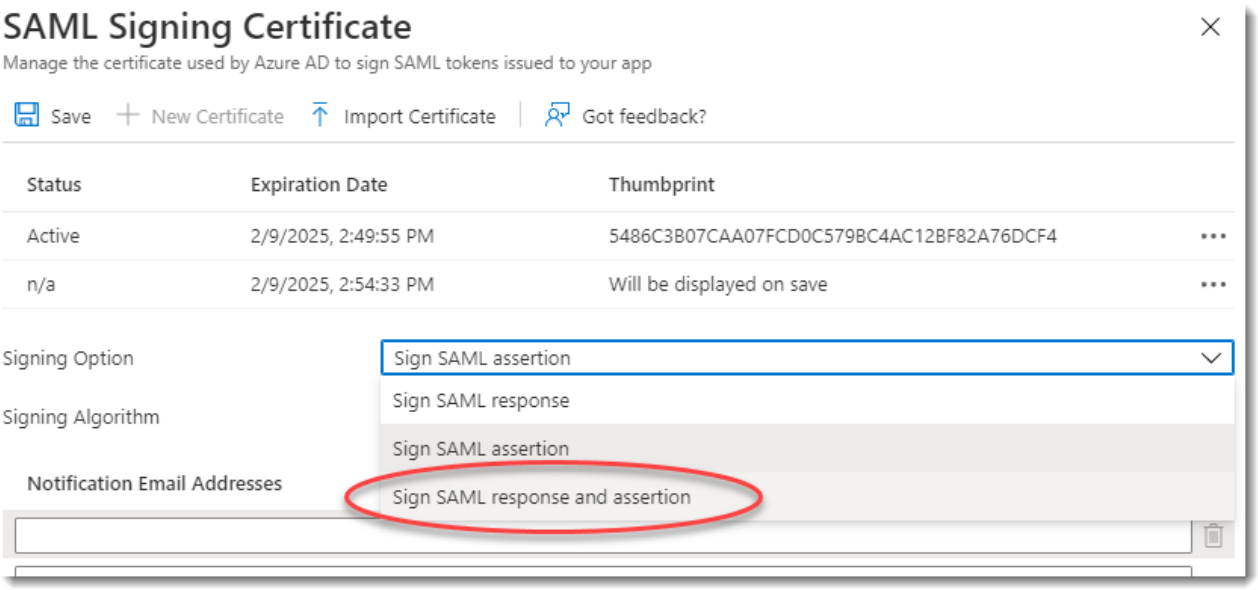
7. Edit the **Basic SAML Configuration** then fill in the required information:
   a. Identifier (Entity ID): **https://fortimonitor.forticloud.com/sso/<URL_Fragment>/metadata** or **https://my.panopta.com/sso/<URL_Fragment>/metadata**
   b. Reply URL: **https://fortimonitor.forticloud.com/sso/<URL_Fragment>/acs** or **https://my.panopta.com/sso/<URL_Fragment>/acs**
   Note: New users should use **fortimonitor.forticloud.com** for the SSO URL. For long time customers of Panopta, continue using **my.panopta.com**.
8. Save the configuration. Once the SAML is configured, the succeeding steps in Azure will be unlocked, which will allow you to proceed to the following procedure.

Create a SAML signing certificate and download the Azure metadata

1. To create a new SAML signing certificate, click **Edit** in the **SAML signing certificate** section.

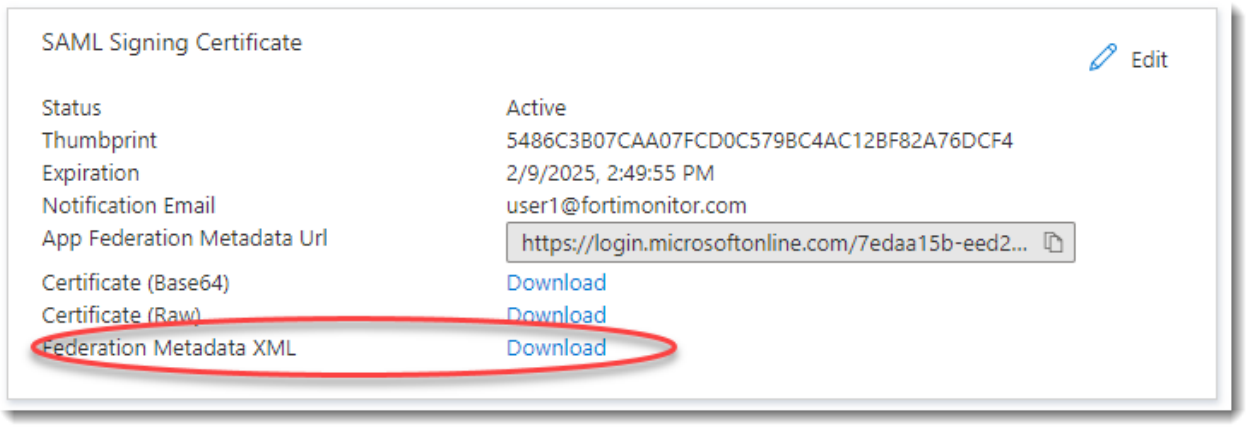


2. Click **New Certificate**.
3. For the **Signing Option**, select **Sign SAML response and assertion**.



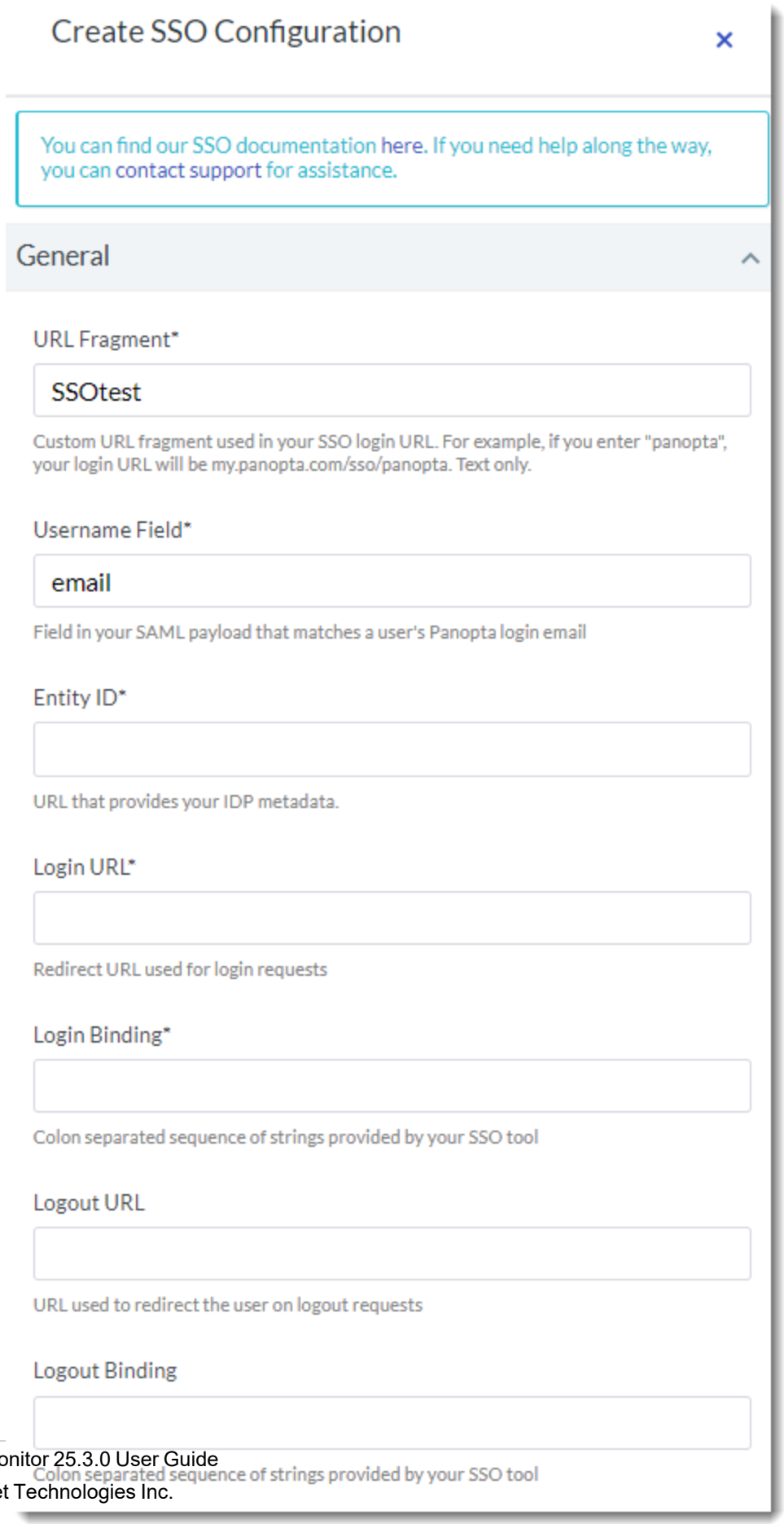
4. Once the new certificate is created, set this to **Active**.
5. Click **Save** to close the configuration drawer.

6. From the **SAML Signing Certificate** section, click **Download** to download the **Federation Metadata XML**.

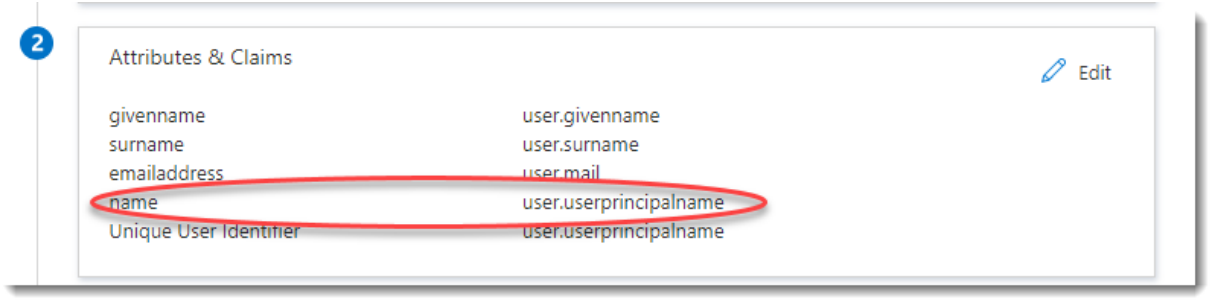


Continue the SSO configuration in FortiMonitor

1. Go back to the FortiMonitor tab and fill in the required information. The fields are described in Single sign-on (SSO).


Create SSO Configuration
You can find our SSO documentation here. If you need help along the way, you can contact support for assistance.
General
URL Fragment*
SSOtest
Custom URL fragment used in your SSO login URL. For example, if you enter "panopta", your login URL will be my.panopta.com/sso/panopta. Text only.
Username Field*
email
Field in your SAML payload that matches a user's Panopta login email
Entity ID*
URL that provides your IDP metadata.
Login URL*
Redirect URL used for login requests
Login Binding*
Colon separated sequence of strings provided by your SSO tool
Logout URL
URL used to redirect the user on logout requests
Logout Binding
Colon separated sequence of strings provided by your SSO tool

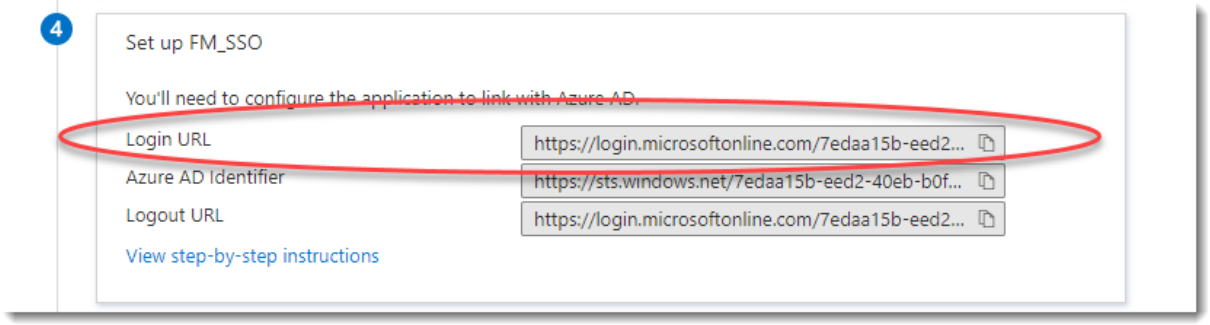
2. The following values can also be acquired from Azure.
   a. **Username Field** is the field that Azure uses for your email address. The correct value must exist in Azure.



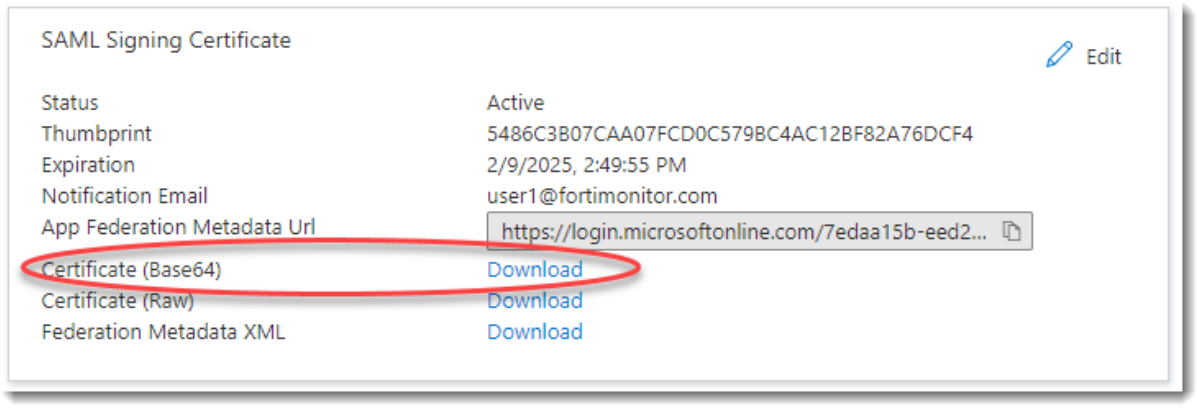
   b. The **Entity ID** and the **Login Binding** can be found in the **Federation Metadata XML** file that you downloaded from Azure.



   c. Copy the **Login URL** from Azure.



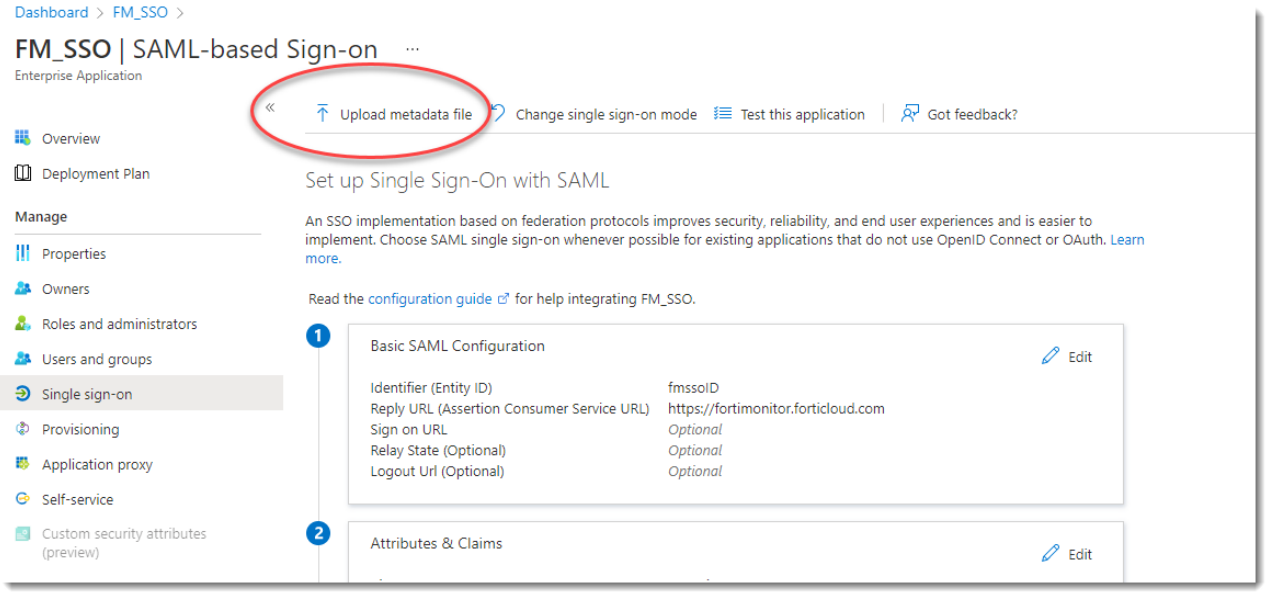
d. Download the certificate from Azure and copy the content to the **Certificate** field in FortiMonitor.



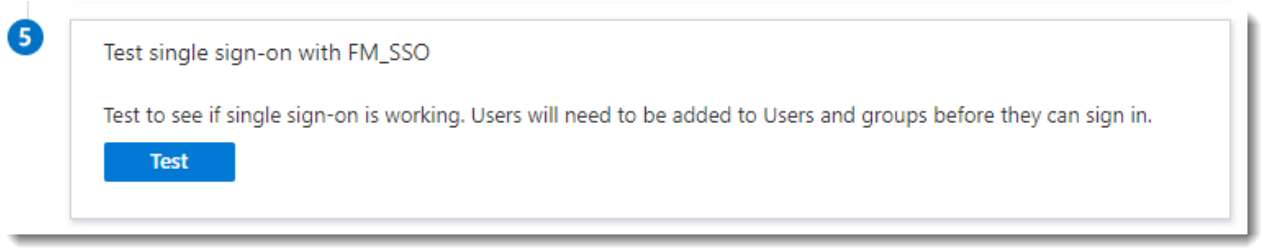
3. In the **User Configuration** section of the configuration drawer, enable **Auto Create Users**.
   You may want to assign the roles manually so that the new user will get that role when they are added to FortiMonitor.
4. Save the changes by clicking on **Save**.

Upload the metadata.xml file from FortiMonitor to Azure

1. Download the **metadata.xml** file from FortiMonitor by going to a browser and entering: **https://fortimonitor.forticloud.com/sso/{URL_Fragment}/metadata** or **https://my.panopta.com/sso/{URL_Fragment}/metadata**
2. Right-click and select **Save As** to save the xml file.
3. Go back to Azure and upload the metadata file. From the Application created previously, upload the metadata.xml file.



4. You can test the integration from the Azure SSO configuration page by clicking **Test** at the bottom of the page.



Note: If you are using **Microsoft Edge Web Browser**, switch to an **InPrivate** Edge window or a different browser to test the SSO integration.

Set up SSO with Google Workplace

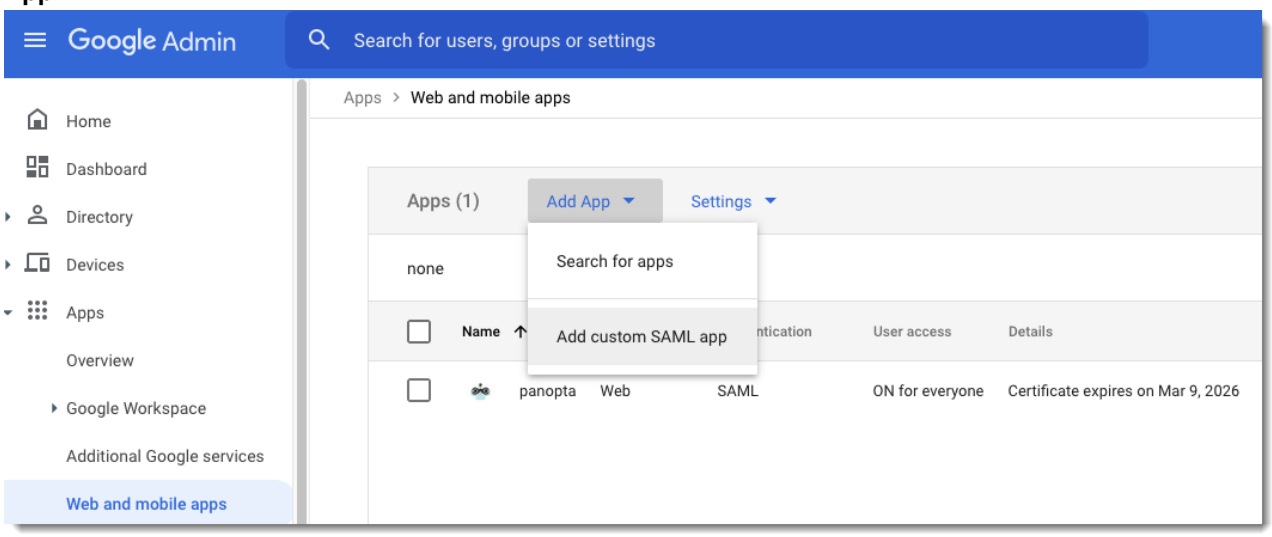
To use Google Workplace as an identity provider with the FortiMonitor Control Panel, you will need to make changes within the Control Panel and within Google Workplace. You will need administrative access for both.

To get started, you need the following:

- Determine what the URL Fragment will be. This will be required in both the Control Panel and Google Workplace setup. Typically it will be the name of your company or group. For this example we'll use "mycompany"
- You will need to go back and forth between the Google Admin Panel and the Control Panel so it is recommended to open a tab for each and log as appropriate.

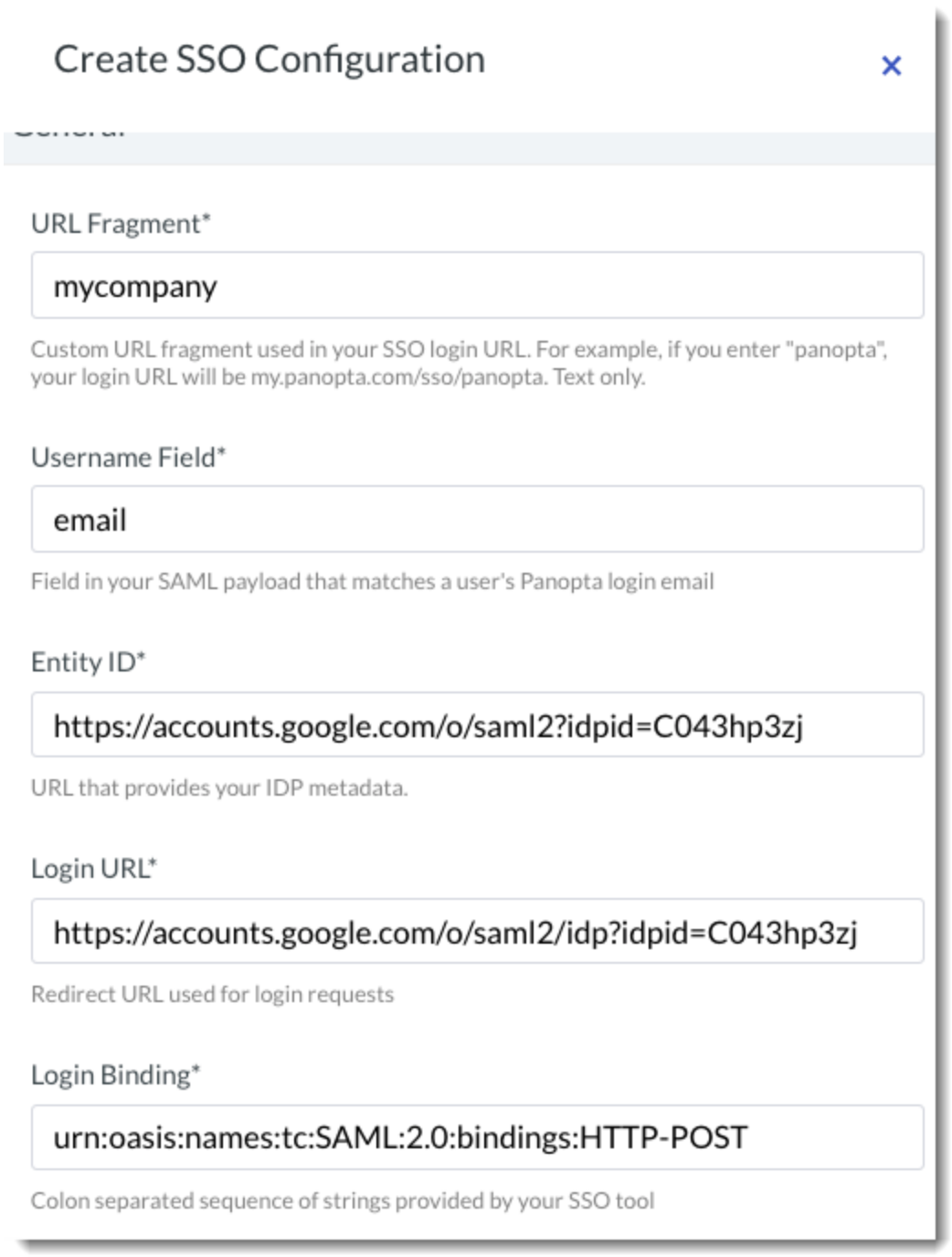
To set up SSO with Google Workplace, perform the following steps:

1. From the FortiMonitor navigation menu, select **Teams & Activity** > **Integrations** > **SAML** > **Configure** to begin the process.
   You can enter the **URL fragment** and the **Login Binding**.
   Note: All fields marked with an asterisk are required. Some of these you will copy over from the Google Admin panel.
2. In the Google Workplace admin interface, go to **Apps** > **Web and Mobile Apps** > **Add App** > **Add Custom SAML App** as shown below:

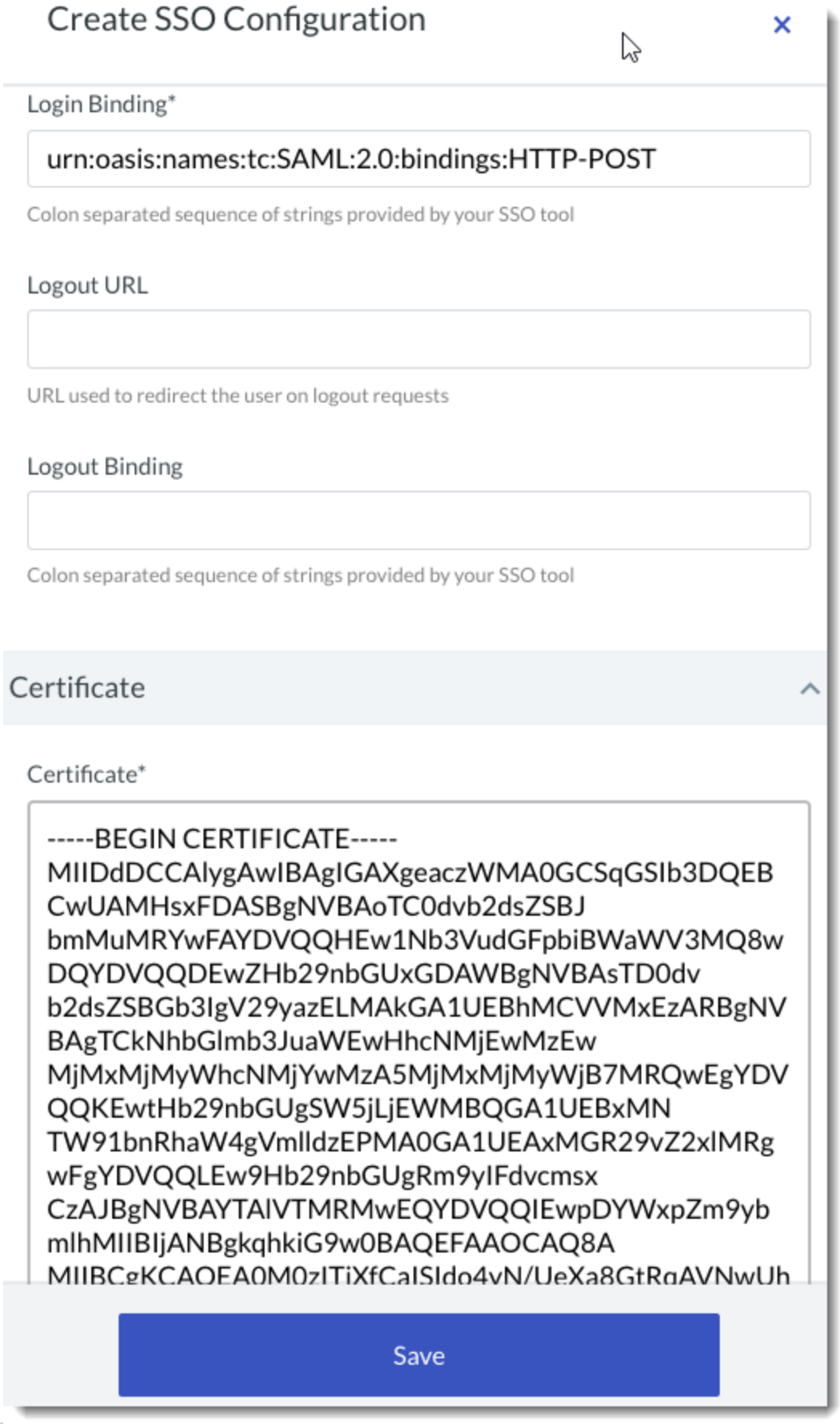


3. Give the App a **Name** and an **Icon (optional)** and click **Continue**.

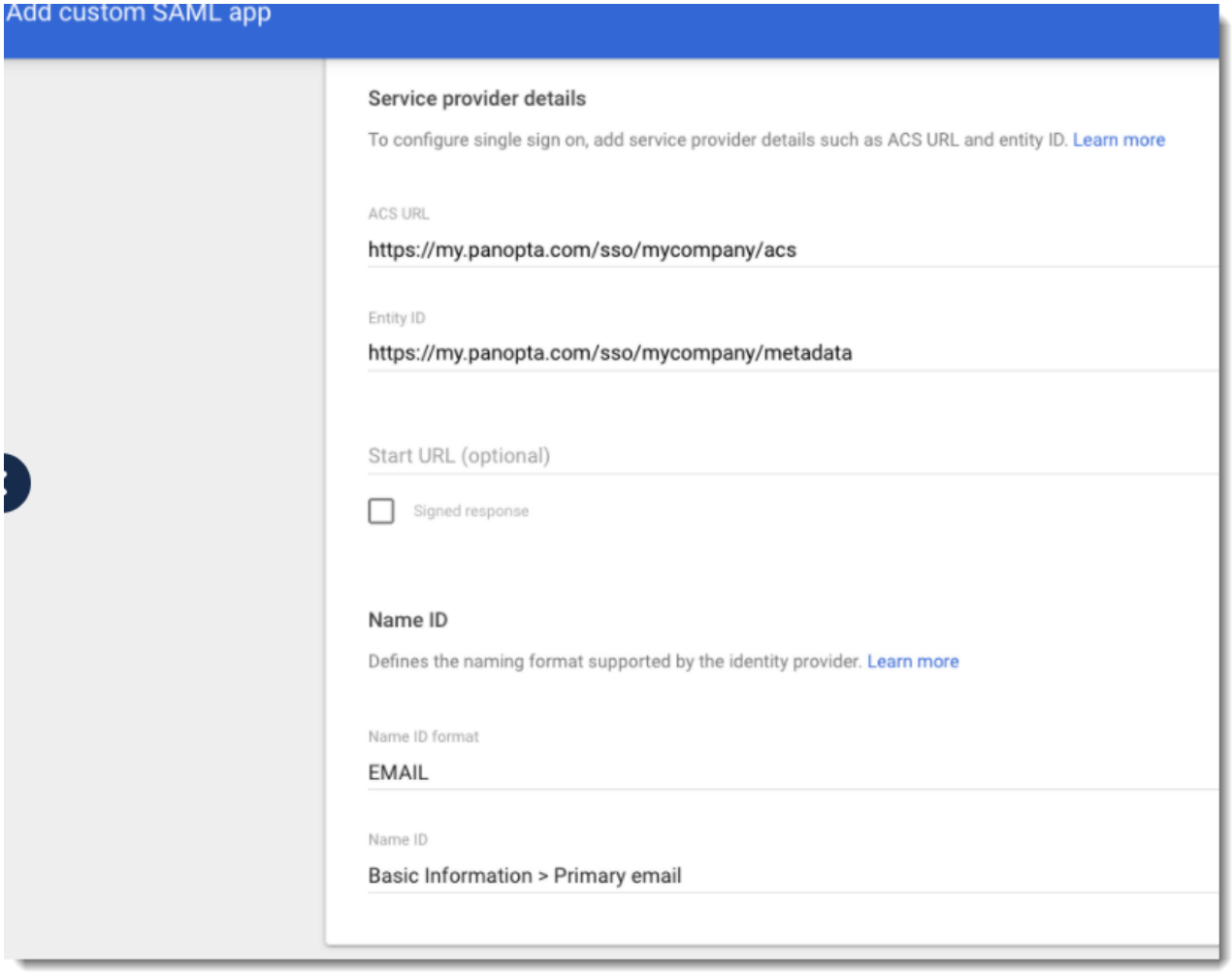
4. On the next screen, you will be presented with an **SSO URL**, an **Entity ID**, and a **Certificate**. You will need to copy each of these into the FortiMonitor **Create SSO Configuration** pane as shown below. The **Entity ID** URL goes in the **Entity ID** field and the **SSO URL** (Google) goes in the **Login URL** box (FortiMonitor):



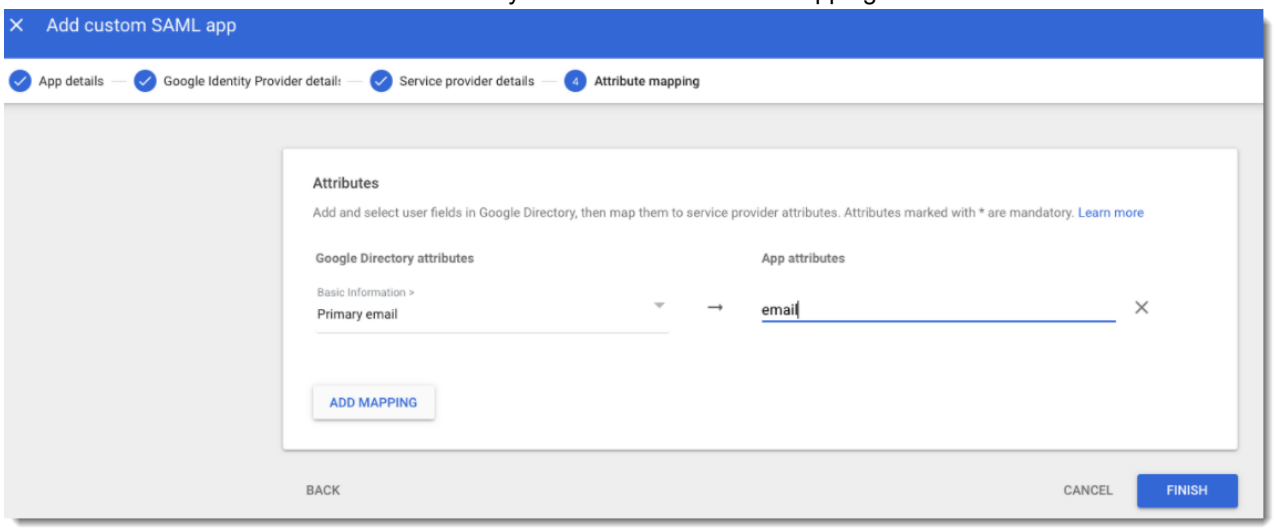
5. Next scroll down in the Control Panel and copy the **Certificate** from the Google Admin Console to the Control Panel:


Create SSO Configuration
Login Binding*
urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
Colon separated sequence of strings provided by your SSO tool
Logout URL
URL used to redirect the user on logout requests
Logout Binding
Colon separated sequence of strings provided by your SSO tool
Certificate
Certificate*
-----BEGIN CERTIFICATE-----
MIIDdDCCAlygAwIBAgIGAXgeaczWMA0GCSqGSIb3DQEB
CwUAMHsxFDASBgNVBAoTC0dvb2dsZSBJ
bmMuMRYwFAYDVQQHEw1Nb3VudGFpbiBWaWV3MQ8w
DQYDVQQDEwZHb29nbGUxGDAWBgNVBAsTD0dv
b2dsZSBGb3IgV29yazELMAkGA1UEBhMCVVMxEzARBgNV
BAgTCkNhbGlmb3JuaWEwHhcNMjEwMzEw
MjMxMjMyWhcNMjYwMzA5MjMxMjMyWjB7MRQwEgYDV
QQKEwtHb29nbGUgSW5jLjEWMBQGA1UEBxMN
TW91bnRhaW4gVmlldzEPMA0GA1UEAxMGR29vZ2xlMRg
wFgYDVQQLEw9Hb29nbGUgRm9yIFdvcmsx
CzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9yb
mlhMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEA0M0zITiXfCaISIdo4vN/UeXa8GtRgAVNwUh
Save

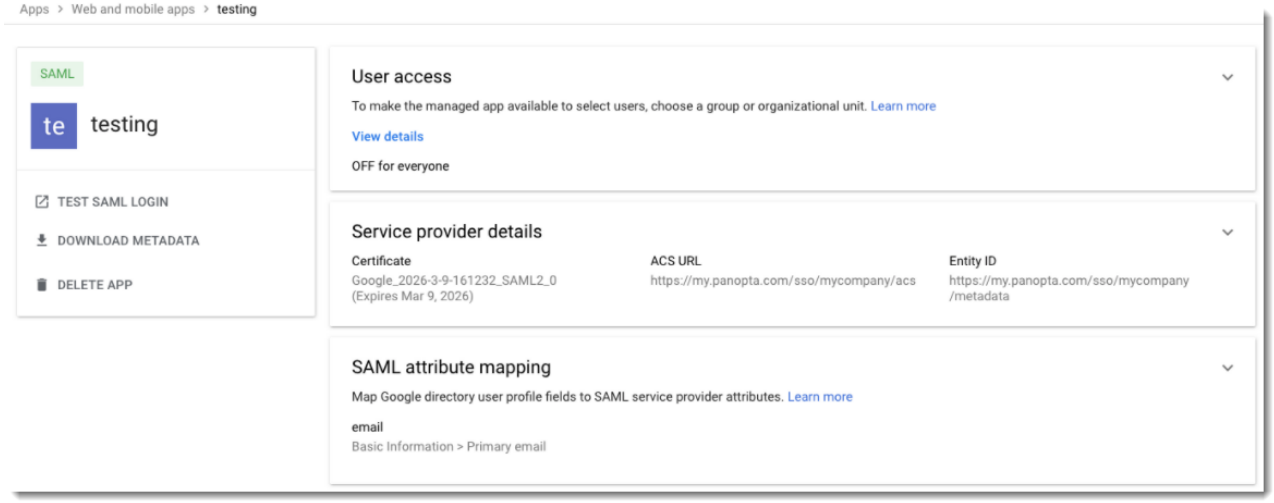
6. You can now hit **Save** on the **Control Panel Create SSO Configuration** screen. You may want to revisit that configuration to force all users to use SSO by enabling **Prevent non-SSO Logins**, but you should test this configuration first before checking that option.
7. Back to the Google Admin Panel, and you can hit **Continue** to go the next screen. Here you will enter the following URLs, again using the example “mycompany”. Replace that string with your own choice for URL Fragment as discussed above.
   ACS URL: **https://my.panopta.com/sso/mycompany/acs**
   Entity ID: **https://my.panopta.com/sso/mycompany/metadata**
   Then set the **Name ID format** = **EMAIL**
   You screen should now look like this:



8. Press continue and on the next screen where you can add an attribute mapping as follows:



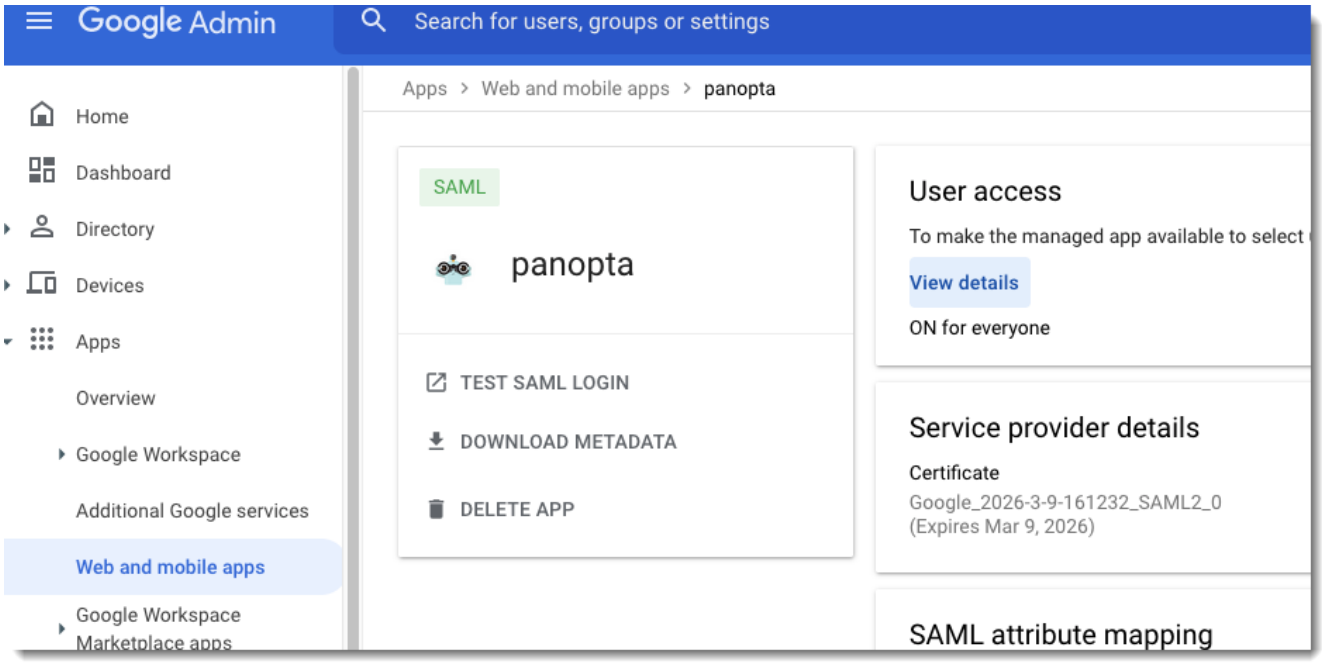
9. Click **Finish**.
10. As you now view the SSO Setup, ensure that this integration is turned **On** for **Everyone** under User Access, or you can also allow for only a group within your organization. Visit the **Google Learn More** link for details. While testing, it is probably best to enable **On for Everyone**. Click the **View Details** link on the **User Access** pane to change this setting.



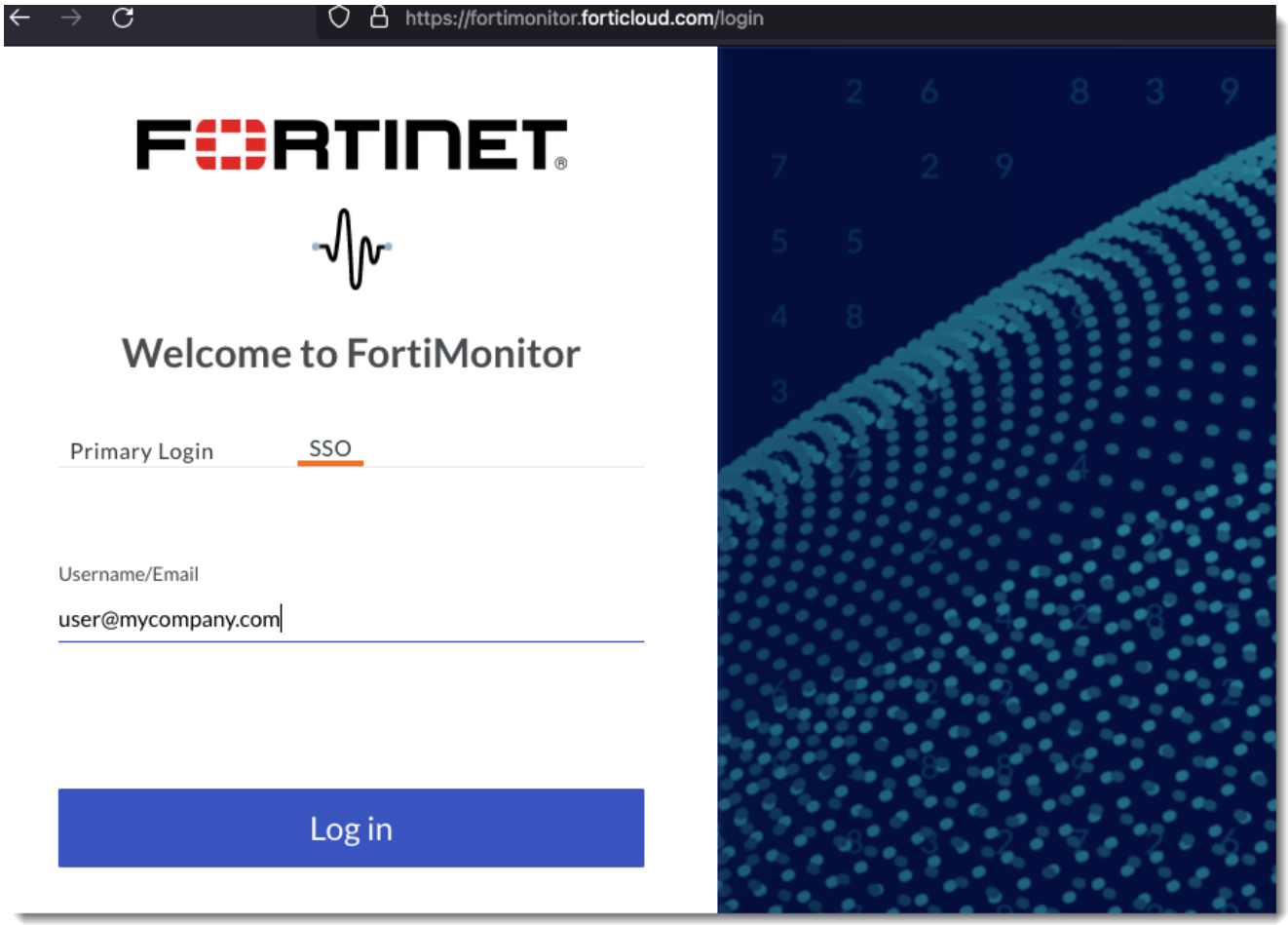
You have now completed the configuration.

Test the SSO configuration

You can test the integration using the **Test SAML Login** link shown below:



You can also test on the FortiMonitor login page:



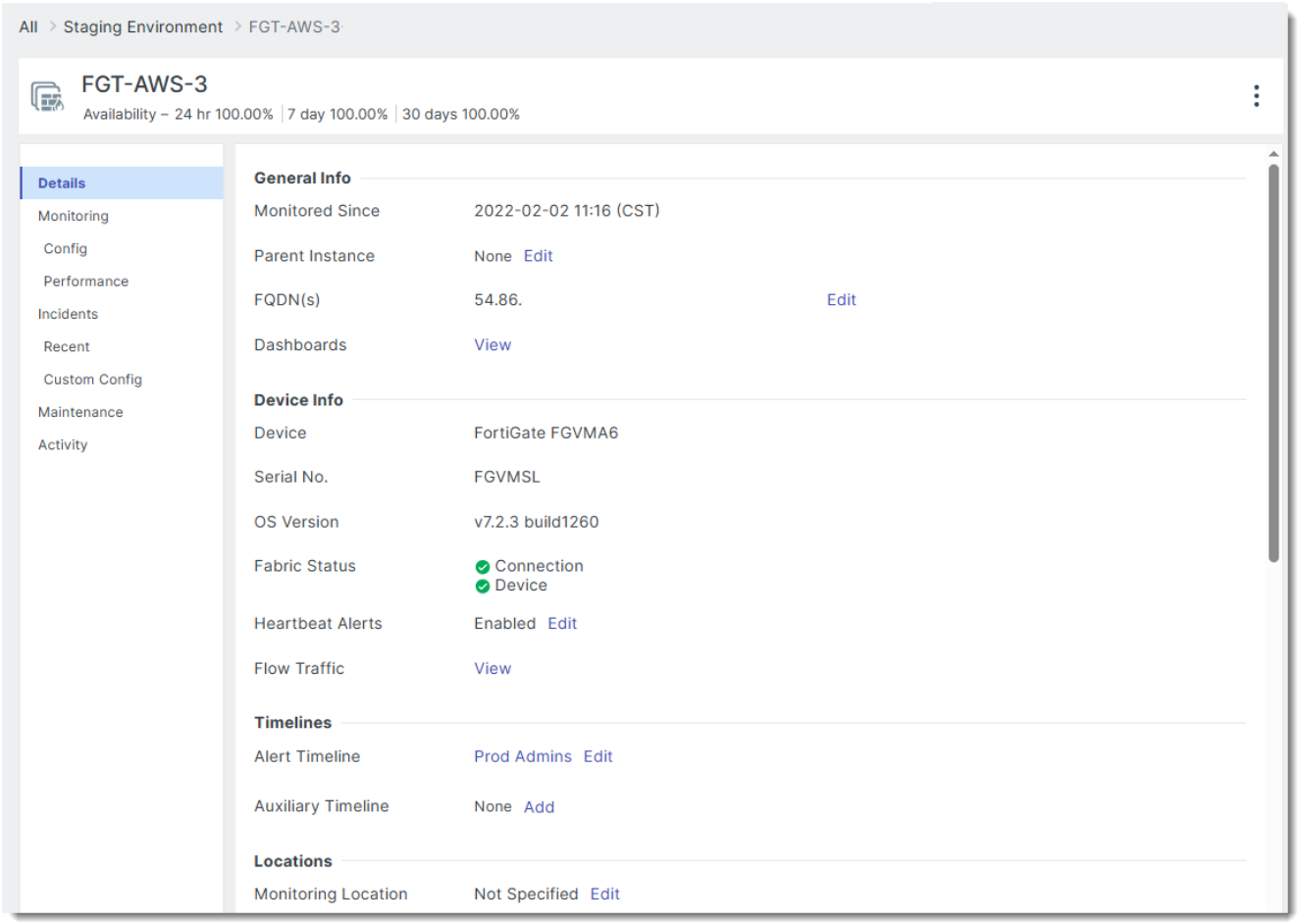
Troubleshooting:

If you run into errors when testing:

- Double check all of the URLs and URL Fragments
- Ensure you are logged into the correct Google Account
- Make sure you have enabled this for all users in Google, or if you limit by groups that your test user is in the correct groups
- If you had previous SAML integrations you may have an expired Certificate show up in the Google configuration. You can view and manage the Google certificates under Apps → Web and Mobile Apps → Settings

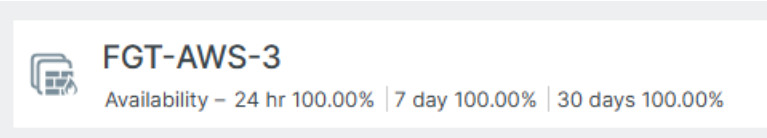
The **Instance Details** page shows all of the relevant information about the instance, including the instance performance, monitoring configuration, incidents, maintenance schedules, availability, and more. The **Instance Details** page supports several actions to manage the instance metrics, maintenance, and if installed, the FortiMonitor Agent and CounterMeasures. Performance-based graphs are displayed in the **Performance** section of the **Instance Details** page to help you visualize the status of each metric configured for the instance.

The **Instance Details** page is divided into several sections and tabs. Go through the following sections to learn more.



Availability

The **Availability** section shows the availability of the instance in a 24-hour, 7-day, or 30-day time range.



Actions

This section lists all of the actions that you can perform from the **Instance Details** page. Note that depending on the type of instance, some actions may not be available. Select the **3-dot** icon to see the full list of actions available to the instance.

- Flush DNS – Forces a DNS update after an IP address change.
- Delete Instance – Select the **Delete Instance** button to remove the instance from FortiMonitor.

For network devices, you can initiate port discovery by clicking the **Port Scan** button.



Details tab

General information

Field	Description
Monitored Since	The date and time when this instance began being actively monitored.
Server Key	Unique identifier assigned by FortiMonitor to the instance.
Parent Instance	Primary instance to which this instance is linked.
Agent Status	The status of the FortiMonitor Agent, including the Agent's version, last sync, heartbeat alerts, CounterMeasures status, and last metadata build. If no Agent is installed on the instance, you may click **Install** to install a supported Agent.
FQDN(s)	Fully Qualified Domain Names (FQDNs) associated with this instance.
Dashboards	Links to dashboards where this instance is included.

Device information (Fabric devices)

Available for Fabric devices only.

Field	Description
Device	The name of the hardware device.
Serial No.	The unique serial number assigned to the hardware device by the manufacturer.
OS Version	The current operating system version running on the hardware device.
Fabric Status	Indicates the device's connectivity and health with the Fabric. Click **Rediscover** to run discovery for Fabric devices.
Heartbeat Alerts	Enable or disable heartbeat alerts.
Flow Traffic	View data showing the volume and pattern of network traffic flowing through the device.

Device information (SNMP devices)

Available for Fabric devices only.

Field	Description
Device	The name of the hardware device.
SNMP Description	A text description retrieved via SNMP that typically includes details about the device.
SNMP Location	The physical or logical location of the device as reported via SNMP.
SNMP Credential	SNMP credential used by the device. Click **Edit** to change the credential.
Flow Traffic	View data showing the volume and pattern of network traffic flowing through the device.
Port Discovery	Shows port discovery status. For example: Port Discovery Status — complete Result — success Last Discovery Time — 2020-08-13 12:43:13 (CDT) Ports Discovered — 59 Port Scan — Request Port Scan

Timelines

Field	Description
Alert Timeline	The **Alert Timeline** associated with this instance.
Auxiliary Timeline	The auxiliary **Alert Timeline** associated with this instance.

Locations

Field	Description
Monitoring Location	The public checker or OnSight vCollector used to monitor this instance.
Geo Location	Location entered in this field is used in Geomap widgets. Latitude and longitude are entered in degrees, with + values corresponding to north latitude and east longitude, and - values corresponding to south latitude and west longitude.

Settings

Field	Description
Instance Group	The group where this instance belongs to.
Templates	The templates applied to this instance.
Tags	The tags applied to this instance. Click **Edit** to add or remove tags.
Attributes	The attributes applied to this instance. Click **Edit** to add or remove attributes.
Flapping Incident	Click **Configure** to set the flapping incident threshold for this instance.

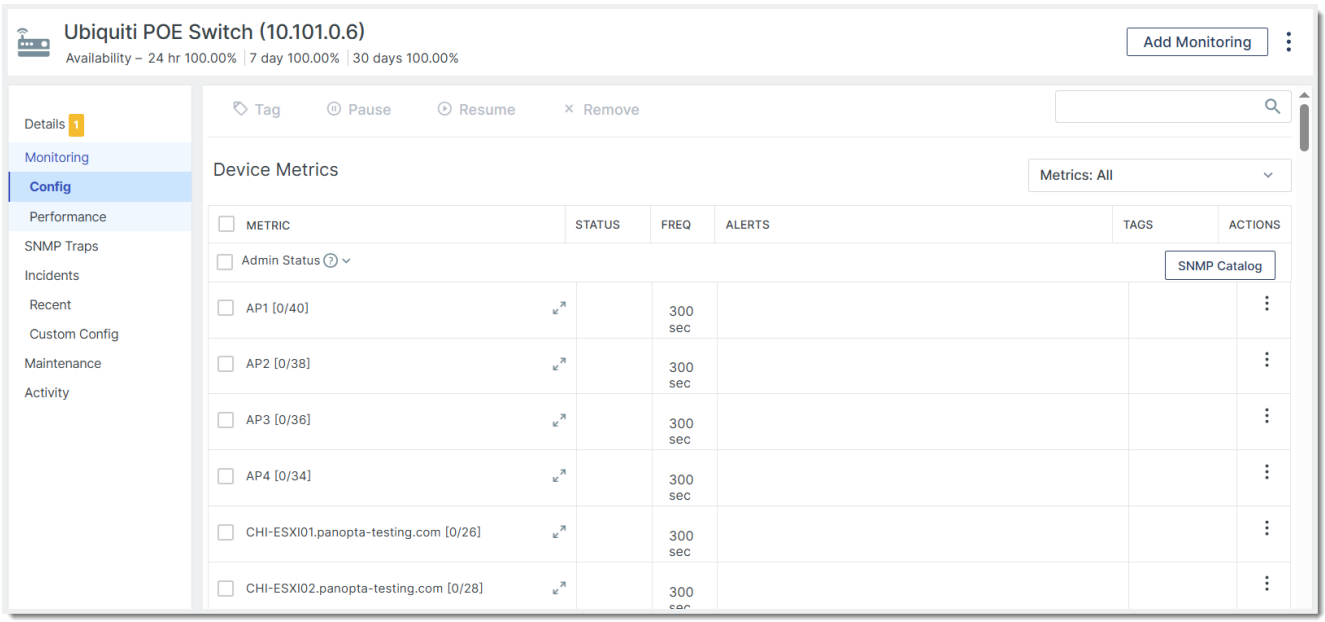
Others

Field	Description
Prometheus Endpoints	Add or remove Prometheus endpoints.

Monitoring tab

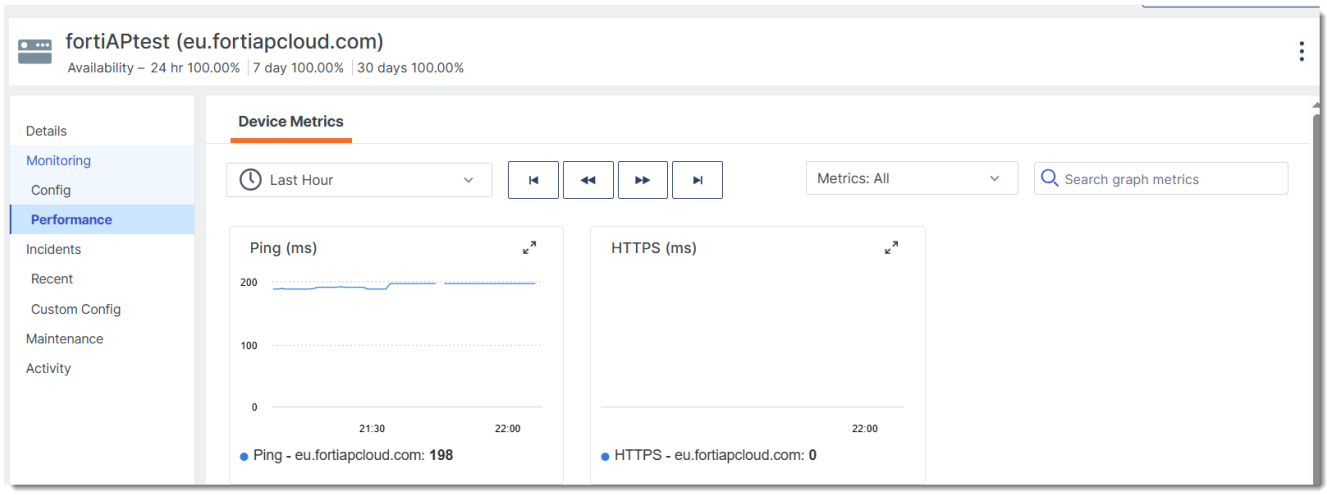
Config

The **Config** section shows all of the metrics configured for this instance. You can add monitoring and configure metrics from this section.



Performance

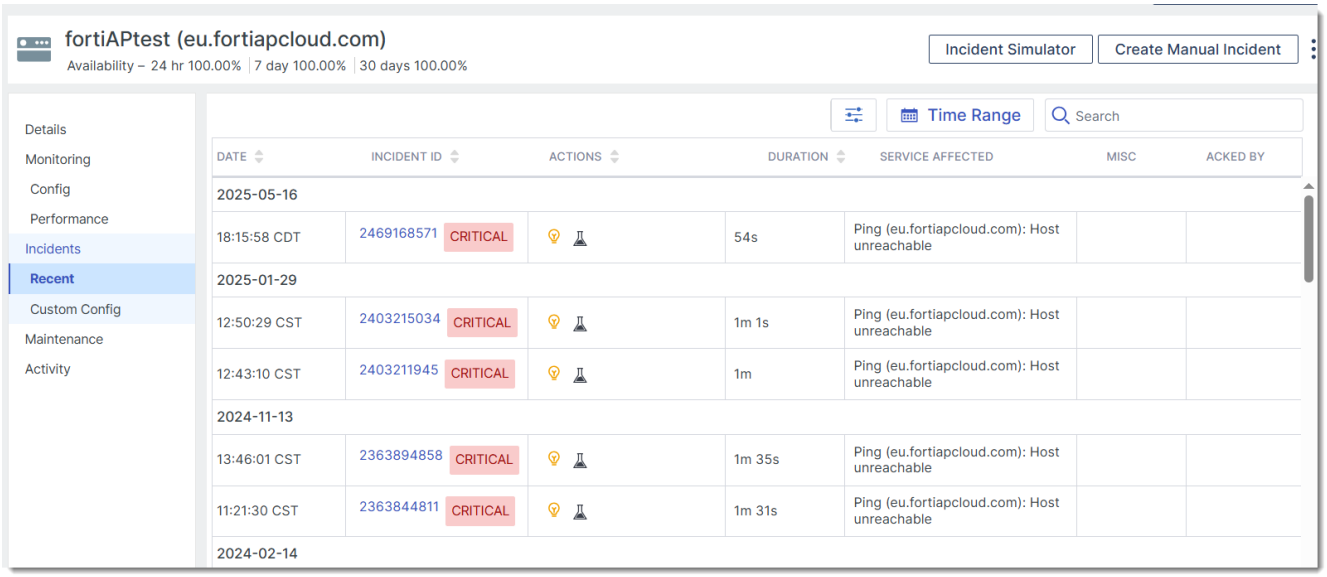
The **Performance** section provides a visual representation of the status of the instance. Each metric is represented by a line graph that can be filtered based on the metric type and scoped using a specific time range.


fortiAPtest (eu.fortiapcloud.com)
Availability – 24 hr 100.00% | 7 day 100.00% | 30 days 100.00%
Details
Monitoring
Config
Performance
Incidents
Recent
Custom Config
Maintenance
Activity
Device Metrics
Last Hour
Metrics: All
Search graph metrics
Ping (ms)
200
100
0
21:30
22:00
Ping - eu.fortiapcloud.com: 198
HTTPS (ms)
22:00
HTTPS - eu.fortiapcloud.com: 0

Incidents tab

Recent

The **Recent** section shows you a summary of all the incidents, whether active or resolved, for this instance. You can filter the list of incidents by selecting a severity type and by setting a time range. You can also create manual incidents and run incident simulations from this tab.



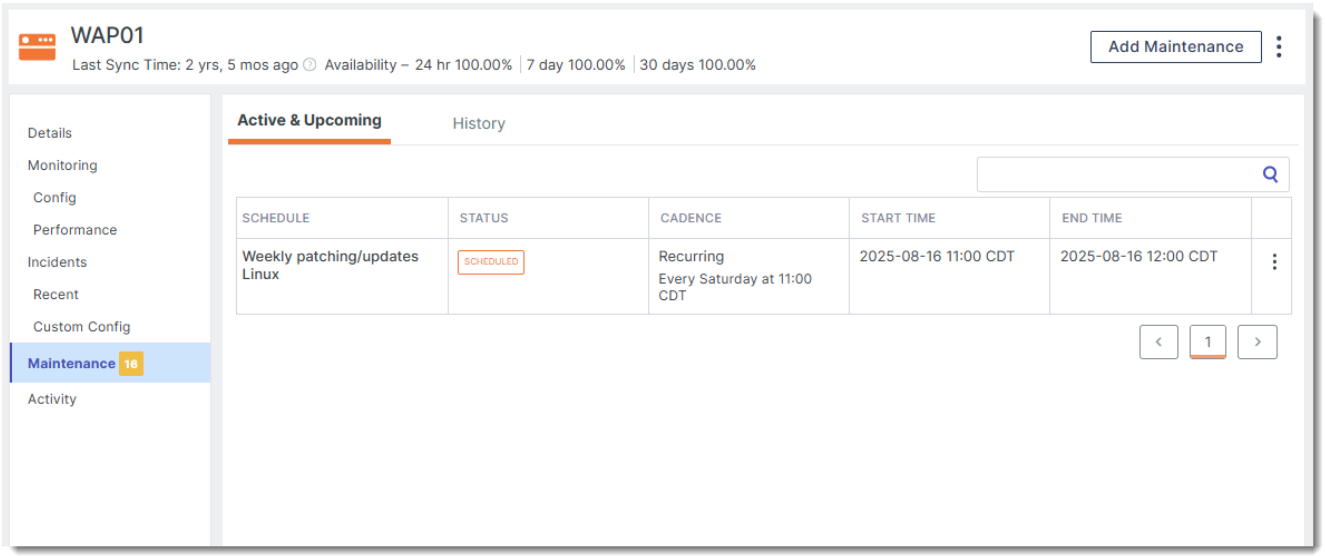
Custom config

The **Custom Config** section shows a list of created custom incident types. You can also add an incident type from this tab.

Maintenance

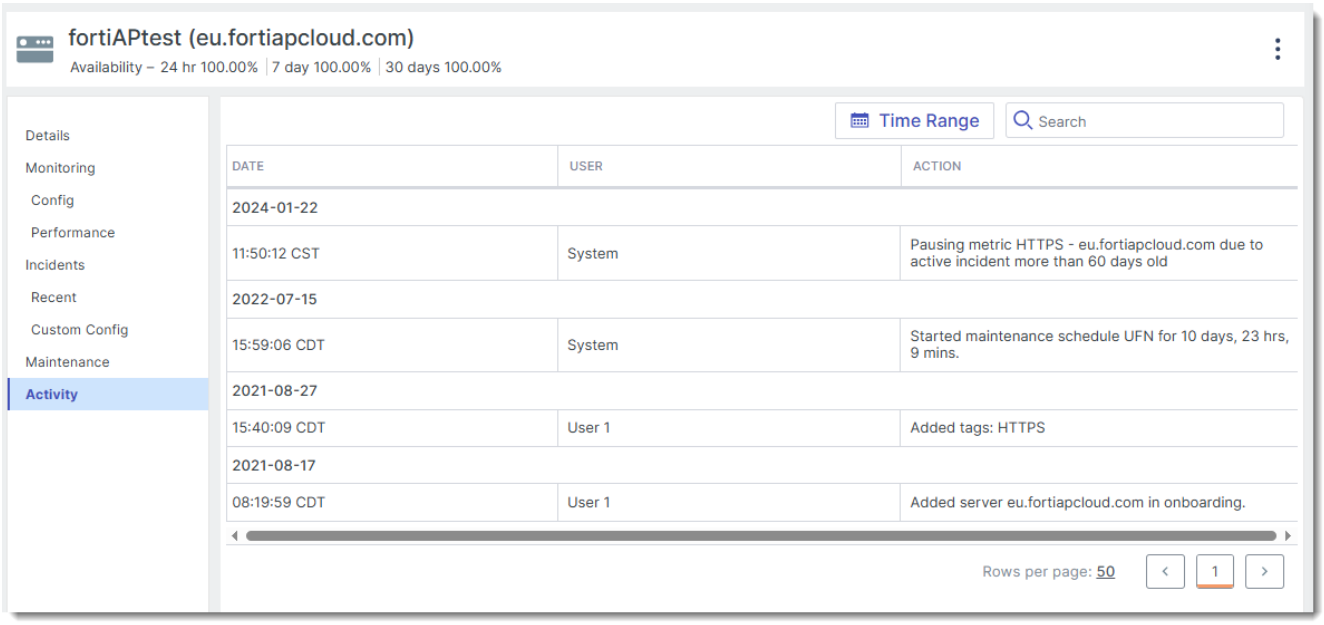
The Maintenance section shows all active and upcoming maintenance on the instance.

- Add Maintenance – Create a maintenance period for this instance.



Activity

The **Activity** tab provides a log of all actions performed on the instance. The initiator of the action, the date, and the action performed are detailed in the **Activity Log** table.



Assign Alert Timelines

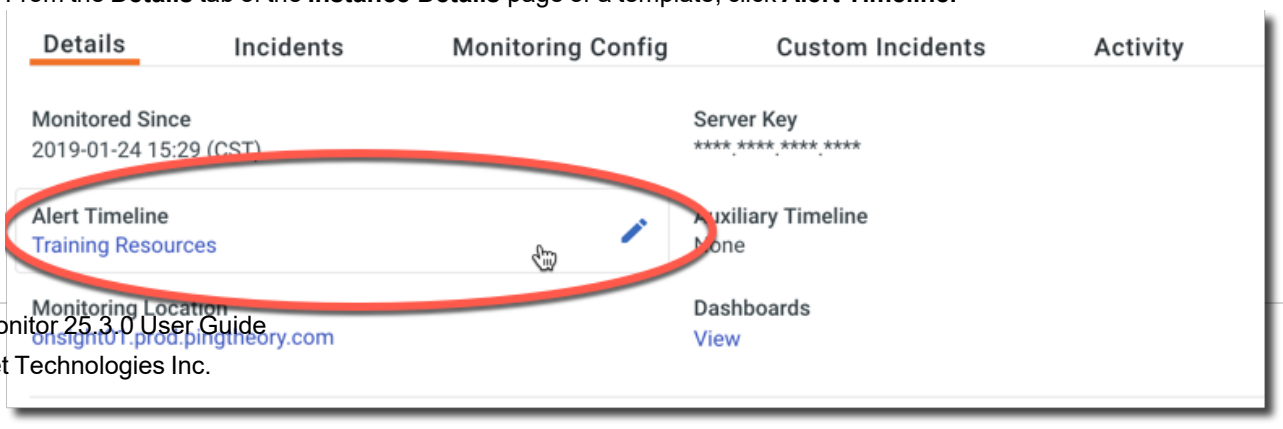
There are several types of Alert Timelines that you can assign to instances either from the **Instance Details page** or the Templates. The following Alert Timelines are available:

Primary Alert Timeline

The primary Alert Timeline used for the instance's incidents.

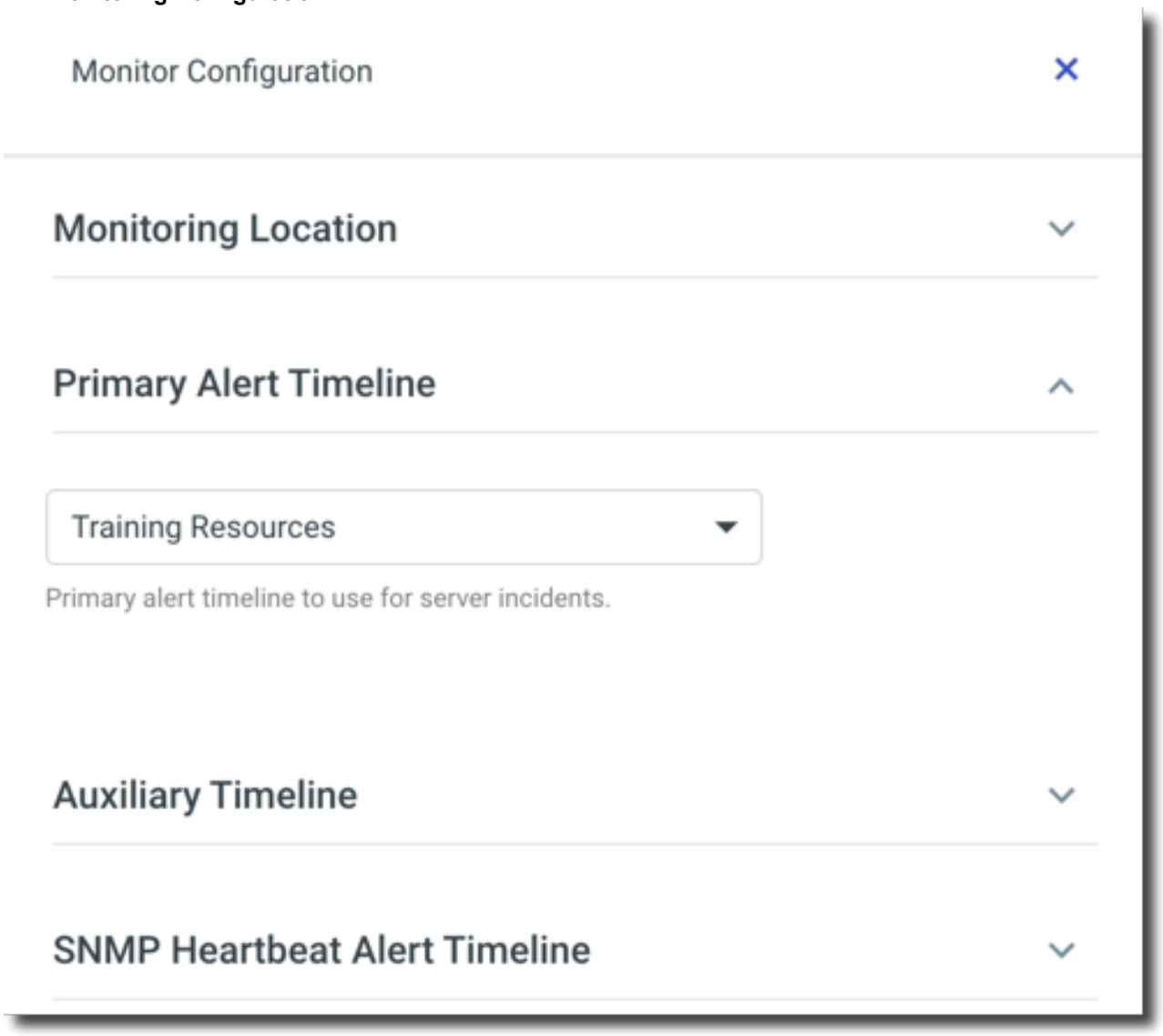
To assign a primary Alert Timeline, perform the following steps:

From the **Details** tab of the **Instance Details** page or a template, click **Alert Timeline.**



1.

The **Monitoring Configuration** drawer will slide out.



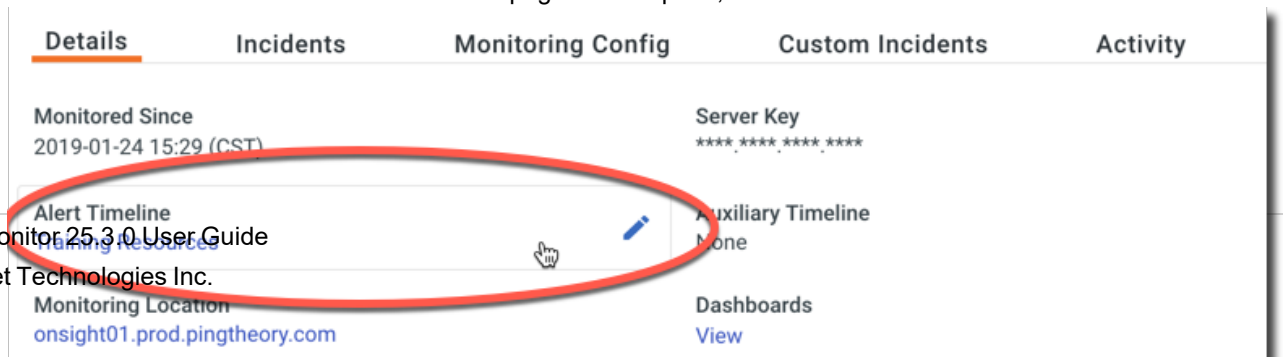
2.
3. From the **Primary Alert Timeline** module, choose an **Alert Timeline**.
4. Click **Save Changes**.

Agent Heartbeat Alert Timeline

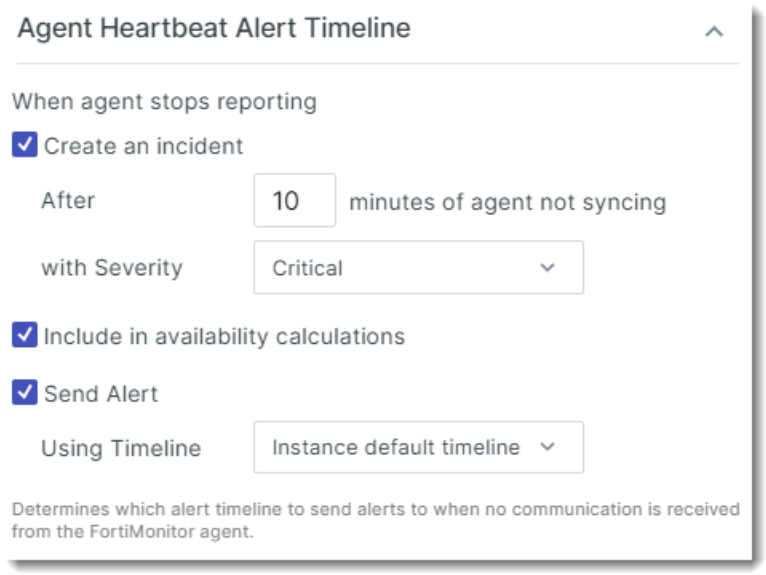
The Agent Heartbeat Alert Timeline determines which alert timeline to send alerts to when no communication is received from the FortiMonitor Agent for a specified amount of time.

To assign an Agent Heartbeat Alert Timeline, perform the following steps:

From the **Details** tab of the **Instance Details** page or a template, click **Alert Timeline.**



The **Monitoring Configuration** drawer will slide out.



2. From the **Agent Heartbeat Alert Timeline** module, configure the following:

- Create an incident.
- Set the time in minutes in which no communication is received before an incident is created.
- Severity of the incident.
- Select which Alert Timeline to send alerts to when no communication is received from the FortiMonitor Agent.
- Whether to include the incident in availability calculations.

4. Click **Save Changes**.

Metric Heartbeat Alert Timeline

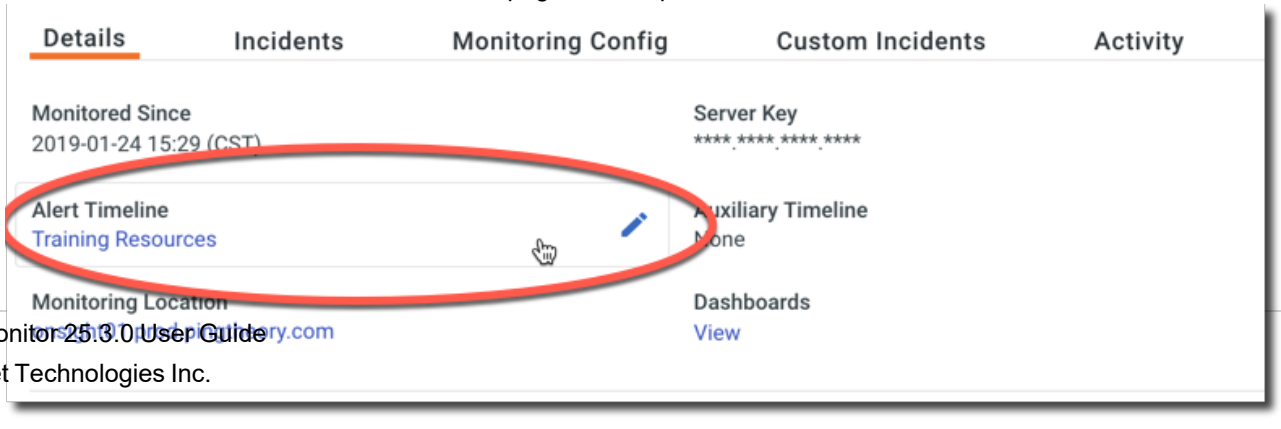
The Metric Heartbeat Alert Timeline determines which Alert Timeline to send alerts to when no communication is received from one or more metrics configured on the instance. Alerts are not triggered for:

- metrics that are paused
- metrics under maintenance
- metrics on an instance which have an active Agent Heartbeat incident

Currently, metric heartbeat checks are run every 60 minutes.

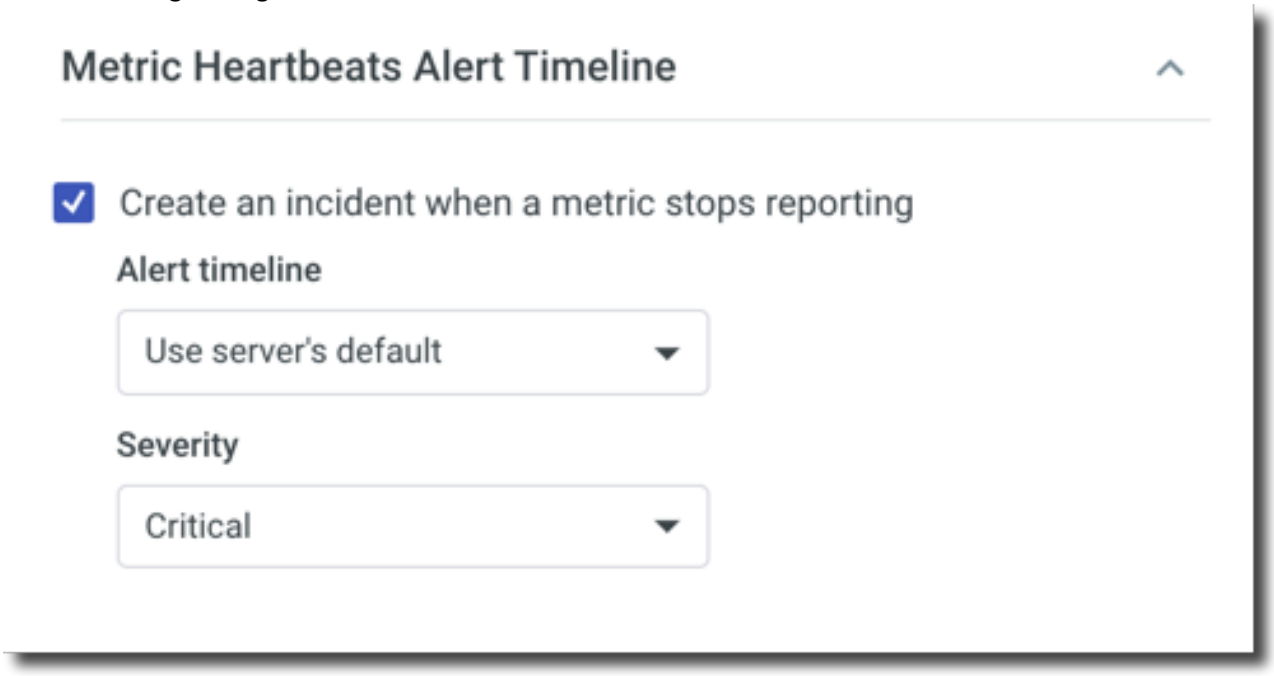
The **Metric Heartbeat Alert Timeline** feature is not enabled by default. If you do not see this feature, please contact our support team to have it enabled.

From the **Details** tab of the **Instance Details** page or a template, click **Alert Timeline.**



1.

The **Monitoring Configuration** drawer will slide out.



2.

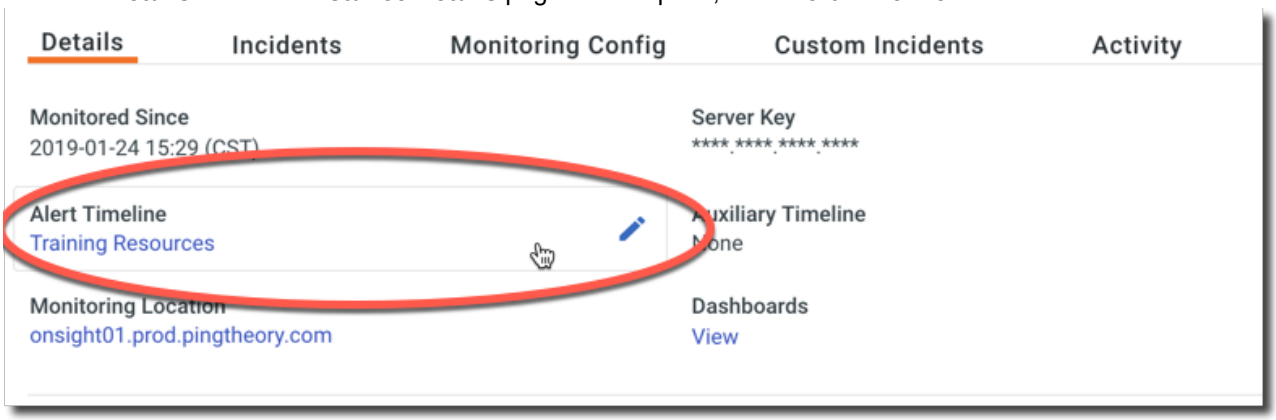
3. From the **Agent Heartbeat Alert Timeline** module, configure the following:
   - Enable or disable the **Create an incident when a metric stops reporting** option.
   - Select which Alert Timeline to send alerts to when a metric stops reporting.
   - Severity type
4. Click **Save Changes**.
5. After saving the changes, incidents will be created for any metric currently not reporting data.

SNMP Heartbeat Alert Timeline

The SNMP Heartbeat Alert Timeline determines which alert timeline to send alerts to when no communication is received from SNMP polling.

To assign an SNMP Heartbeat Alert Timeline, perform the following steps:

From the **Details** tab of the **Instance Details** page or a template, click **Alert Timeline.**



1.

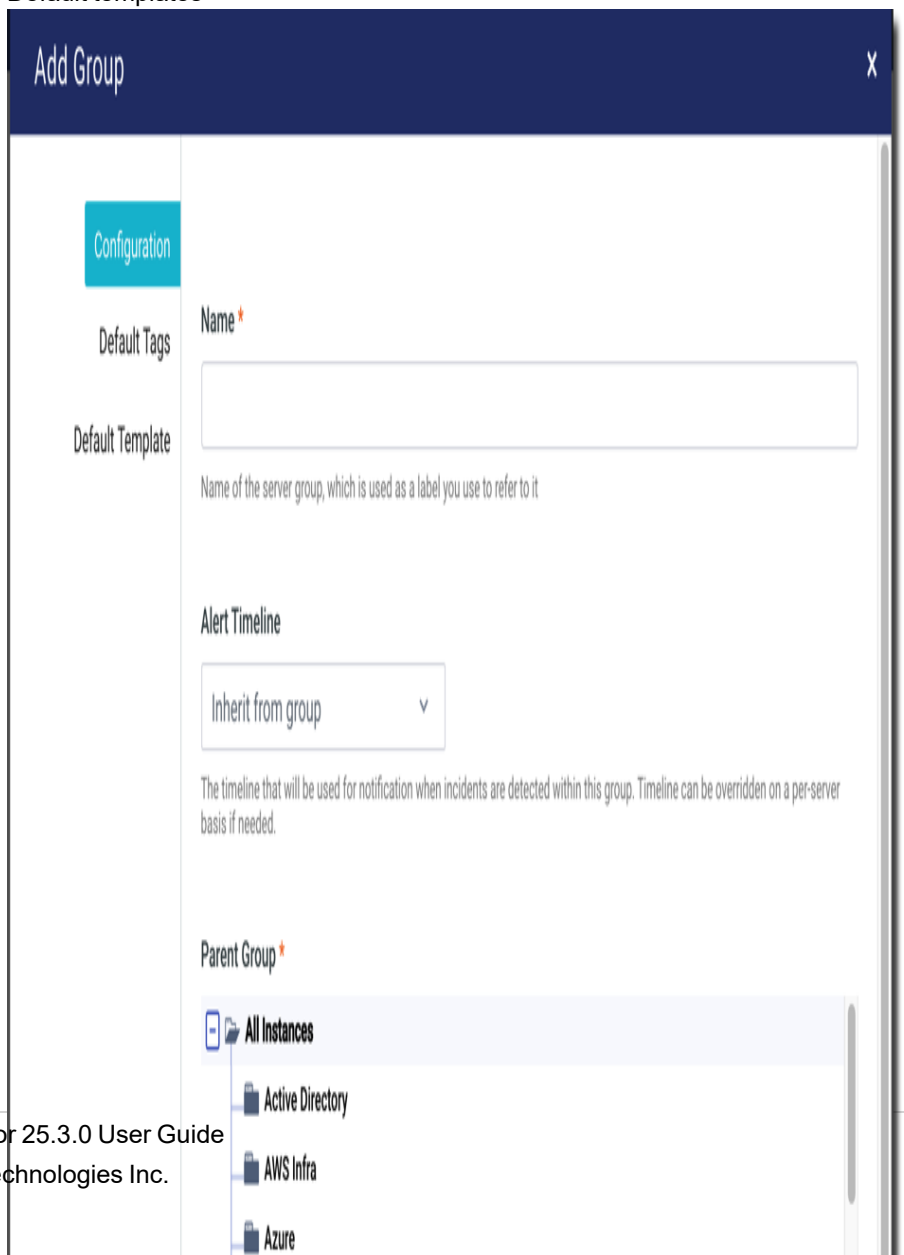
2. From the **SNMP Heartbeat Alert Timeline** module, configure the following:
   - Create an incident.
   - Set the time in minutes in which no communication is received from SNMP polling before an incident is created.
   - Select which Alert Timeline to send alerts to when no communication is received from SNMP polling.

4. Click **Save Changes**.

Create an Instance Group

To create an Instance Group, perform the following steps:

1. From the main navigation header, click **Add**.
2. From the **Config** section of the **Infrastructure and Resource Catalog**, locate **Instance Group** then click **Select**.
3. Configure the following details.
   a. Alert Timeline
   b. Default tags
   Default templates



4. Click **Save** to create the instance group. The instance group appears as a folder in the **Instance Tree**.

Multi-region support

Overview

Multi-region support provides the ability to specify the geographic location where your data is stored, ensuring compliance to local data regulations. This feature allows you to decide precisely where your data is handled.

This article explores the scope of multi-region support, key differences between regions, and outlines the necessary steps that you must follow to configure your environment.

Data residency

Data residency refers to the physical or geographical location where an organization's data is stored. Under data privacy laws like the GDPR, organizations may be required to store certain data within the country or region where it is collected.

Data residency vs. monitoring location

While data residency refers to the physical or geographical location where data is stored, monitoring location is the geographical area from which monitoring activities are conducted. Even if a user selects a specific region, they will still have access to all monitoring locations within FortiMonitor's global monitoring network.

Installing Agents, OnSights, and NetFlow appliances in a Multi-Region setup

During Agent, OnSight, or NetFlow appliance installation, you will find the proper region to use. The appliances will register and subsequently communicate directly with that region.

Always use the installer command line provided in the control panel for your specific region. This command line is tailored to include the necessary region reference.

Using an incorrect installer command line may result in the appliance failing to register with the correct region, leading to potential configuration issues and communication failures.

Regions

The following table describes the key differences between the two available regions.

Region	Aggregator URL	Data storage location
United States	rx.us01.fortimonitor.com	GCP US central region
Europe	rx.eu01.fortimonitor.com	GCP Frankfurt region

Note: Configuration, alerting, reporting, and user interfaces are strictly separated by region. Unless there's a compelling reason to use multiple regions, it is generally more efficient to stick to a single region.

Requirements

- Users must have a FortiCloud account. To create a FortiCloud account, see Creating a FortiCloud account.

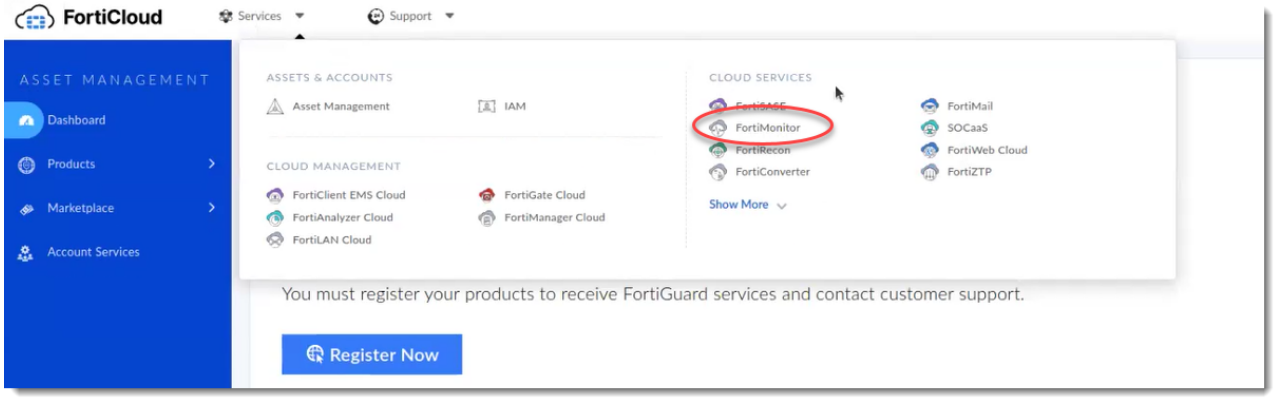
User access

As users are added in FortiCloud and granted access to FortiMonitor, they will be subsequently created in each of the FortiMonitor regions your account is using. Each user will be created with no ACL roles. These must be granted by an administrator before the new user account can be used.

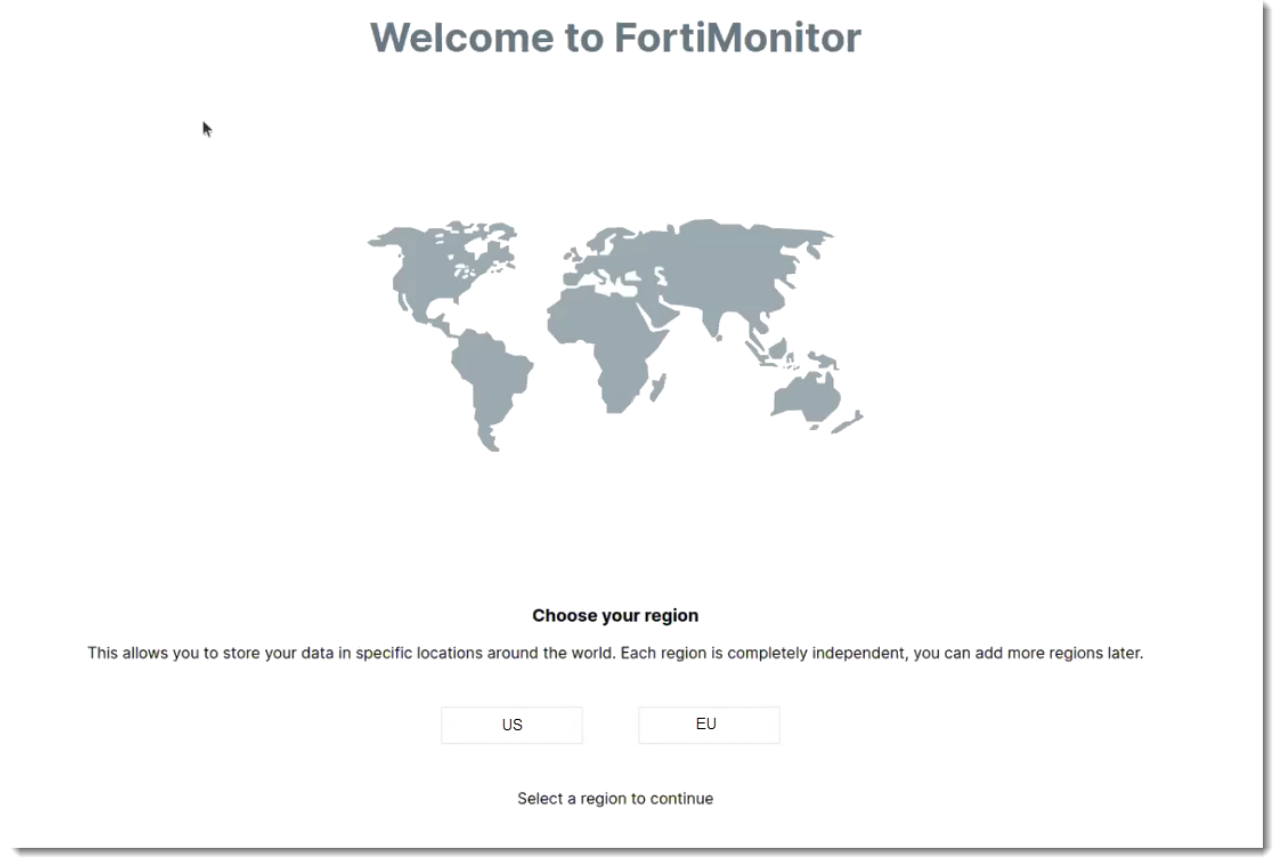
Users can have different ACL roles in different regions, meaning they don’t need to have the same level of access across all regions. Administrators have the flexibility to set local ACL settings to **No Access** in regions they prefer the user not to access, although the user’s account will still appear in those regions.

Selecting a region

1. Log in or create a FortiCloud account. To create a FortiCloud account, Creating a FortiCloud account.
2. After creating a FortiCloud account, go back to the main **FortiCloud** page.
3. Click **FortiMonitor** from the **FortiCloud Services** menu.



4. New users are presented with the region selection screen. Select a region.



5. After selecting a region, a user with the **Administrator** role will automatically be created for you on the selected region. You can now access FortiMonitor on your selected region.

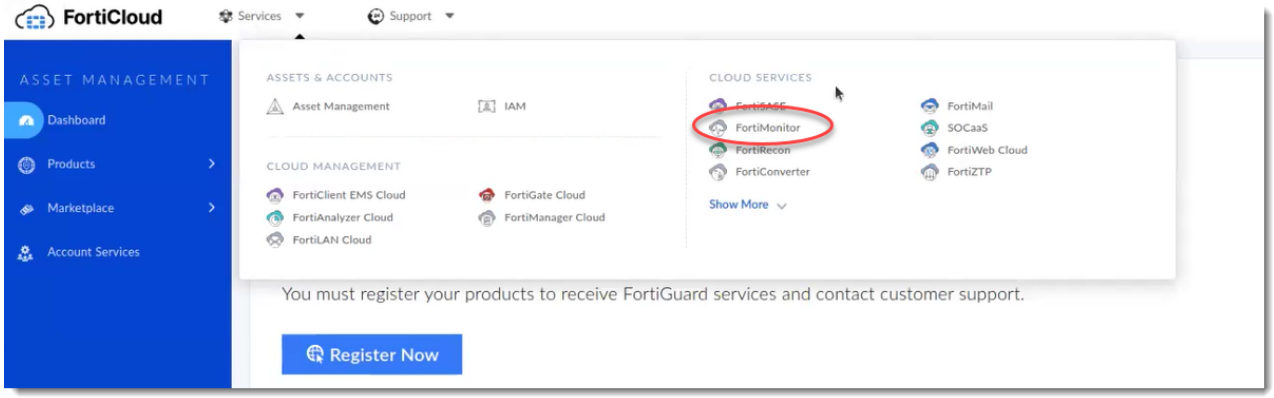
Adding a user

Add a user using FortiMonitor

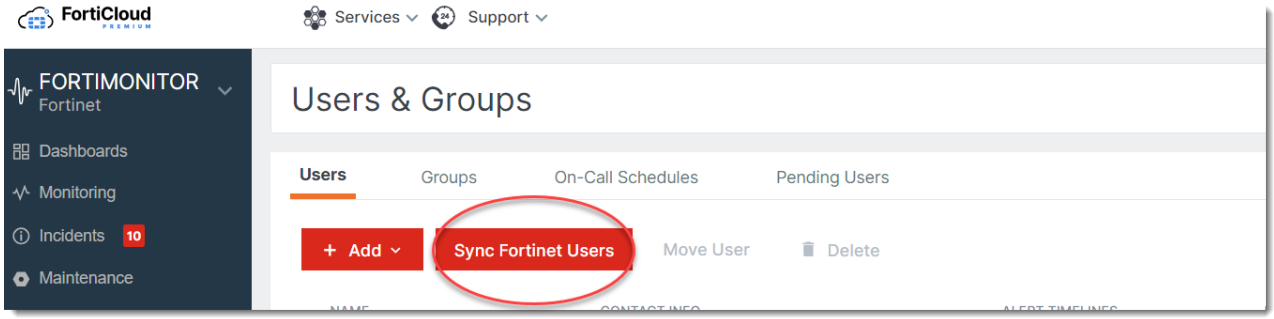
Follow the steps detailed in this article Add users using FortiMonitor on page 57. Note that users created using this method will only exist in the region where they were created.

Add a user using FortiCloud

1. Create a FortiCloud user by following the steps detailed in this page .
2. After creating a FortiCloud user, go back to the main FortiCloud page.
3. Go back to FortiMonitor by clicking **FortiMonitor** from the **FortiCloud Services** menu.

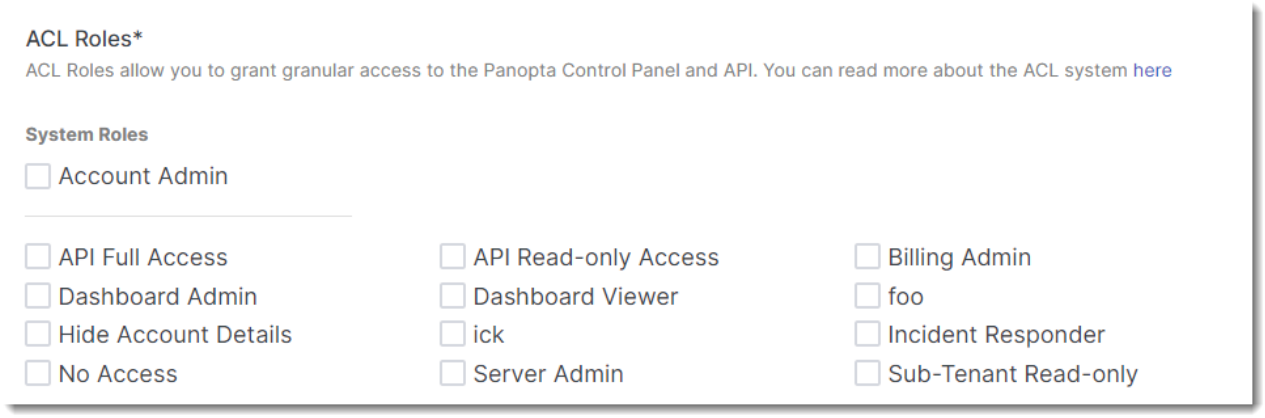


4. From the FortiMonitor control panel, go to **Teams & Activity** > **Users, Groups & On-Call**.
5. Click the **Sync Fortinet Users** button. This will add the created users to FortiMonitor.
Note: A discovery process is ran nightly to sync users created in FortiCloud to FortiMonitor. Use the **Sync Fortinet Users** button if you want to sync the created users immediately.



6. Users created in FortiCloud will have the **No Access** ACL role enabled by default. An Administrator must set the

permissions in FortiMonitor. For more information, see Access control.



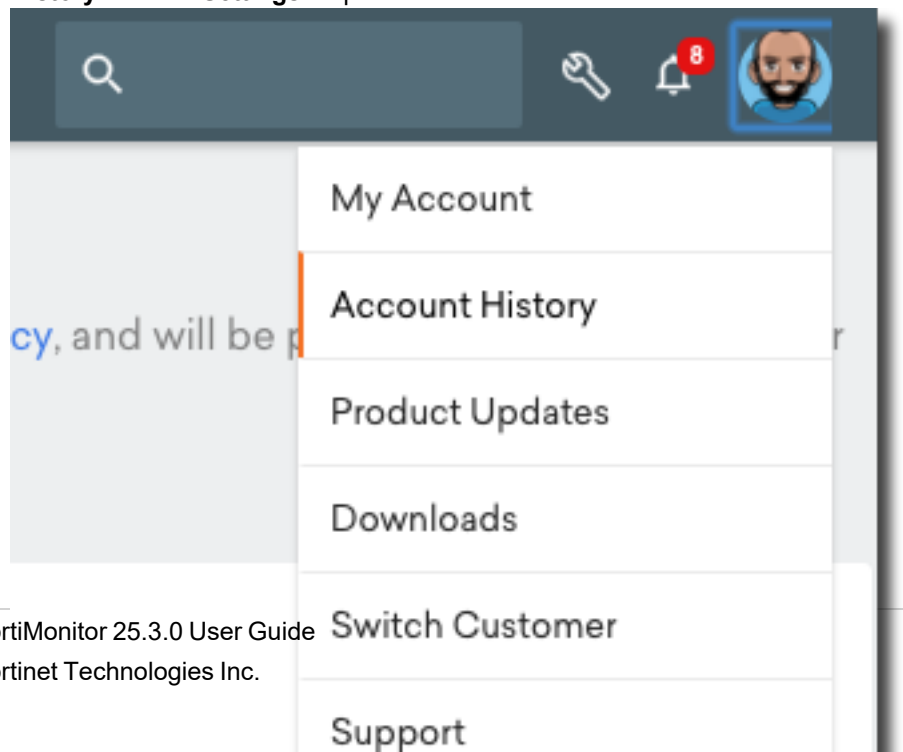
My Account

If you have any questions regarding your account that you cannot find answers to here, please reach out to your account manager or to our support team.

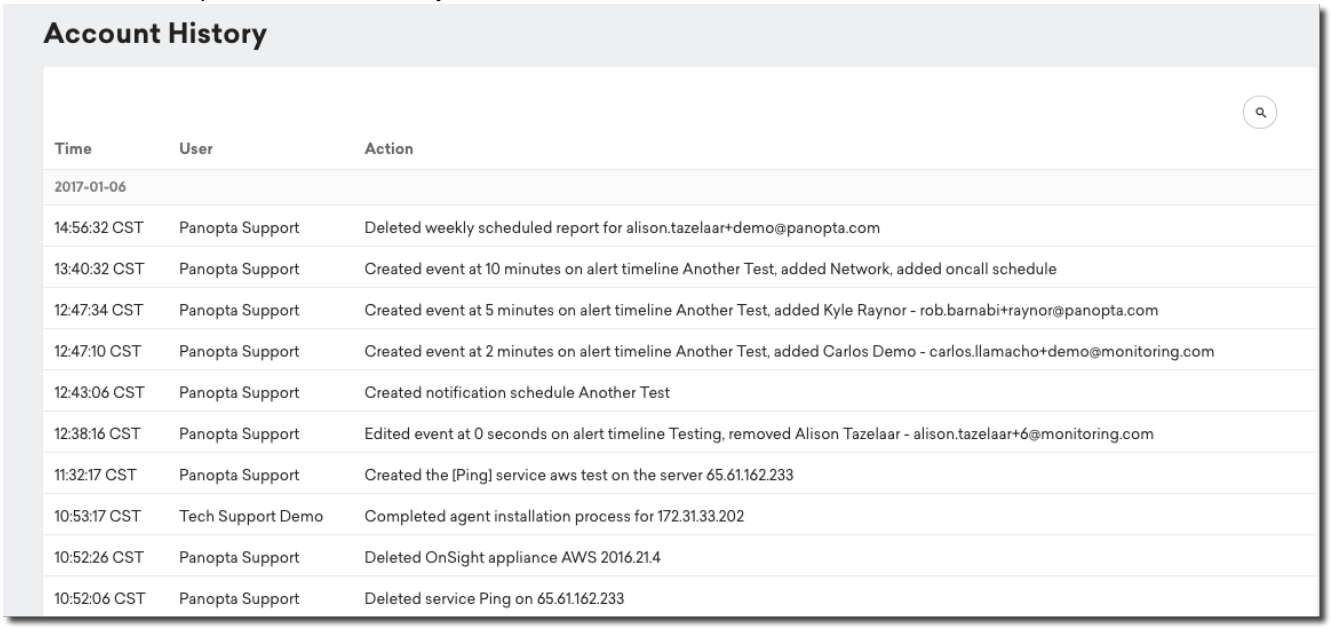
- Can I see what changes have been made to my account and by whom?
- Manage email preferences

Can I see what changes have been made to my account and by whom?

Yes, you can, your account history keeps track of all of this information. To look at your account history select **Account History** from the **Settings** dropdown.



On this new page, you can filter by user or by the action taken. Just enter whatever you would like to filter your search by into one or both input boxes and click **Update**.



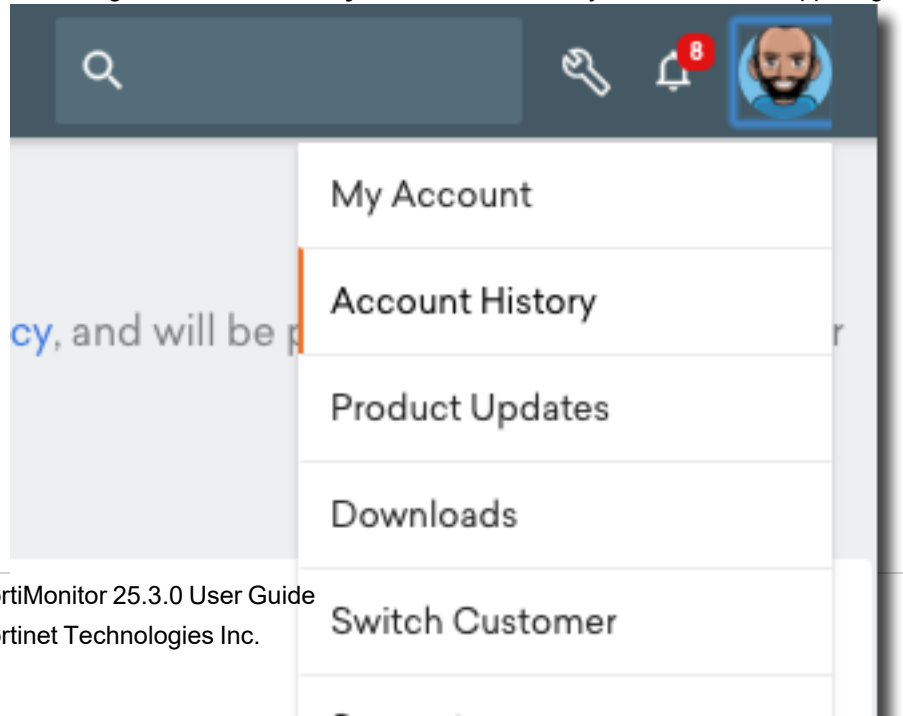
For example, if you would only like to see what changes were made to a notification schedule you can just enter that schedule's name into the action filter and click **Update**.

The account history can be a wonderful way of making sure that everyone is on the same page as far as configuration is concerned. If you notice a surprising change to your configuration, the account history will help you find out who you need to talk to about these changes.

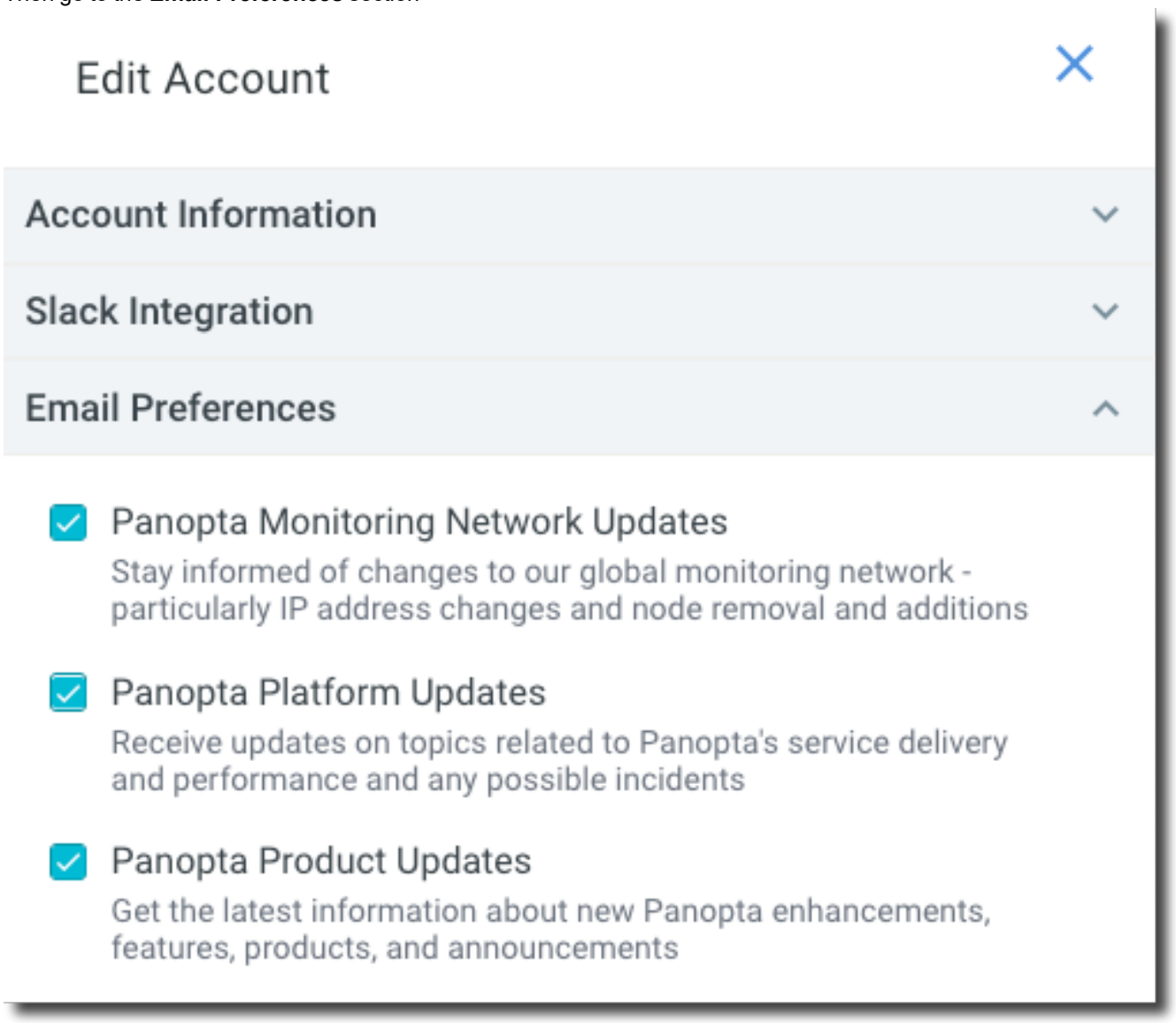
Manage email preferences

Beyond emails related to incidents, alarms and reports, FortiMonitor oftentimes will send out emails related to our weekly feature enhancements, production system issues and changes to our public monitoring node network.

To manage this, click on the **My Account** link under your avatar in the upper right-hand corner.



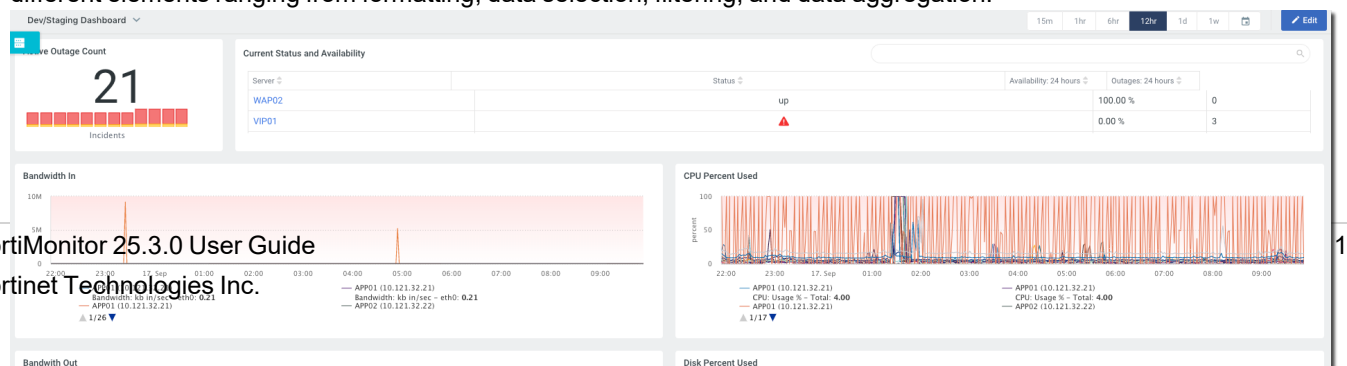
Then go to the **Email Preferences** section



From here, you can subscribe or unsubscribe from the various lists based on what you would like to stay informed about.

Dashboards

Dashboards are powerful mechanisms for displaying and visualizing the various streams of data that are being captured by the FortiMonitor Monitoring Cloud. Configuring a dashboard is easy and uses our simple drag and drop builder for widgets in order to achieve the layout you require. In addition, each of the widgets has several options for controlling different elements ranging from formatting, data selection, filtering, and data aggregation.

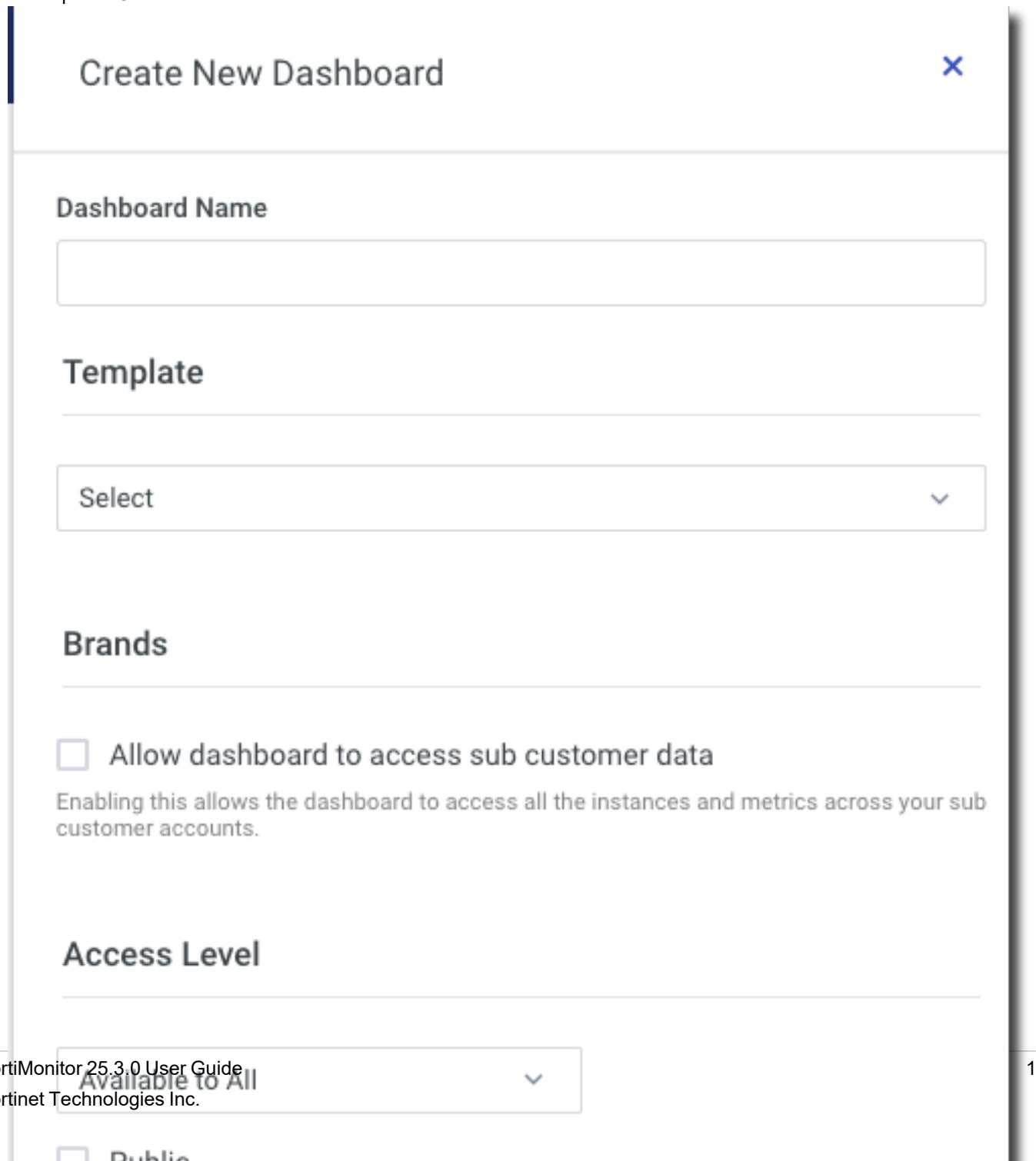


Create a dashboard

There are two ways to create a new dashboard:

- From the navigation menu, select **Dashboards** > **My Dashboards**. Click the name of the dashboard then click **Create New Dashboard**.
- Click **Add** from the navigation menu. From the **Alerting & Reporting** section of the catalog, select the **Dashboard** card.

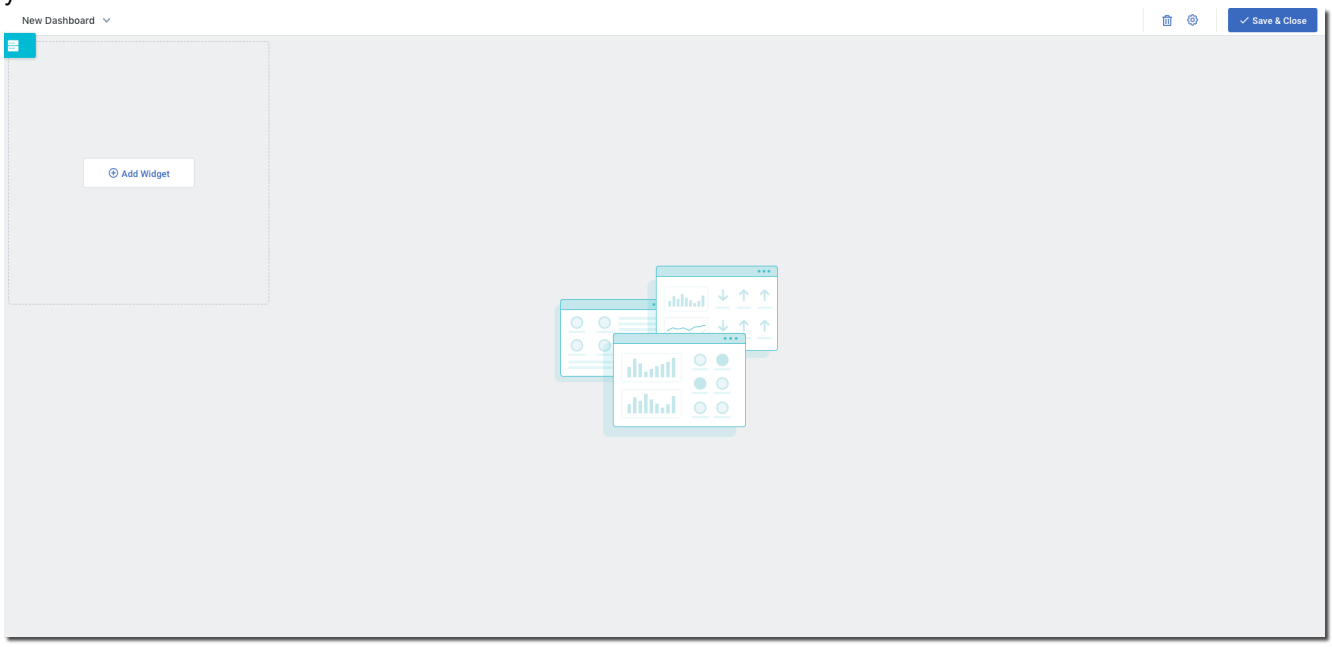
Doing either of the following will show the Ceate a Dashboard drawer. From here you can then set the basic properties for your new dashboard, like name, tags, access level, templates, and whether this dashboard should be made available with a public URL.



The **Access Level** options allow you to limit which of your users can see which dashboards.

- **Available to all** - All users with a login and who have access to the dashboards menu can see these Dashboards
- **Limited by tag** - Dashboard access will be limited by which tags are set on this dashboard and tags set on a user. The tag filter uses `OR` logic when matching dashboard tags and user tags
- **Only for me** - These dashboards will only show under your login

Once you select **Create Dashboard**, a blank dashboard canvas will be created. Select **Add Widget** to start building your dashboard.



Configure dashboard widgets

We have provided extensive details on each widget and their options here.

Instance and metric tags

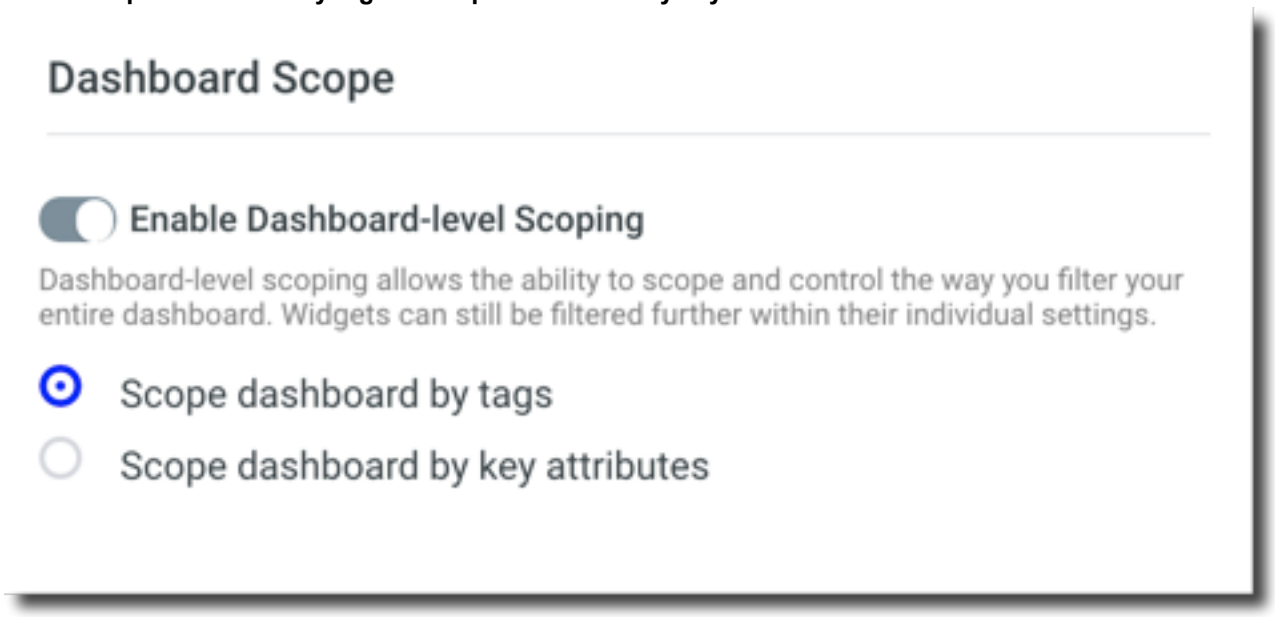
If you are not already using tags to further segment and categorize your infrastructure, dashboards are a great reason to start. Every widget provides options for dynamically displaying the appropriate infrastructure and metrics by tag, rather than hardcoded selections. This allows your infrastructure to expand and contract as needed without requiring constant changes on what is being shown in your live dashboards. For more information on tagging, read Tags.

Dashboard-level scoping

Dashboard-level scoping allows you to control the way you filter your entire dashboard. You can scope your dashboard by **tags** or by **key attributes**. Individual widgets can then be filtered further within their individual settings.

To enable dashboard-level scoping, perform the following steps.

1. From the dashboard where you want to apply dashboard-level scoping, click **Edit**.
2. Click the **gear icon**. The **Dashboard Settings** drawer will slide out.
3. Under **Dashboard Scope**, enable dashboard-level scoping.
   Select **Scope dashboard by tags** or **Scope dashboard by key attributes**.



4.
   a. If you select **Scope dashboard by key attributes**, click **Add new Value** to add a key attribute.
   b. Select a value from the dropdown list.
   c. Repeat steps **a** and **b** to add more attributes.
5. Click **Save**.

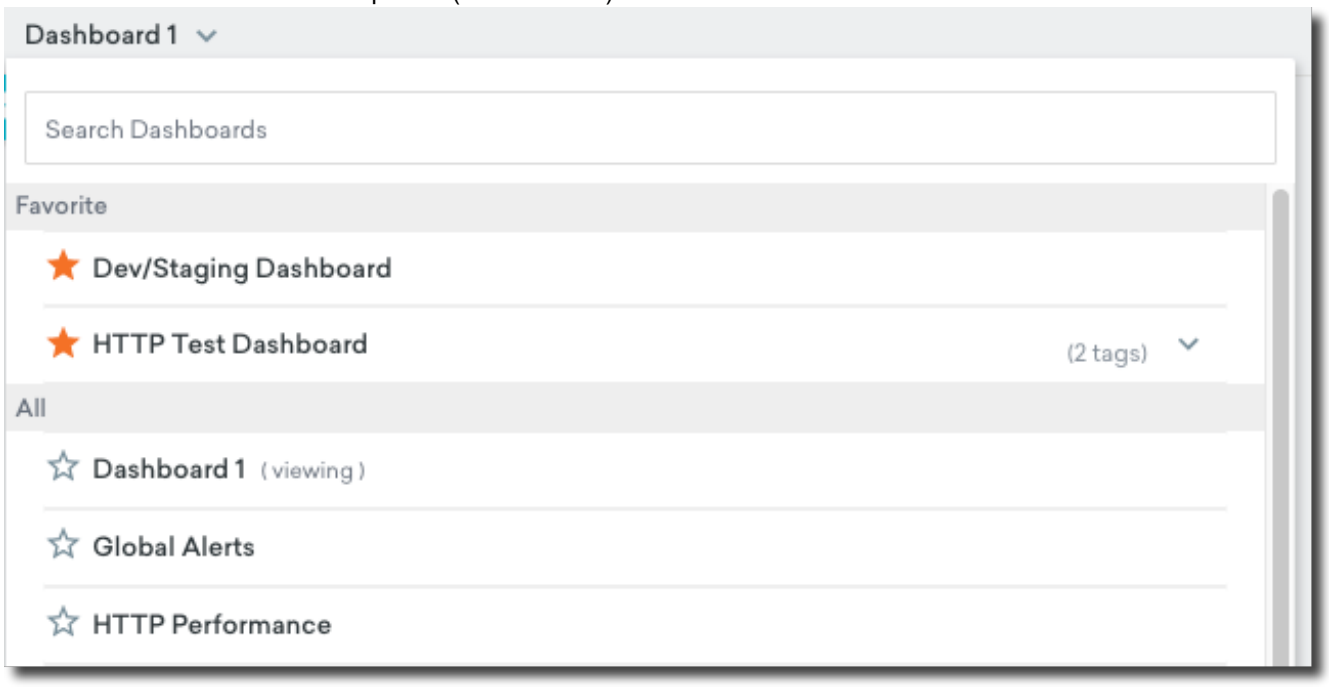
If you have selected **Scope dashboard by tags**, the field **Scope dashboard by tags** will become available. Enter any number of tags to filter the dashboard.

If you have selected **Scope dashboard by attributes**, a dropdown field containing all of the values stored in the attribute will become available. Select one or more values to filter the dashboard.



Favorites and search

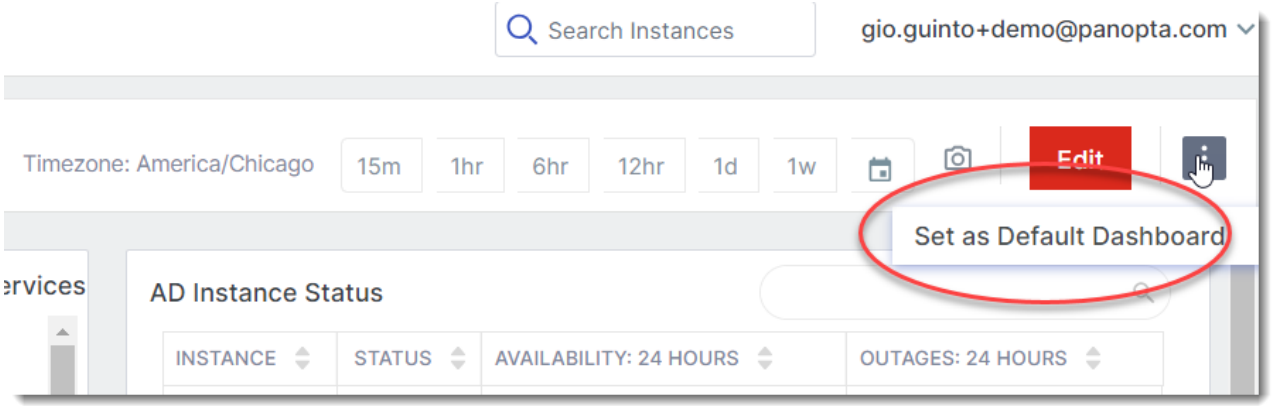
Setting favorite dashboards makes them easier to access by prioritizing them at the top of the dashboard menu and within the dashboard selector dropdown (shown below).



Setting a default dashboard

To set a default dashboard, perform the following steps:

1. Open a dashboard.
2. Click the **3-dot menu** then select **Set as Default Dashboard**.



Dashboard widgets

One of the major strengths of FortiMonitor's Dashboards is the catalog of widgets to pick from. Each widget provides a different suite of options for formatting, data selection and data aggregation. See the following sections for ,more information on each widget.

Active Incident Counter

The **Active Incident Counter** widget displays the count of all active incidents in your infrastructure, based on the configuration of the widget. You can filter which incidents are included based on specific metric types or severity.

Use case

The **Active Incident Counter** widget is useful when:

- Monitoring the total number of active incidents in your infrastructure
- You want to see the number of specific incidents by metric or by severity

Examples

All active incidents

The widget shows the total number of incidents in your infrastructure.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Severity levels	Select which severity level will be included in the calculated number of incidents. For example, selecting Critical displays the number of all critical incidents. **Values**: • Critical • Warning • All
Metrics	Add the metrics that you want to include in the calculation of total incidents.
Filter metrics	You can also use the filters to fine-tune your search results. **Values**: • No filter • By tags • By option string
Instances	Filter instances by the following: • No filter • Tags • Group • Name The instances selected here will be included in the counter when incidents occur.
Hide incidents	Choose which incidents you do not want to include in the number of incidents displayed on the widget. **Values**: • Hide acknowledged incidents • Hide incidents under maintenance
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Active Incident Details

The **Active Incident Details** widget provides the details of each active incident included in your filter. The configurable columns for different data points allow you to display only the columns you care about. The ordering of the columns can be set along with which column is used for sorting (the carrot). Like most widgets, you can scope the results to specific instances and metrics.

The **Active Incident Details** widget is limited to displaying incidents that are up to 1 year old. Any incidents older than 1 year will not appear in the widget.

Use case

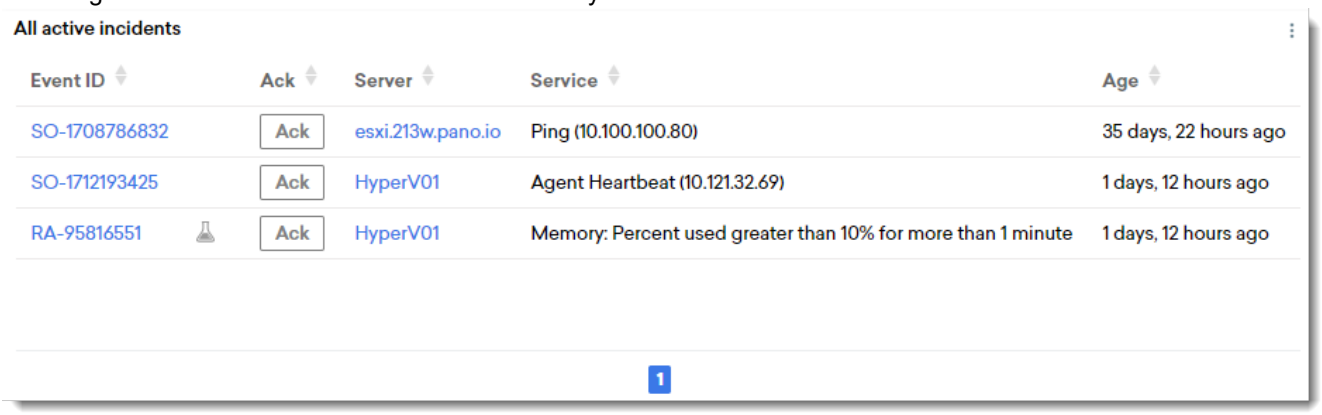
The **Active Incident Details** widget is useful when:

- You want to see details about the incidents in your infrastructure
- You want to see details about the active incidents on specific metrics, instances, or based on the incident's severity

Examples

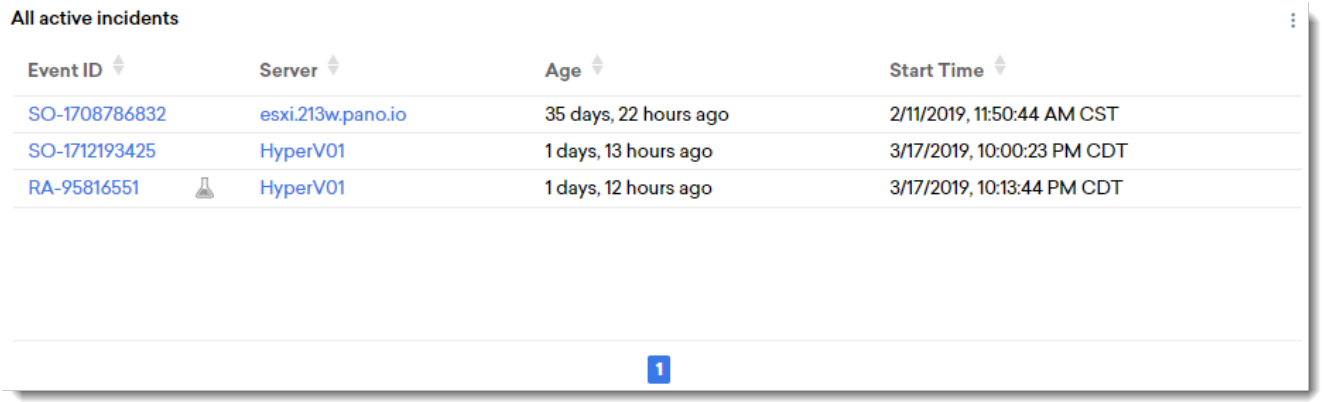
All active incidents

The widget shows a detailed list of all the incidents in your infrastructure.



All active incidents (with Age and Start Time)

In this example, the widget is customized to show only the age and start date of all incidents.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Filter Instance by	Filter instances by the following: • No filter • Tags • Group • Name The details of the instances selected here will be displayed on the widget.
Metrics	Add the metrics that you want to show up on the widget.
Filter Metrics	You can also use the filters to fine-tune your search results. **Values**: • No filter (default) • By tags • By option string
Incident Severity	Select the incidents to display on the widget based on severity level. For example, selecting Critical shows only the details of critical incidents. **Values**: • Critical • Warning • All
Data Points	Select which data points you want to show or hide. **Values**: • CounterMeasures • Ack • Server • FQDN • Service • Age • Start Time • Severity • Assigned To
Hide Incidents	Choose which incidents you do not want to include in the widget. **Values**: • Hide acknowledged incidents

	• Hide incidents under maintenance
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Advanced Line Graph

Like the **Line Graph** widget, you can use the **Advanced Line Graph** widget to represent a time series metric, but with the addition of a table that provides the following supplemental details:

- Value
- Minimum value over the defined time range
- Maximum value over the defined time range
- Availability
- Number of incidents related to the metric

Metrics can be for a single instance or spread across multiple instances. Like most widgets, you can scope the results to specific instances and metrics using a variety of filtering options.

Use case

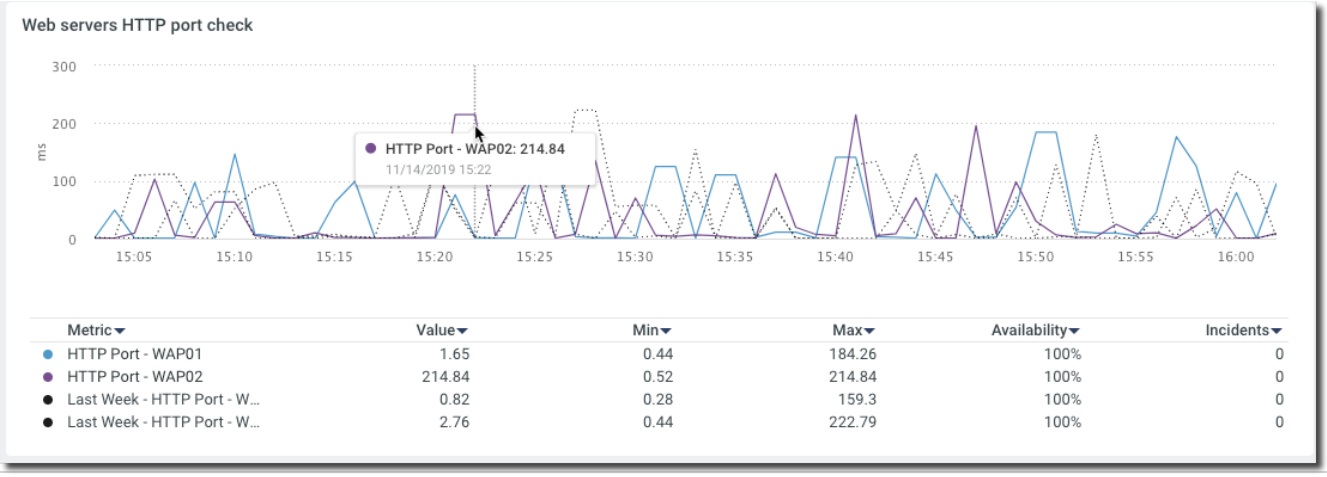
The **Advanced Line Graph** widget is useful when:

- Identifying performance trends
- Comparing period over period performance
- Comparing individual metric values
- Exact values are required

Examples

HTTP response time comparison

In this example, the Advanced Line Graph displays the HTTP response time values of all the Linux web servers. The dotted line represents the historical value of the selected metric. Below the line graph, the data is presented in the form of a table.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Block Title	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Metrics	Add the metrics that you want to be visualized in the Advanced Line Graph.
Filter metrics	You can also use the filters to fine-tune your search results. **Values**: • No filter • By tags • By option string
Filter Instance by	Filter instances by the following: **Values**: • No filter • Tags • Group • Name
Time Range	The time range for the trendline. **Values**: • Last Hour (Default) • Last 15 minutes • Last 6 hours • Last week • Last month • Last year
Formatting	Select which detail you want to hide from the chart. **Values**: • Hide Server Name • Hide Data Point Units • Hide Server FQDN • Hide Metric Type • Hide Option Name
Graph Options	Additional options for the line graph. **Values**: • Label points • Show Crosshairs • Aggregate Metrics

	• Comparison
Metric Calculation	Select if you want to display the metric data aggregated over the selected time range. The following values can be selected to calculate the aggregated data: **Values**: • Average (default) • Min • Max • Sum **Note**: The **Metric Calculation** option only appears if you select the **Aggregate metric** checkbox.
Comparison	Compare the trendline to its historical values. A dotted line represents historical values. **Values**: • None • Same time yesterday • Same time last week • Four weeks prior
Table Rows	Select the maximum number of rows for the table.
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Aggregated Availability

The **Aggregated Availability** widget combines the availability of multiple instances into a single value. Choose from different aggregate functions, like Average, Min or Max throughout a specific time range.

Use case

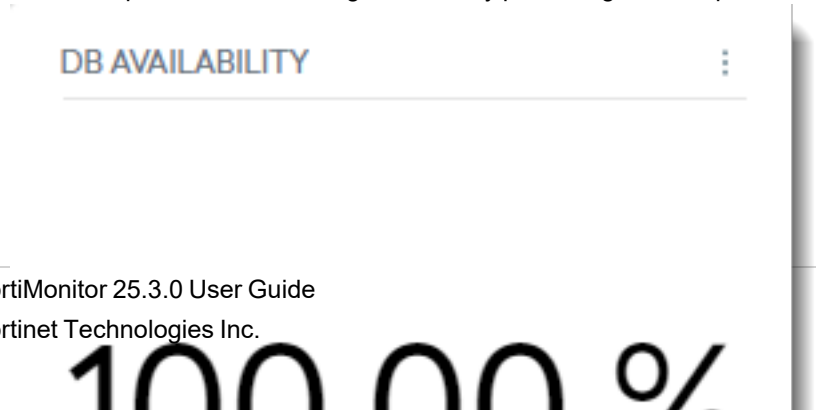
The **Aggregated Availability** widget is useful when:

- You want to display the aggregate availability percentage of multiple instances.

Examples

Average availability

This example shows the average availability percentage of multiple database instances.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Instances	Filter instances by the following: **Values**: • No filter • Tags • Group • Name The aggregated values will be collected from the selected instances.
Metrics	The selected metric's aggregated value will be displayed on the widget.
Filter metrics	You can also use the filters to fine-tune your search results. **Values**: • No filter • By tags • By option string
Time Range	Shows the aggregated availability percentage of a group of instances based on the selected time range. **Values**: • Last 24 hours • Last week • Last month • Last year
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Aggregated Metric

Use this widget to combine multiple metric values of the same type into a single value. Choose from different aggregate functions, like Average, Min or Max throughout a specific time range.

Use case

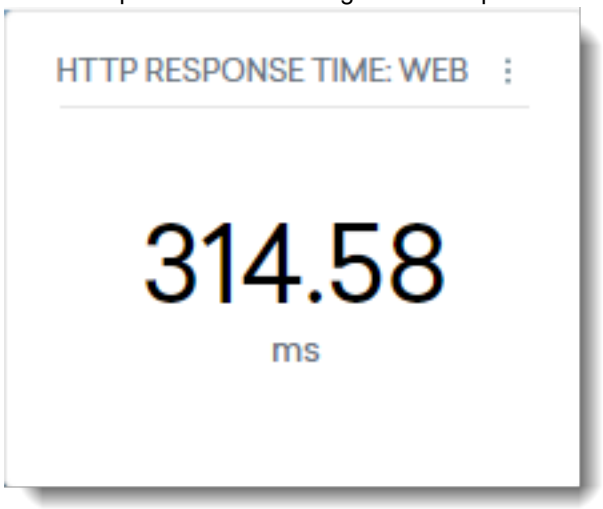
The **Aggregated Metric** widget is useful when:

- You want to display the aggregate value of multiple metrics of the same type

Examples

Average HTTP response time

This example shows the average HTTP response time across multiple web servers.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Metrics	The selected metric's aggregated value will be displayed on the widget.
Additional Metric Filters	You can also use the filters to fine-tune your search results. **Values**: • By tags • By option string
Instances	Filter instances by the following: • No filter • Tags • Group • Name The aggregated metric values will be collected from the selected instances.
Aggregation	Aggregated data can be calculated based on the following values: **Values**: • Average (Default)

	• Min • Max • Sum
Time Range	Shows the live or aggregated metric value of a metric based on the selected time range. **Values**: • Live • Last hour • Last 15 minutes • Last 6 hours (default) • Last 12 hours • Last 24 hours • Last week • Last month • Last year
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Bar Chart

The **Bar Chart** allows you to visually represent metric levels across multiple instances. Like most widgets, you can scope the results to specific instances and metrics using a variety of filtering options. Select whether to show live metric values or aggregates over a certain period. Also, you have options for controlling the X-axis legend and the orientation of the chart.

Use case

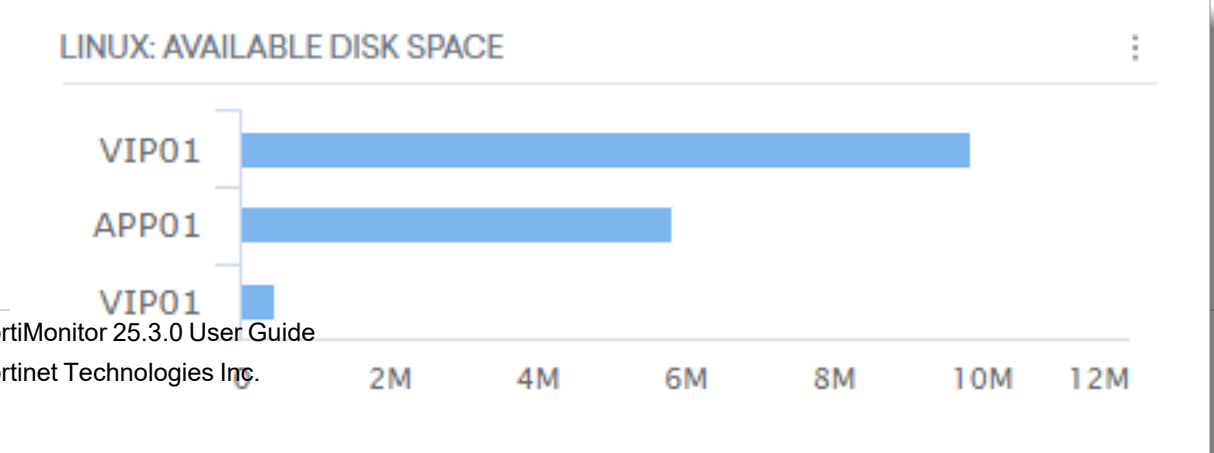
The **Bar Chart** widget is useful when:

- Comparing non-time series metrics across multiple instances

Examples

Available disk space

In this example, the available disk space for selected Linux servers is represented by the blue bars in the graph.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Metrics	Add the metrics that you want to be visualized in the bar chart.
Additional Metric Filters	You can also use the filters to fine-tune your search results. **Values**: • By tags • By option string
Instances	Filter instances by the following: • No filter • Tags • Group • Name The metrics of the selected instances will be visualized in the bar chart.
Time Range	Shows the metric levels of selected instances based on the selected time range. **Values**: • Latest value • Last hour • Last 15 minutes • Last 6 hours • Last 12 hours • Last 24 hours • Last week • Last month • Last year
Formatting	Select which detail you want to hide from the chart. **Values**: • Hide Server Name • Hide Units • Hide Server FQDN • Hide Metric Type • Hide Option Name • Float Legend Right
Orientation	Select the orientation of the bar graph. **Values**:

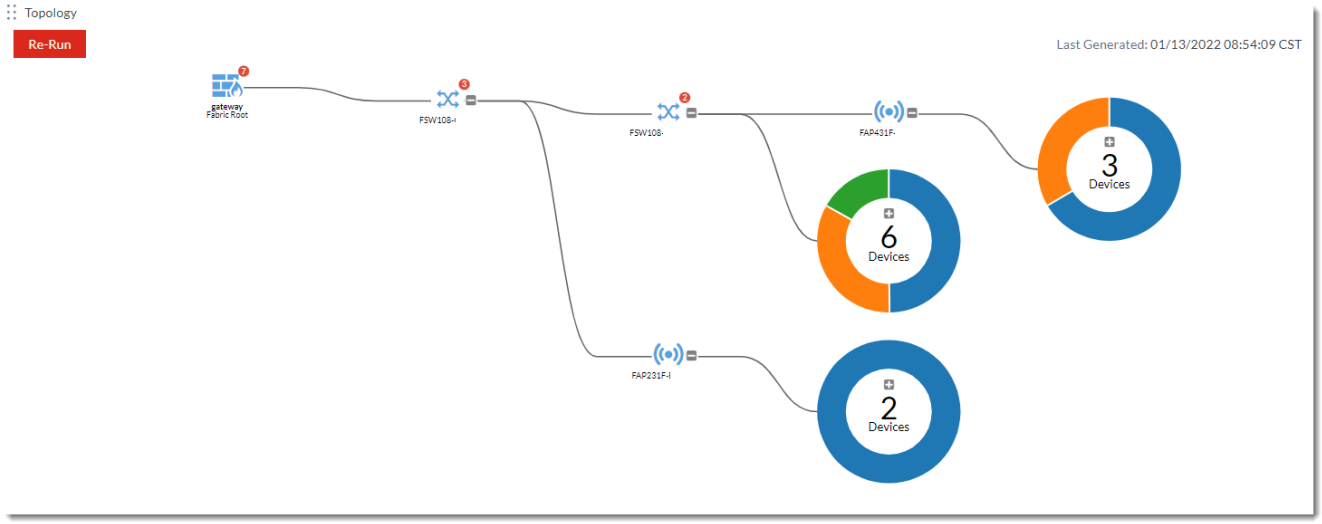
	• Rows • Columns
Limit visible items by performance	Enabling the option allows you to set a limit to the number of items displayed in the bar chart.
Number of items to display	The number of items displayed in the bar chart.
Sort Items	Sort by: • Display highest first • Display lowest first
Aggregate metrics and Metric calculation	Select if you want to display the metric data aggregated over the selected time range. Aggregated data can be calculated based on the following values: **Values**: • Average (default) • Min • Max • Sum **Note**: The **Metric calculation** option only appears if the **Aggregate metric** checkbox is selected.
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Fabric topology widget

The Fabric Topology widget shows the physical structure of your Fabric environment, including all connected devices and their interconnections. Before creating the widget, make sure that your active Security Fabric is connected to FortiMonitor.

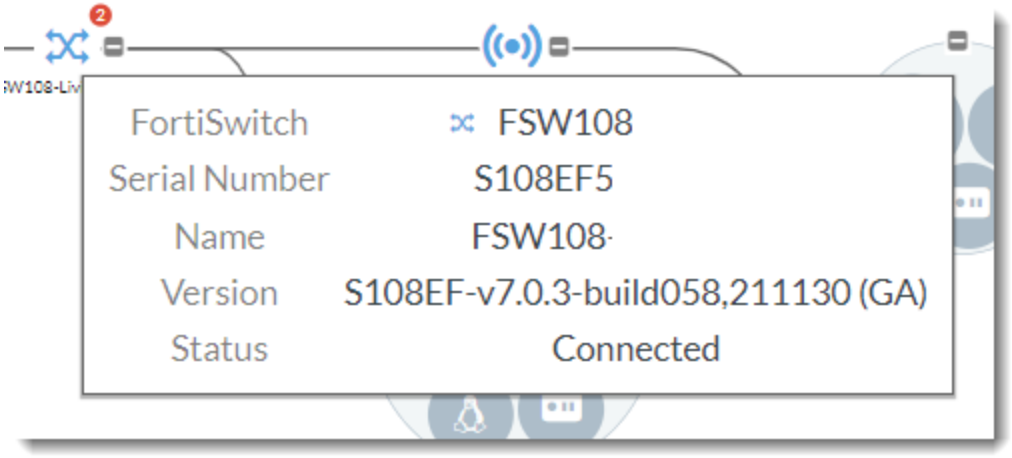
Note: For FortiManager-based integrations, the widget may only display a single FortiGate unless you have set up a Fabric group on the FortiGates. If Fabric is not set up, the widget will not work.

For more information on Dashboards, see Dashboards.



On hover

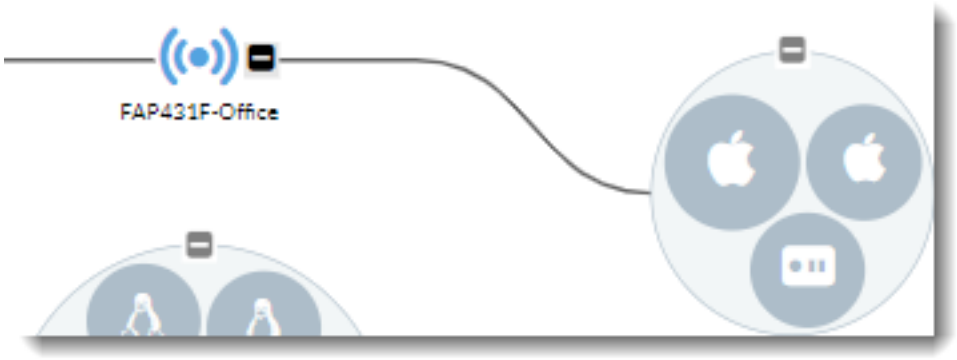
Hover the cursor over a device icon to display information about the device.



Some devices may display a small icon with a colored number. This icon represents the number of security ratings recommendations or warnings for that device. The circle color indicates the severity of the highest security rating check that failed. See Security rating for more information.

Expand and collapse

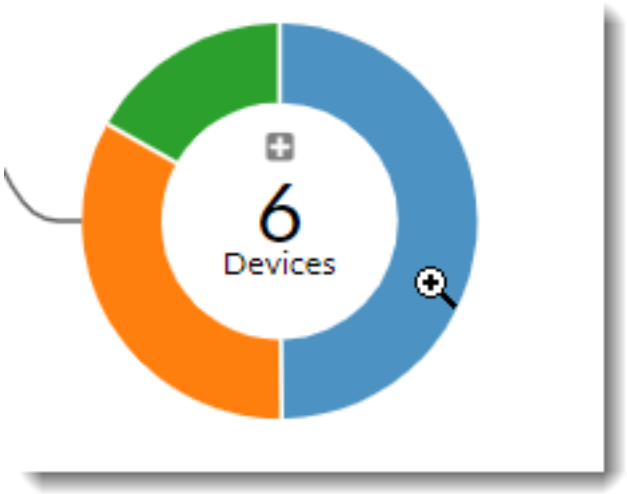
Click the **+** or **-** icon to expand or collapse the groups connected to a device. You can also expand individual endpoint groups.



Additional information on each endpoint device can also be displayed on hover.

Device count

Device counts are represented by a donut chart where the total number of endpoints in the group is displayed at the center.



Each sector of the chart represents a different endpoint OS.

Sector color	OS
Green	FortiMail
Red	FortiManager
Orange	Linux
Blue	Other

Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Fabric Root	Name or ID of the Fabric Root in FortiMonitor.

Gauge

The **Gauge** widget allows you to visually represent a specific metric level for a single instance where there is a logical upper bound. For metrics based on percentages, the upper bound will automatically set to 100%. For non-percentage based metrics, you will need to enter a value. If the metric level has breached a configured threshold, the gauge color will change from green to either red/orange, based on the severity.

Use case

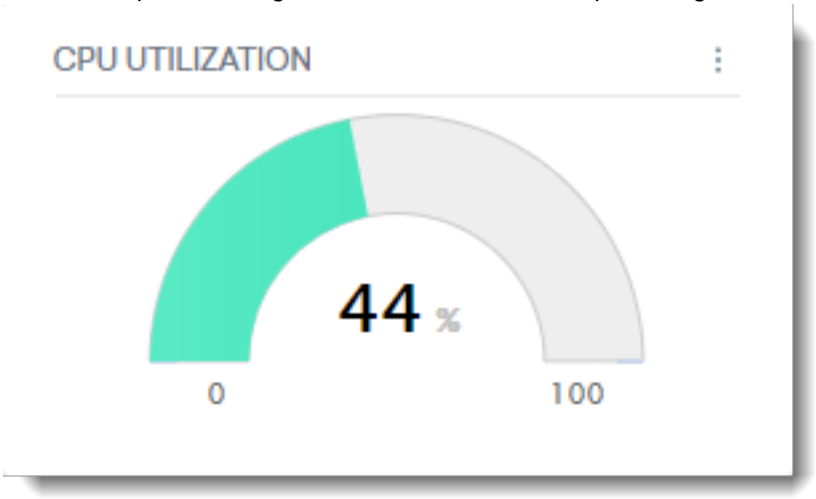
The **Gauge** widget is useful when:

- you want to display the current value of a single metric in an instance

Examples

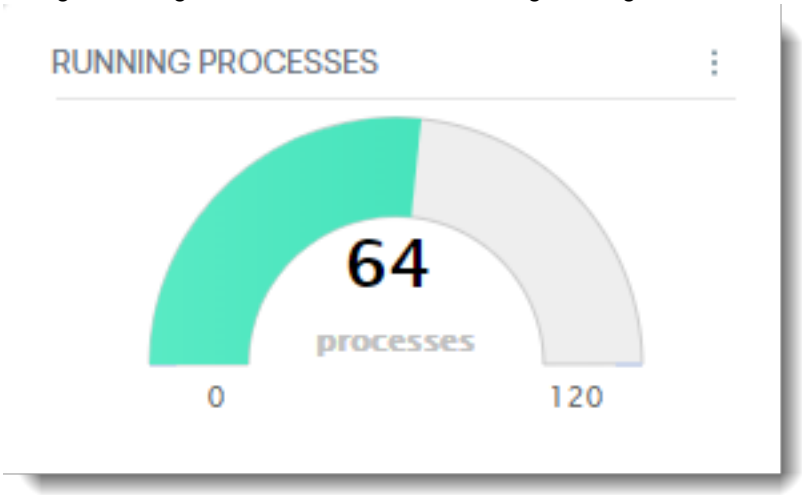
CPU utilization (percentage)

In this example, the Gauge shows the CPU utilization percentage of an instance.



Number of processes running

In this example, the Gauge shows the number of processes running on an instance. The threshold (120) can be configured using the **Max Value** field on the widget configuration drawer.



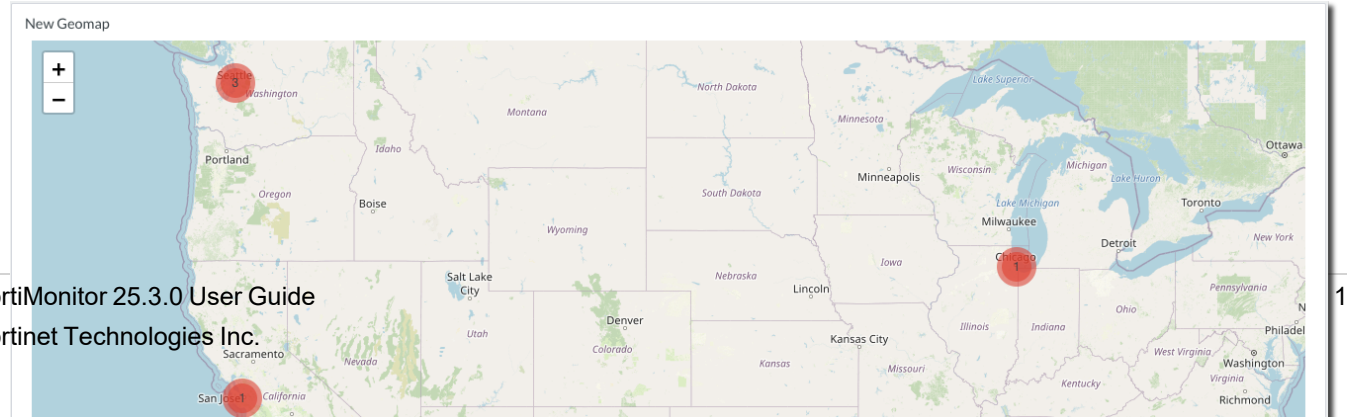
Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Metric	The selected metric's value is displayed on the widget.
Max Value	Use this field to set an upper value or threshold for metrics not based on percentages
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Geomap

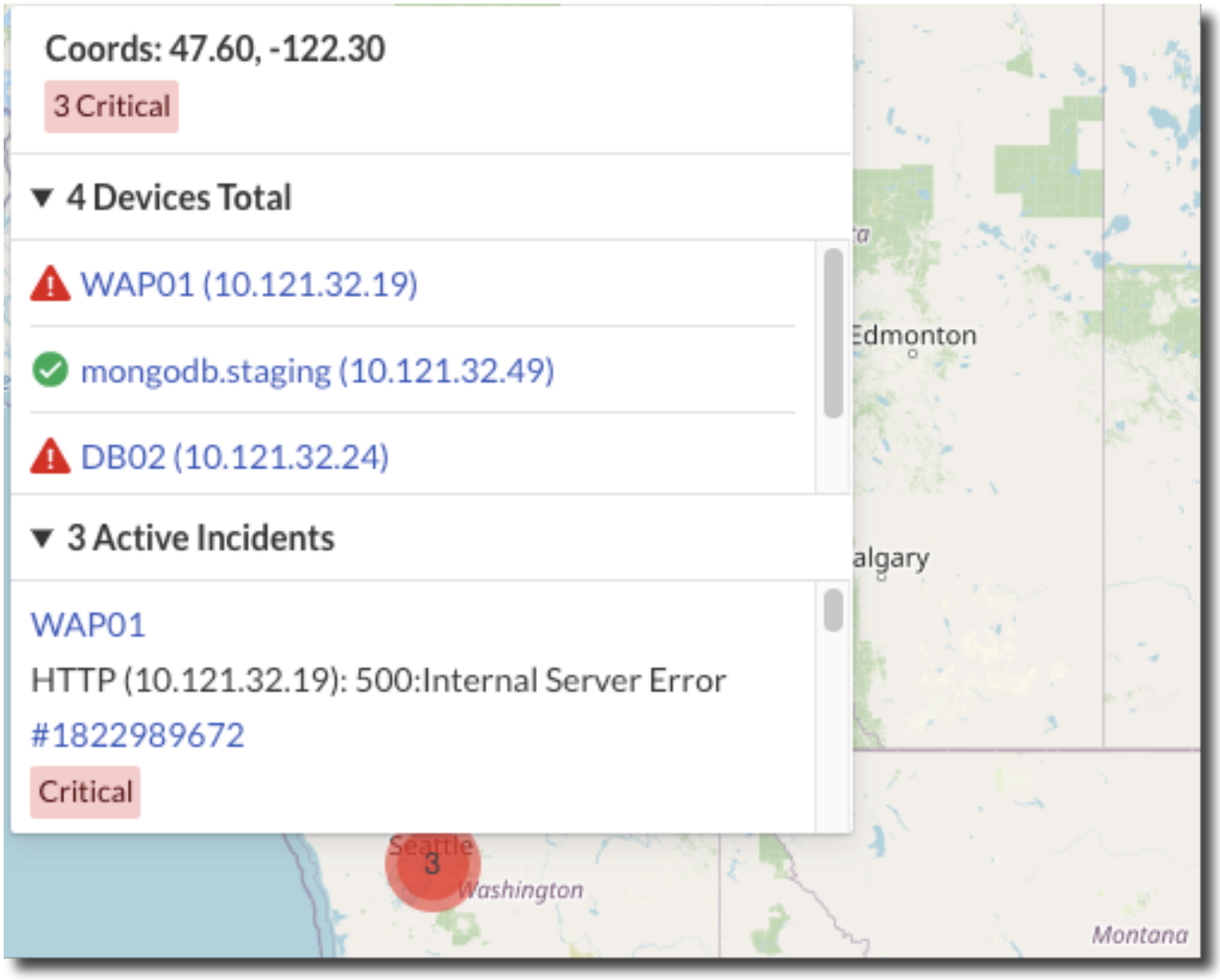
Use the **Geomap** widget to display a rollup of incident counts by geographic location. Similar to other widgets, the Geomap widget can be scoped by instance (tag, group, or name) and by metric type. Geographic location data is entered on an instance's **Details** tab in the **Instance Location** field.



Each geographic location is represented by a color-coded marker that shows the number of incident for that particular region. The color of the marker is based on the severity of the incidents (Warning or Critical).

Clicking on the marker opens a panel that displays the following information:

- Coordinates of the location
- The devices on the location and their individual status
- The number of active incidents
- Incident details



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Before creating the widget, make sure to set the **Instance Location** for each of the instance that you want to display on the Geomap. For more information, see Instance Details page

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Instances	Filter instances by the following: **Values**: • No filter • Tags • Group • Name
Metrics	The selected metric's status is displayed on the widget. Use the search field to search for specific metrics.
Severity Levels	Select which severity level will be included in the map. For example, selecting **Critical** displays the number of all critical incidents. **Values**: • Critical • Warning • All
Additional Metric Filters	You can also use the filters to fine-tune your search results. **Values**: • By tags • By option string

The **HTML Block** widget enables you to create custom widgets using HTML, CSS, and JavaScript.

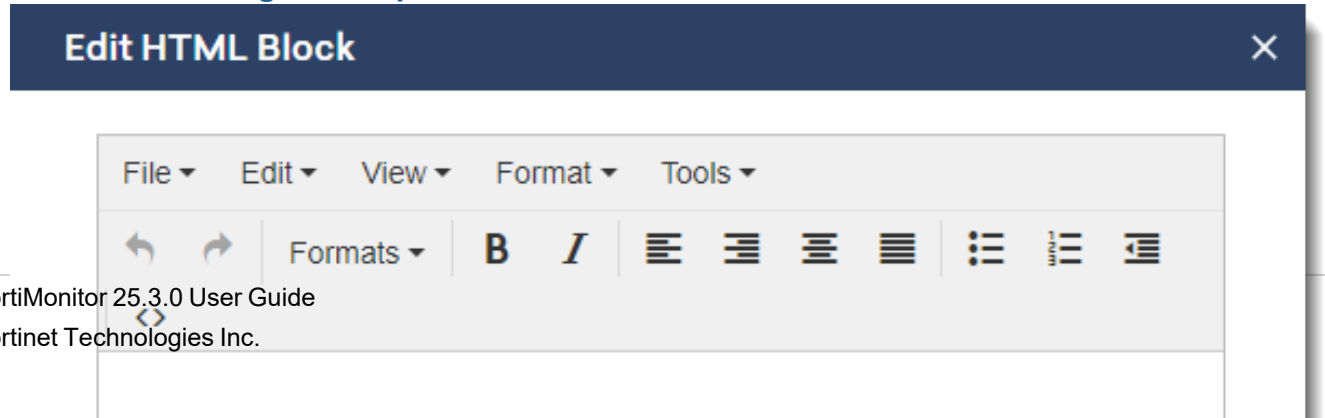
Use case

The **HTML Block** widget is useful when:

- You want to present dynamic information from an external source (RSS feed or API)
- You want to display custom headers and images

Example

HTML Block widget example



Infrastructure Counter

The **Infrastructure Counter** widget allows you to display the number of instances in your FortiMonitor account based on how you configured the widget. You can configure which instances are included in the counter by selecting a combination of one or more instance types and filters. See the following example for more information.

Example

In this example, the **Infrastructure Counter** widget is configured to show the number of Linux-based web servers that have the FortiMonitor Agent installed. This can be accomplished by selecting **Agent instances** in the **Type selection** dropdown field and adding the **linux** and **web** tags. The following section describes the available configuration options in detail.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard.
Type selection	Select one or more instance types.

Instances	Filter instances by:
	• No filter
	• Tags
	  • **Match all tags** – Instances that match ALL of the tags entered will be returned.
	  • **Match any tag** – Instances that match at least one of the tags entered will be returned.
	• Group
	• Name

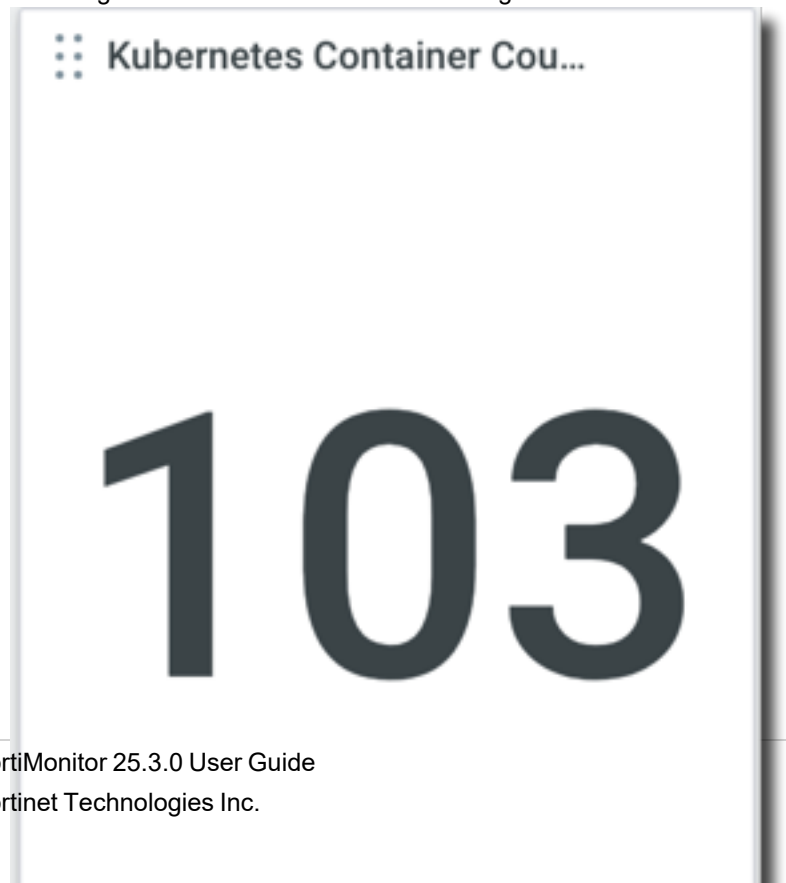
Background Color	Select a background color.
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Kubernetes Container Counter

The **Kubernetes Container Counter** widget displays the number of Kubernetes containers in your infrastructure, based on how the widget is configured. You can filter which containers are included based on its container state, termination reason, or wait reason.

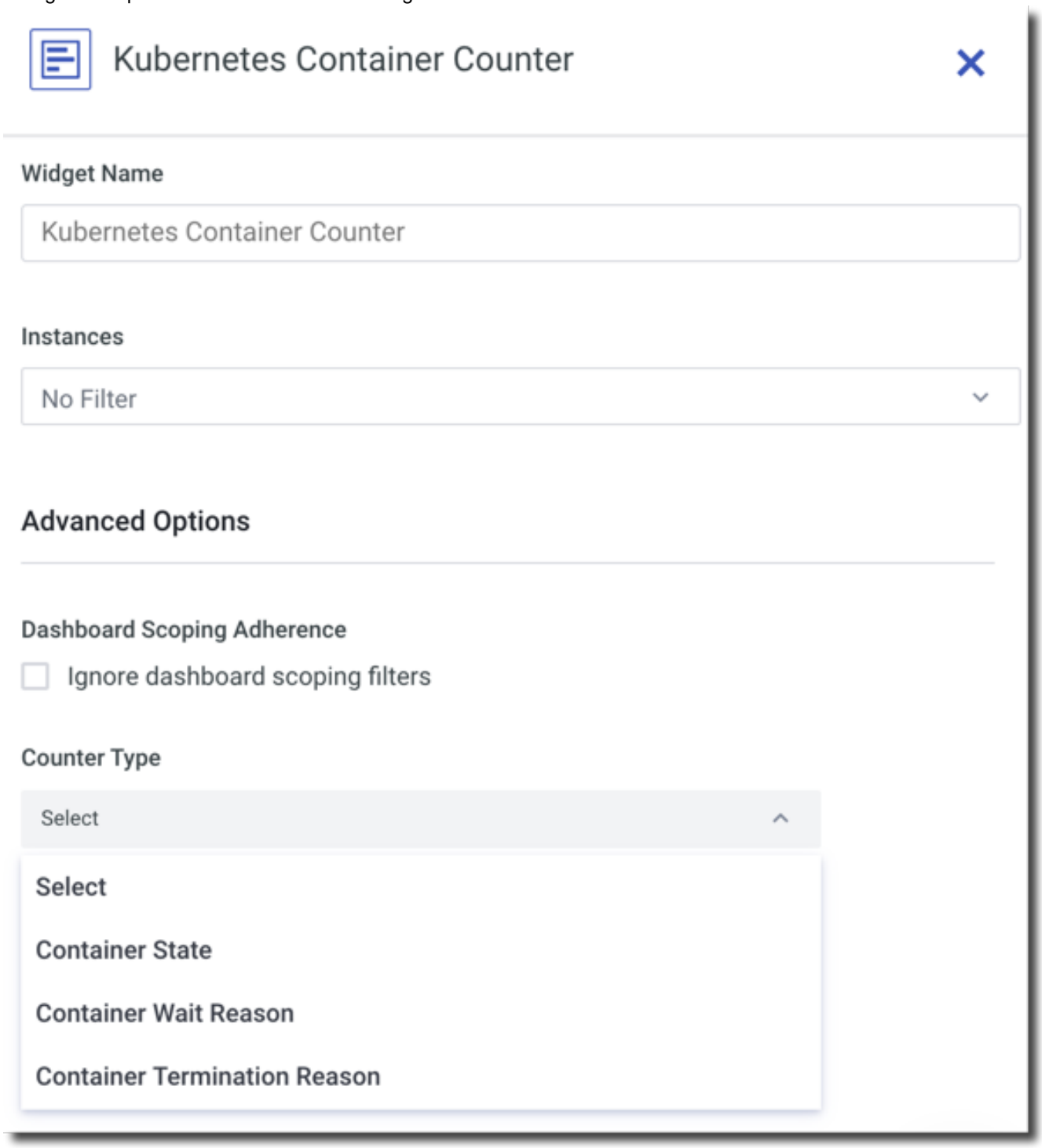
Example

The widget shows the total number of running Kubernetes containers in your infrastructure.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.



Field	Description

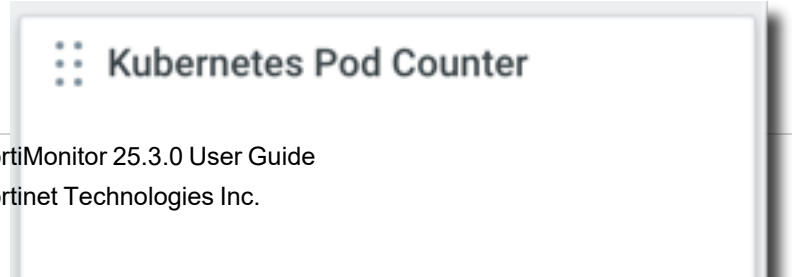
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Instances	Filter instances by the following: • No filter • Tags • Group • Name
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.
Counter Type	
Container State	The state of each container inside a Pod. Containers with the selected **State** will be included in the count displayed on the widget. • Waiting • Running • Terminated
Container Wait Reason	The reason for a container's wait status. Containers with the selected **Container Wait Reason** will be included in the count displayed on the widget. • Container Creating • Crash Loop Back Off • Image Pull Error • Image Pull Back Off • Create Container Config Error • Invalid Image Name • Create Container Error
Container Termination Reason	The reason for a container's termination status. Containers with the selected **Container Termination Reason** will be included in the count displayed on the widget. • OOM Killed • Error • Completed • Container Cannot Run

Kubernetes Pod Counter

The **Kubernetes Pod Counter** widget displays the number Kubernetes pods in your infrastructure, based on how the widget is configured. You can filter which pods are included based its pod phase.

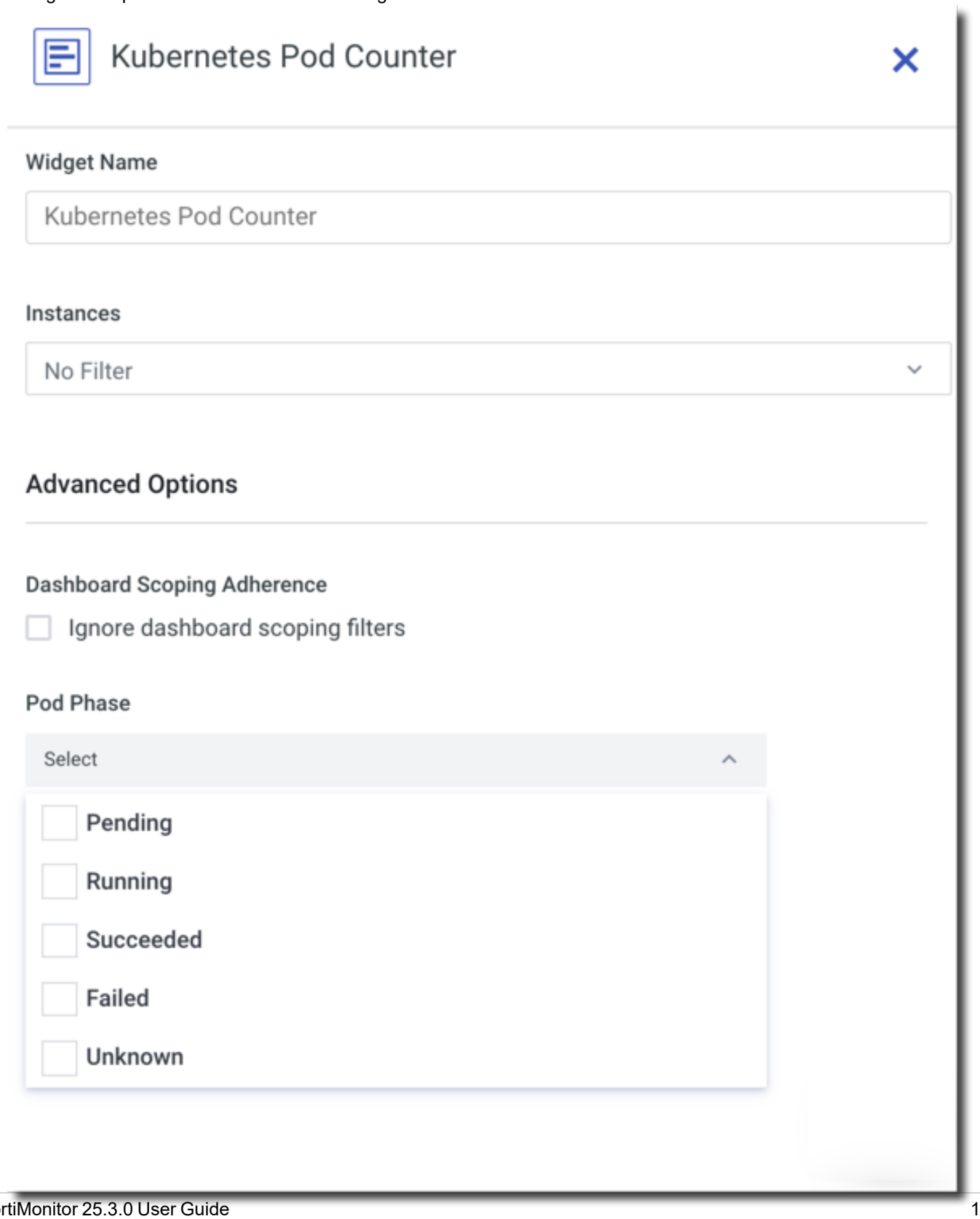
Example

The widget shows the total number of running Kubernetes pods in your infrastructure.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.



Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Instances	Filter instances by the following: • No filter • Tags • Group • Name
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.
Pod Phase	Choose which pods to include in the count depending on where the pod is in its lifecycle (phase). • Pending • Running • Succeeded • Failed • Unknown

Line Graph

Use the **Line Graph** widget to visually represent a time series-based trend of metrics. Metrics can be for a single instance or spread across multiple instances. Like most widgets, you can scope the results to specific instances and metrics using a variety of filtering options.

Select the time range for the trendline and optional lines for comparison to the same time yesterday, last week, or last month. Also, you have options for controlling the X-axis legend formatting and other graph features, like crosshairs and graph points.

Use case

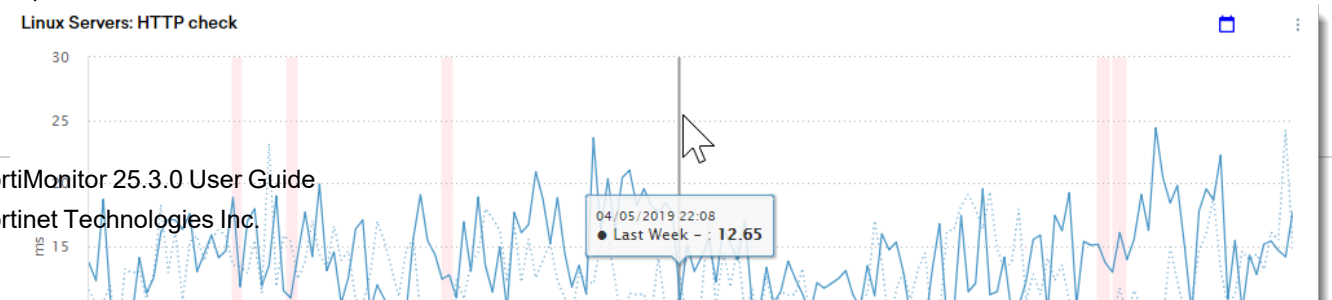
The **Line Graph** widget is useful when:

- Identifying performance trends
- Comparing period over period performance

Examples

HTTP response time comparison

In this example, the line graph displays the aggregated HTTP response time values of all Linux servers. The dotted line represents the historical value of the selected metric.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Metrics	Add the metrics that you want to be visualized in the line graph.
Additional Metric Filters	You can also use the filters to fine-tune your search results. **Values**: • By tags • By option string
Instances	Filter instances by the following: **Values**: • No filter • Tags • Group • Name
Time Range	The time range for the trendline. **Values**: • Last Hour (Default) • Last 15 minutes • Last 6 hours • Last week • Last month • Last year
Formatting	Select which detail you want to hide from the chart. **Values**: • Hide Server Name • Show Units • Hide Server FQDN • Hide Metric Type • Hide Option Name • Float Legend Right
Graph Options	Additional options for the line graph. **Values**: • Show Data Point Labels • Show Crosshairs
Metric Calculation	Select if you want to display the metric data aggregated over the selected time range. Aggregated data can be calculated based on the following values:

	Values: • Average (default) • Min • Max • Sum **Note**: The **Metric Calculation** option only appears if the **Aggregate metric** checkbox is selected.
Time Comparison	Compare the trendline to its historical values. Historical values are represented by a dotted line. **Values**: • None • Same time yesterday • Same time last week • Four weeks prior
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Metric Heatmap

Use the **Metric Heatmap** for quick visual indicators on the current health of metrics. Each cell represents a specific metric and cells will remain green if the current status is healthy, otherwise red or orange depending on the severity of an active incident for that metric. Hovering over the cell will provide details on that metric and instance. Like most widgets, you can scope the results to specific instances and metrics using a variety of filtering options.

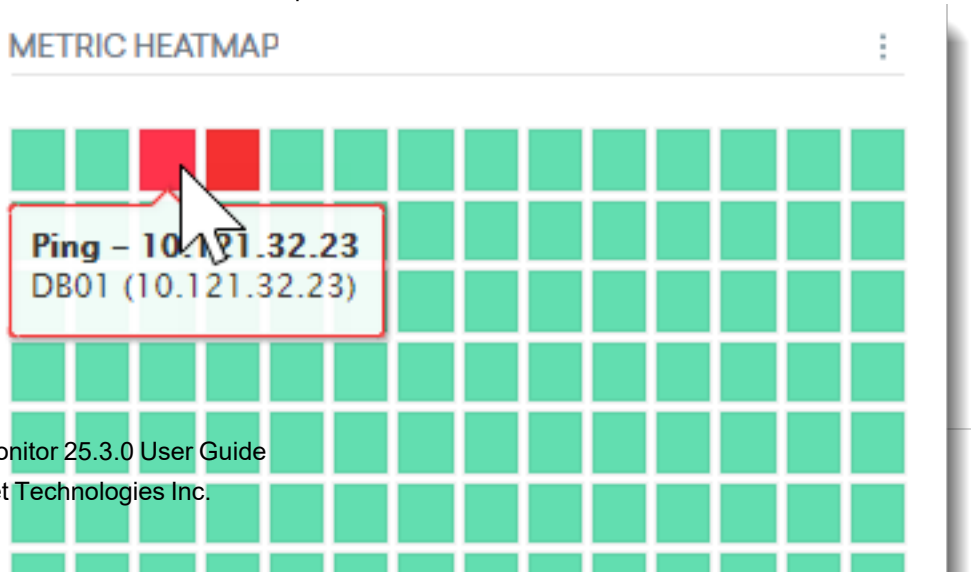
Use case

The **Metric Heatmap** widget is useful when:

- You want to quickly view the current health of the metrics.

Examples

In this example, the widget shows the status of the metrics in an instance group. Healthy metrics are represented by green blocks. The red block represents an incident on the instance.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Metrics	The selected metric's status is displayed on the widget. Use the search field to search for specific metrics.
Filter Metrics	You can also use the filters to fine-tune your search results. **Values**: • By tags • By option string
Instances	Filter instances by the following: **Values**: • No filter • Tags • Group • Name The health of the selected metrics will be visualized in the widget.
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Metric Rollup

The **Metric Rollup** widget is used for showing the current status and historical availability of multiple metrics across different instances in one consolidated row. You can scope the results to specific instances and metrics using tagging. Also, optionally enable/disable several columns that show availability statistics in aggregate.

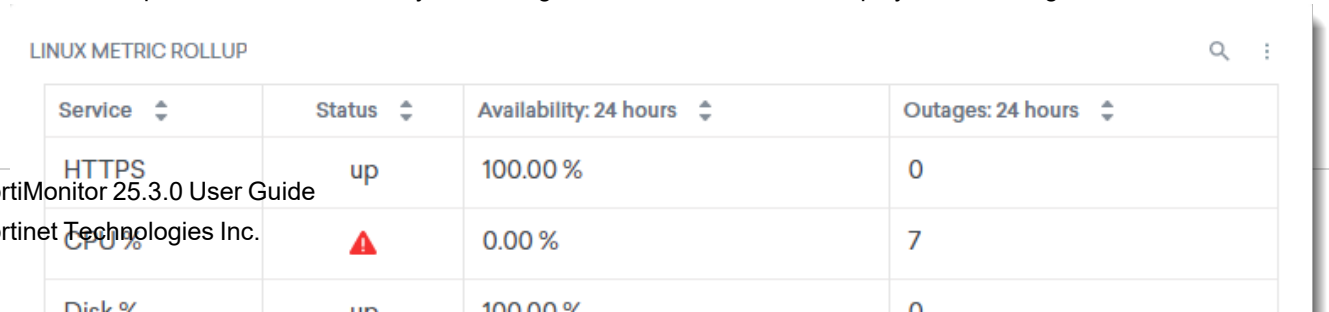
Use case

The **Metric Rollup** widget is useful when:

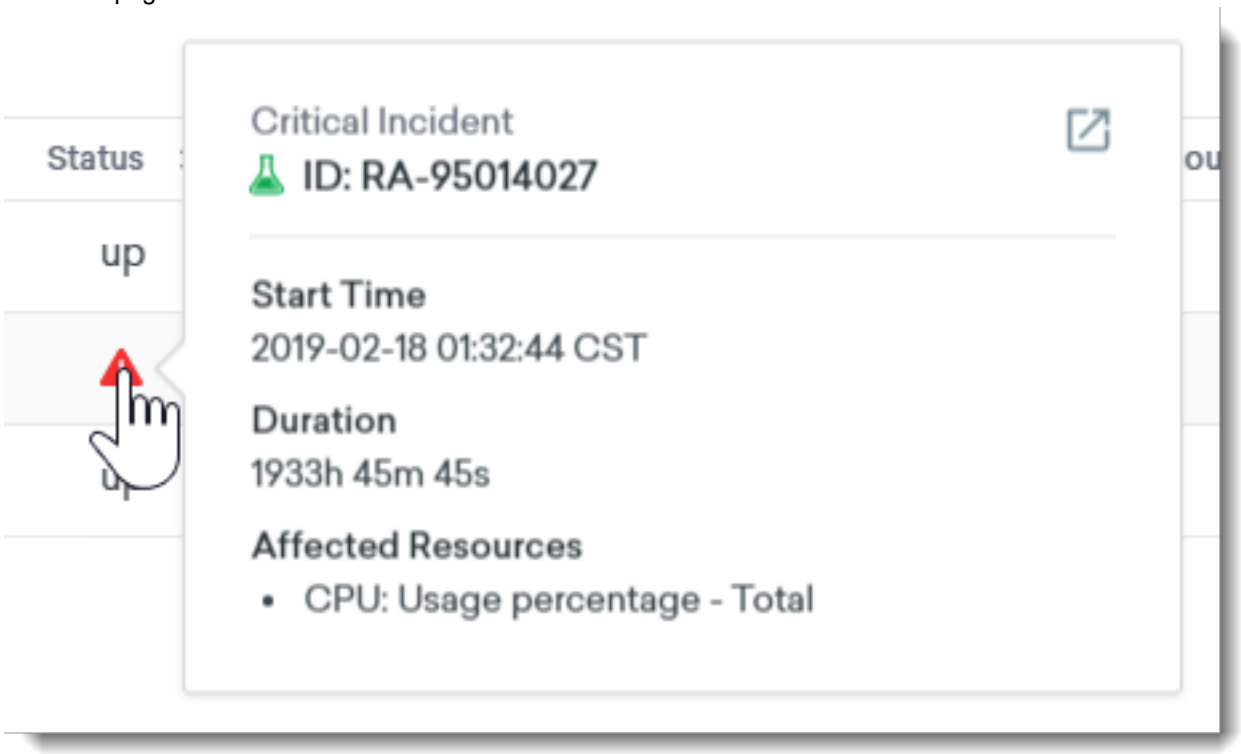
- You want to display the current status and historical availability of several metrics.

Examples

In this example, the status, availability, and outages of several metrics are displayed in the widget.



You can also hover your mouse to an incident to show the details. Selecting the **Incident ID** will redirect you to the incident's page.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Service name	Name of the service/metric. Select the pen icon to rename the service/metric.
Metric Types	Add the metrics that you want to display. You can search by metric name or by category.
Metric Tag Filter	Add specific metric using metric tags.
Instance Tag Filter	Add specific metrics using an instance's tag.
Add Service	Select the **Add Service** button to add another row to the widget.
Data Points	Select which data points you want to show or hide. **Values**: • Service

	• Status • Availability: 24 hours • Availability: 7 days • Outages: 24 hours • Outages: 7 days
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Metric Summary

The **Metric Summary** widget is used for showing current and aggregate metric values in a tabular form. Like most widgets, you can scope the results to specific instances and metrics using a variety of filtering options. Also, pick from several configurable columns that you can add to the summary table to display availability/SLA, min, max, avg values. Tune the specific order of the columns as well as the sorting of rows.

Use case

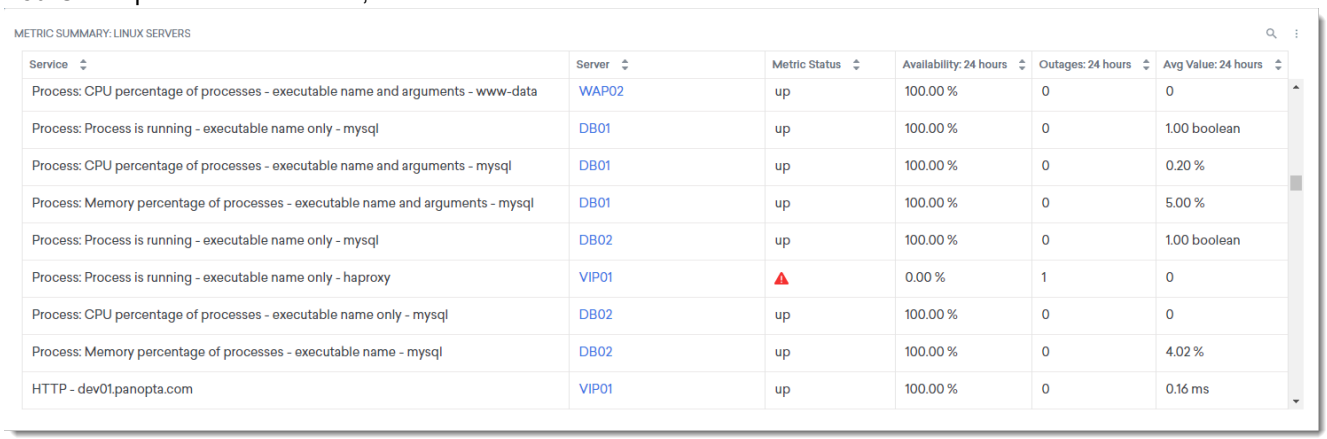
The **Metric Summary** widget is useful when:

- You want to display the current status of the metrics in an instance or a group of instances.
- You want to display the historical availability, averages, and outage information of the selected metrics.

Example

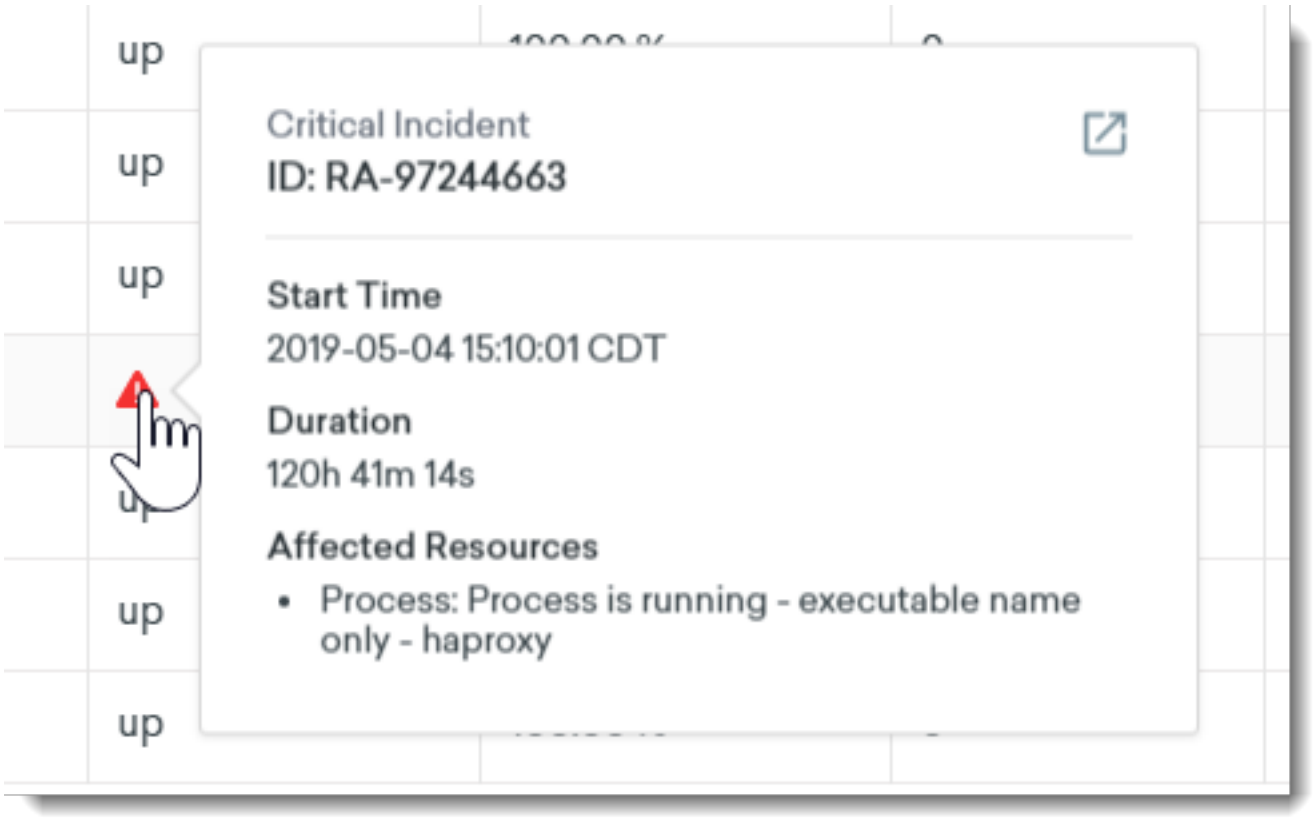
Linux servers status and availability

In this example, the widget displays the status and availability (24 hours) of all Linux servers. The **Outages: 24 hours** data point is also enabled, which shows the number of incidents that occurred on the Linux instance.


METRIC SUMMARY: LINUX SERVERS

Service	Server	Metric Status	Availability: 24 hours	Outages: 24 hours	Avg Value: 24 hours
Process: CPU percentage of processes - executable name and arguments - www-data	WAP02	up	100.00 %	0	0
Process: Process is running - executable name only - mysql	DB01	up	100.00 %	0	1.00 boolean
Process: CPU percentage of processes - executable name and arguments - mysql	DB01	up	100.00 %	0	0.20 %
Process: Memory percentage of processes - executable name and arguments - mysql	DB01	up	100.00 %	0	5.00 %
Process: Process is running - executable name only - mysql	DB02	up	100.00 %	0	1.00 boolean
Process: Process is running - executable name only - haproxy	VIP01	⚠	0.00 %	1	0
Process: CPU percentage of processes - executable name only - mysql	DB02	up	100.00 %	0	0
Process: Memory percentage of processes - executable name - mysql	DB02	up	100.00 %	0	4.02 %
HTTP - dev01.panopta.com	VIP01	up	100.00 %	0	0.16 ms

You can also hover your mouse to an incident to show the details. Selecting the **Incident ID** will redirect you to the incident's page.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Metrics	Availability, status, and outage information will be derived from the selected metrics.
Additional Metric Filters	You can also use the filters to fine-tune your search results. **Values**: • By tags • By option string
Instances	Filter instances by the following: **Values**: • No filter • Tags • Group • Name

Data Points	Select which data points you want to show or hide. **Values**: • Service • Server • Metric Status • Metric Value • Availability: 24 hours • Availability: 7 days • Outages: 24 hours • Outages: 7 days • Avg Value: 24 hours • Avg value: 7 days • Min value: 24 hours • Min value: 7 days • Max value: 24 hours • Max value: 7 days
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Network device widgets

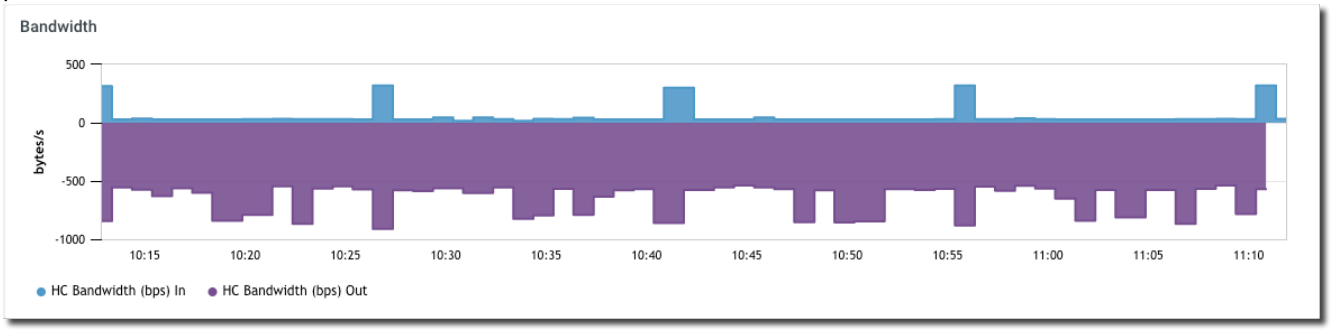
Available network device widgets are described in the following list:

- **Bandwidth In/Out** – This widget shows the inbound and outbound network traffic within a specified time period.
- **Network Errors In/Out** – This widget shows the inbound and outbound network error traffic within a specified time period.
- **Packets In/Out** – This widget shows the inbound and outbound network error traffic within a specified time period.

Examples

Bandwidth in and out

In this example, the widget shows the hourly inbound and outbound network traffic (in bits per second) of the selected port.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

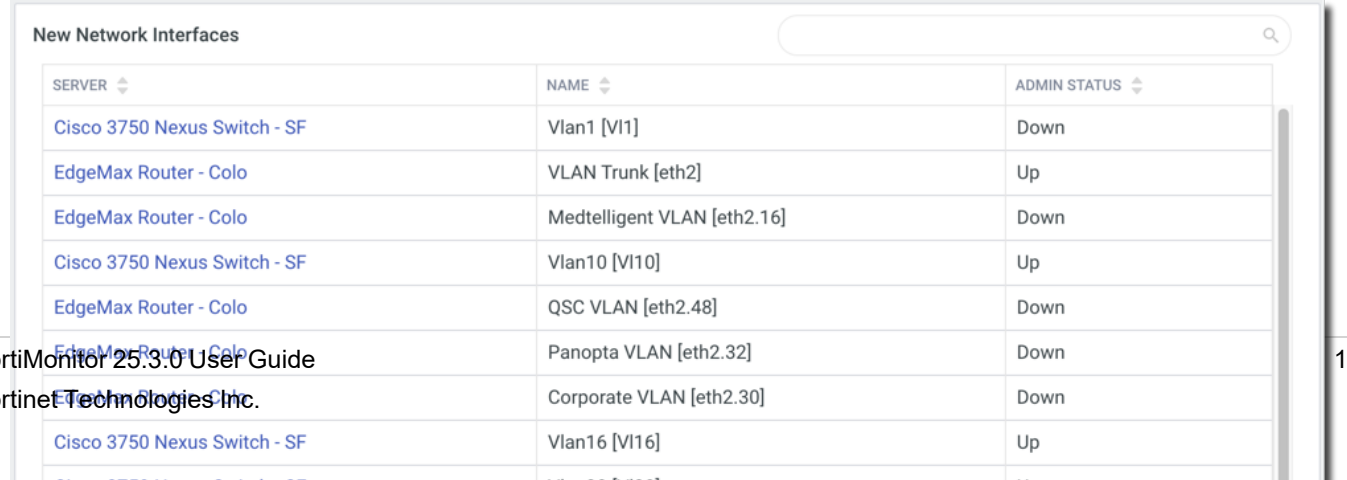
Field	Description
Block Title	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Instance	Choose a network device. Use the **Search** field to filter the list of instances.
Port	Choose the port that you want to monitor.
Time Range	In the **Options** tab, select a time range for the widget. **Values**: • Last Hour (Default) • Last 15 minutes • Last 6 hours • Last 12 hours • Last 24 hours • Last week • Last month • Last year
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Network Interfaces

The **Network Interfaces** widget provides a table displaying the status and bandwidth utilization of network device interfaces. The widget can also be configured to show or hide other data points as shown in the table below.

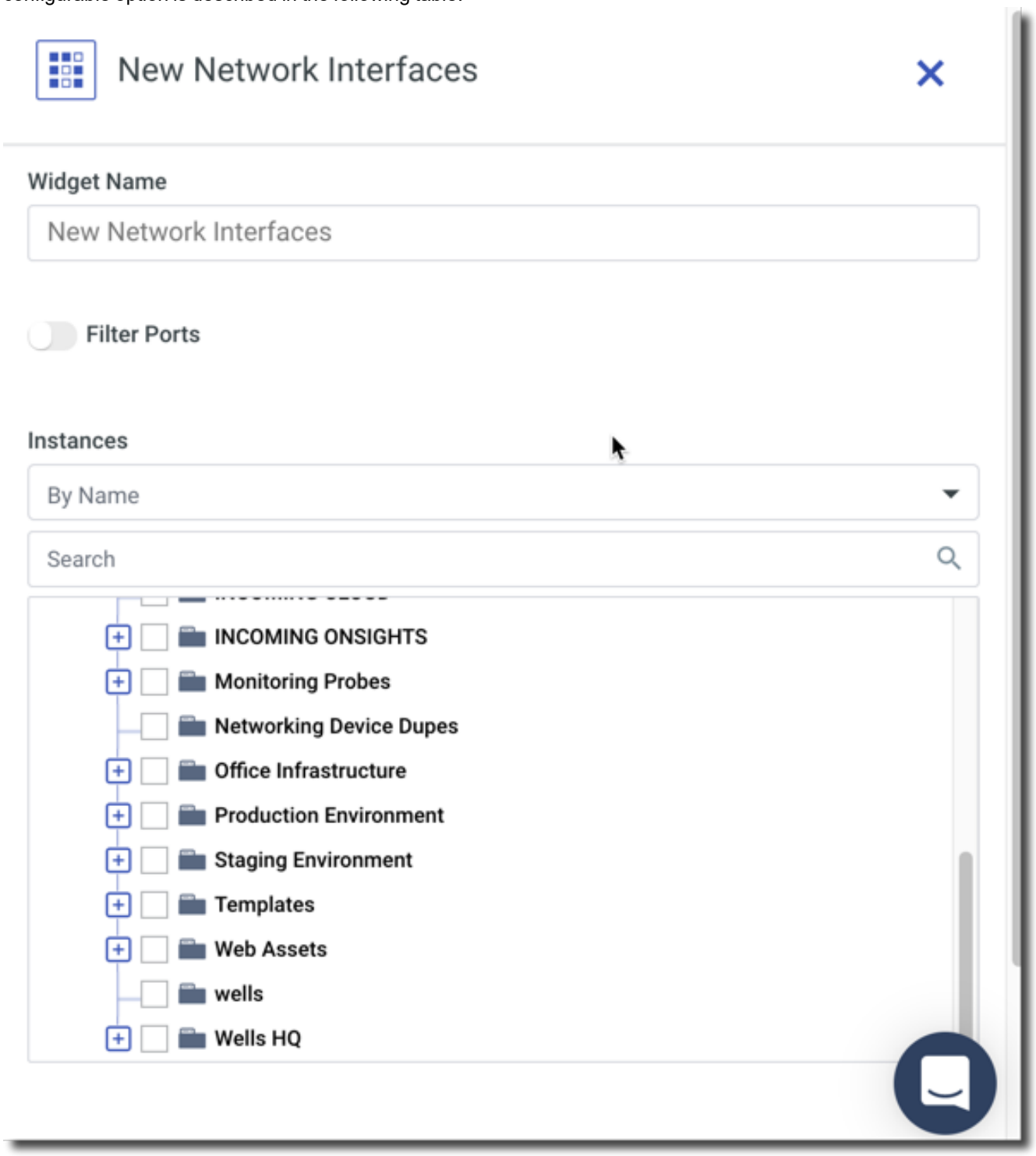
Example

In this example, the **Network Interfaces** widget is configured to show the status of all of the network interfaces in the **Core Networking** group. By default, only the **Server**, **Name**, and **Admin Status** data points are shown. There are a few more data points that you can toggle on or off while setting up the widget. This and other configuration options are discussed in the following section.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.



Field	Description
Widget name	Name of the widget as it appears on the dashboard.
Filter Ports	Enable this option if you want to filter ports by name.

Instances	Filter instances by:
	• No filter
	• Tags
	  • **Match all tags** – Instances that match ALL of the tags entered will be returned.
	  • **Match any tag** – Instances that match at least one of the tags entered will be returned.
	• Group
	• Name

Data Points	The following data points are available. You can toggle each data point on or off by clicking the **eye** icon. Click the **up** or **down** arrow icons if you want to reorder them. • Server (enabled by default) • Name (enabled by default) • Admin Status (enabled by default) • Operational Status • Discards In • Discards Out • Errors In • Errors Out • Bandwidth In • Bandwidth Out • Bandwidth In Util • Bandwidth Out Util • NetFlow
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Resolved Incident Counter

The **Resolved Incident Counter** widget will display the count of recently resolved incidents in your account, based on the configuration of the widget. Like most widgets, you can scope the results to specific instances and metrics.

Use case

The **Resolved Incident Counter** widget is useful when:

- Monitoring the total number of recently resolved incidents for a specified time range in your infrastructure
- You want to see the number of resolved incidents by metric or by severity for a specified time range

Examples

All resolved incidents

The widget shows the total number of recently resolved incidents in your infrastructure.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Severity levels	Select which severity level will be included in the calculated number of resolved incidents. For example, selecting Critical displays the number of all resolved critical incidents. **Values**: • Critical • Warning • All
Metrics	Add the metrics that you want to include in the calculation of resolved incidents.
Filter metrics	You can also use the filters to fine-tune your search results. **Values**: • No filter • By tags • By option string

Instance	Filter instances by the following: • No filter • Tags • Group • Name Resolved incidents in the selected instances will be included in the calculation of resolved incidents.
Time Range	Shows the number of resolved incidents based on the selected time range. **Values**: • Last hour (default) • Last 15 minutes • Last 6 hours • Last 12 hours • Last 24 hours • Last week • Last month • Last year
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Resolved Incident Details

The **Resolved Incident Details** widget provides the details of each recently resolved incident included in your filter. The configurable columns for different data points allow you to highlight the data that you require. The ordering of the columns can be set along with which column is used for sorting (the carrot). Like most widgets, you can scope the results to specific instances and metrics.

Use case

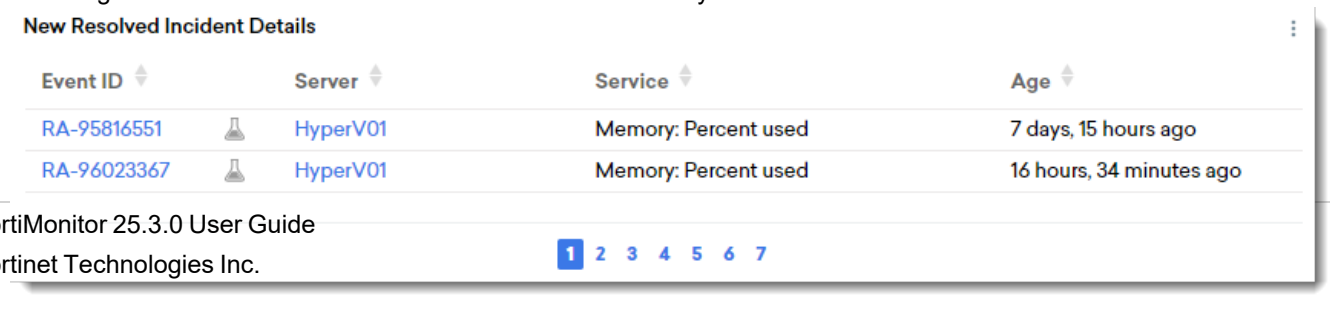
The **Resolved Incident Details** widget is useful when:

- You want to see details about the resolved incidents in your infrastructure
- You want to see details about the resolved incidents on specific metrics, instances, or based on the incident's severity

Examples

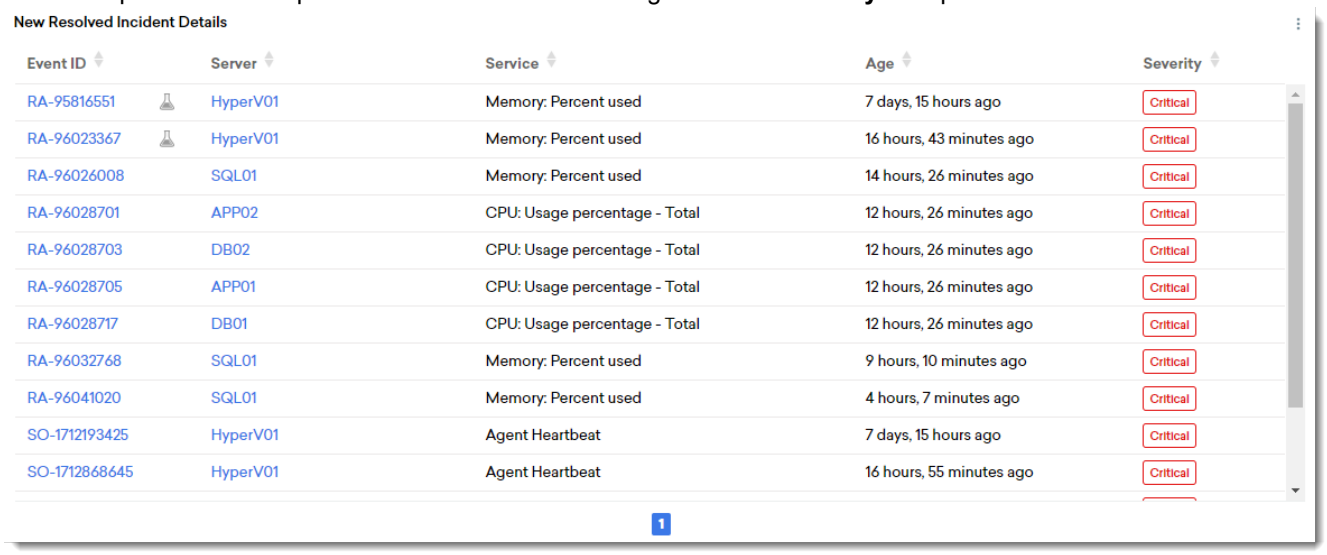
All resolved incidents

The widget shows a detailed list of all the resolved incidents in your infrastructure.



All resolved incidents (expanded and with the Severity data point)

This example shows an expanded **Resolved Incident** widget with the **Severity** data point activated.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Instances	Filter instances by the following: • No filter • Tags • Group • Name The details of the instances selected here will be displayed on the widget.
Metrics	Add the metrics that you want to show up on the widget.
Additional Metric Filters	You can also use the filters to fine-tune your search results. **Values**: • No filter (default) • By tags • By option string
Time Range	Shows the details of resolved incidents based on the selected time range. **Values**: • Last 15 minutes

	• Last hour • Last 6 hours • Last 12 hours • Last 24 hours (Default) • Last week • Last month • Last year
Data Points	Select which data points you want to show or hide. **Values**: • CounterMeasures • Ack • Server • FQDN • Service • Age • Start Time • Severity • Assigned To
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Server Heatmap

The **Server Heatmap** widget is used for visualizing the availability of individual servers. Optionally set which metrics to take into account for the availability of the server or leave it blank for all. Cells will remain green if the current status is healthy, otherwise red or orange depending on the severity of an active incident for that server. Hovering over the cell will provide details on that instance. Like most widgets, you can scope the results to specific instances using a variety of filtering options.

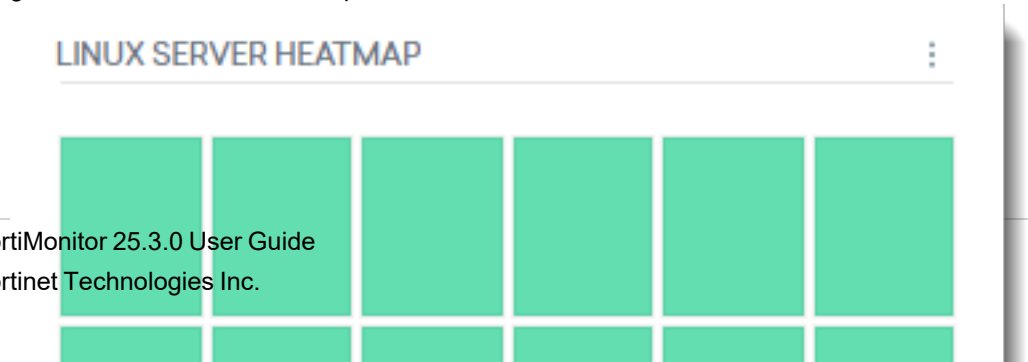
Use case

The **Server Heatmap** widget is useful when:

- You want to quickly view the current health of the instances on an instance group.

Examples

In this example, the widget shows the status of each instance with the **Linux** tag. Healthy instances are represented by green blocks. The red block represents an incident on the instance.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Instances	Filter instances by the following: **Values**: • No filter • Tags • Group • Name The health of the selected instances will be visualized in the widget.
Metrics	The selected metric's status is displayed on the widget. Use the search field to search for specific metrics.
Additional Metric Filters	You can also use the filters to fine-tune your search results. **Values**: • No filter • By tags • By option string
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Instance Scorecard

The **Instance Scorecard** widget is a compact way of seeing the typical resource utilization metrics for a server row by row for a glance on server health. Like most widgets, you can scope the results to specific infrastructure using several filters.

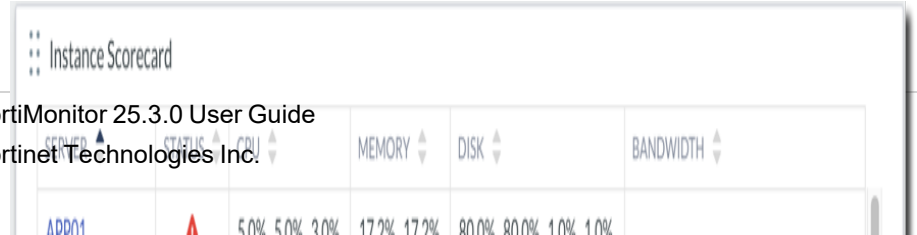
Use case

The **Instance Scorecard** widget is useful when:

- You want to view the status and resource utilization (CPU, memory, bandwidth in/out, and disk) and overall health of a group of instances.

Examples

In this example, the widget shows the status and resource utilization of each instance. The table is also sorted alphabetically by server name.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Instances	Filter instances by the following: **Values**: • No filter • Tags • Group • Name The status and resource utilization of the selected instances will be visualized in the widget.
Data Points	Configure the sorting of rows by: **Values**: • Server • Status • CPU • Memory • Disk • Bandwidth
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Instance Summary Table

The **Instance Summary Table** widget is used for showing the current status and historical availability for a group of servers. Like most widgets, you can scope the results to specific instances and metrics using a variety of filtering options. Also, nine other columns are available to add, which provide more categorization and availability statistics for each server.

Use case

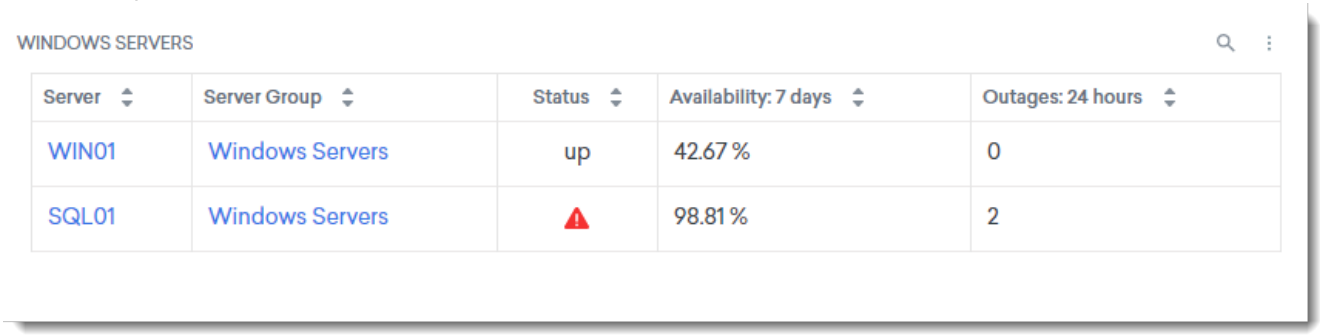
The **Instance Summary Table** widget is useful when:

- You want to display the current status of a group of instances.
- You want to display the historical availability and outage information of a group of instances.

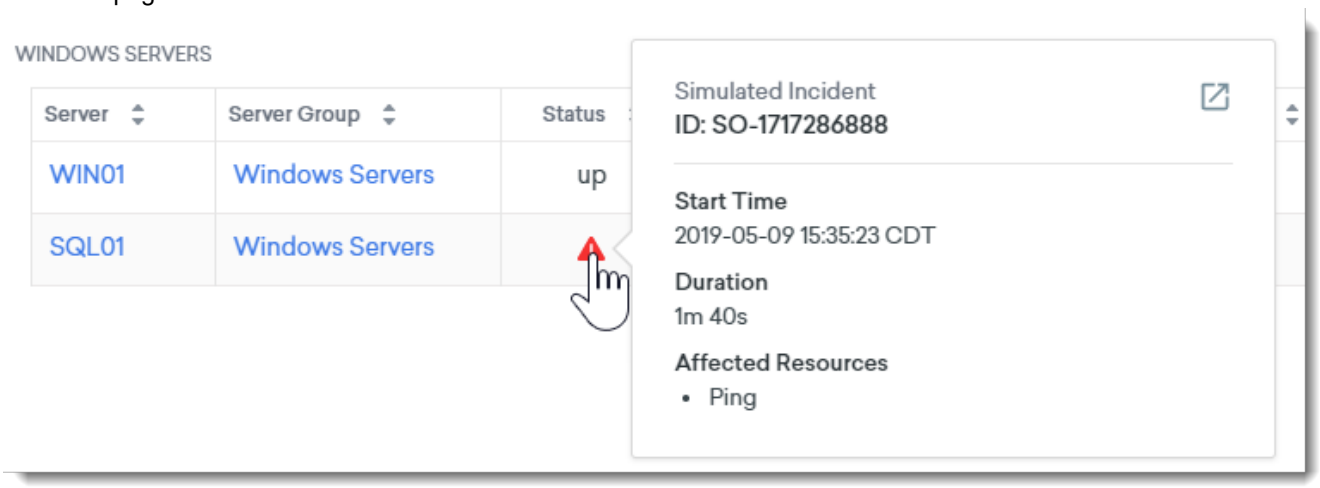
Example

Windows servers status and availability

In this example, the widget displays the status and availability (7 days) of all Windows servers. The **Outages: 24 hours** data point is also enabled, which shows the number of incidents that occurred on the Windows instance.



You can also hover your mouse to an incident to show the details. Selecting the **Incident ID** will redirect you to the incident's page.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Instances	Filter instances by the following: **Values**: • No filter • Tags

	• Group • Name
Metrics	Availability, status, and outage information will be derived from the selected metrics.
Additional Metric Filters	You can also use the filters to fine-tune your search results. **Values**: • No filter • By tags • By option string
Background Color	Select a background color for the widget.
Data Points	Select which data points you want to show or hide. **Values**: • Server Group • Server Device Vendor • Server Device Model • Tags • Status • Availability: 24 hours • Availability: 7 days • Outages: 24 hours • Outages: 7 days
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Single Metric

Use the **Single Metric** widget to highlight a specific metric value. Choose from different aggregate functions, like average, minimum, or maximum throughout a specific time range or choose to show the live value.

Use case

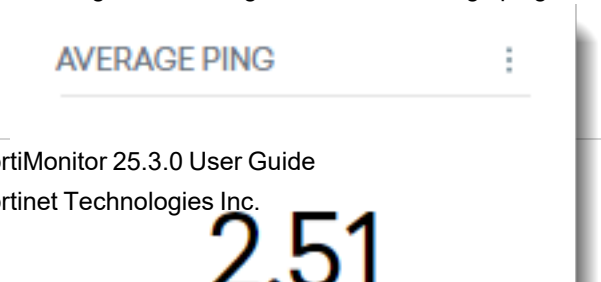
The **Single Metric** widget is useful when:

- you want a live view of the aggregated value of a single metric in an instance
- you want to display the aggregated value of a single metric over a specified time range

Examples

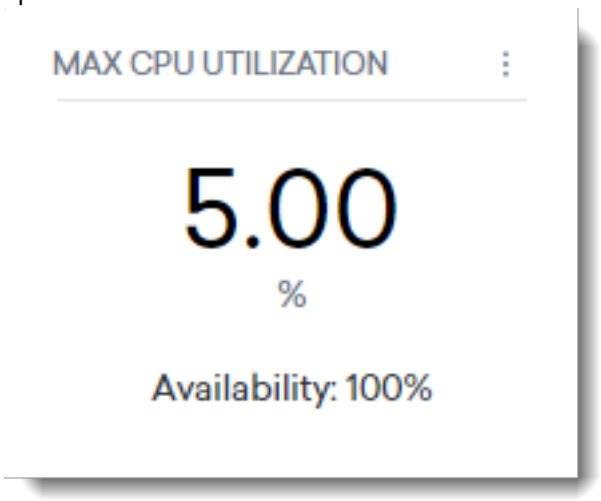
Average ping

The single metric widget shows the average ping of an application server in the last 24 hours.



Aggregated maximum CPU utilization

In this example, the widget shows the maximum CPU utilization of an AWS instance in the last 6 hours. The Availability option is also enabled.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Metrics	The selected metric's value is displayed on the widget. Use the search field to search for specific metrics.
Show availability	Enable this option to display the historical availability of the selected metric.
Aggregation	Aggregated data can be calculated based on the following values: **Values**: • Average (Default) • Min • Max • Sum
Time Range	Shows the live or aggregated metric value of a metric based on the selected time range. **Values**: • Live • Last hour • Last 15 minutes

	• Last 6 hours (default) • Last 12 hours • Last 24 hours • Last week • Last month • Last year
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Stoplight

The **Stoplight** widget is a clear visual indicator of the current availability of a specific service or metric. This is commonly used to display whether a critical service is up or down. The last outage/incident involving this check/metric will be displayed below the metric label.

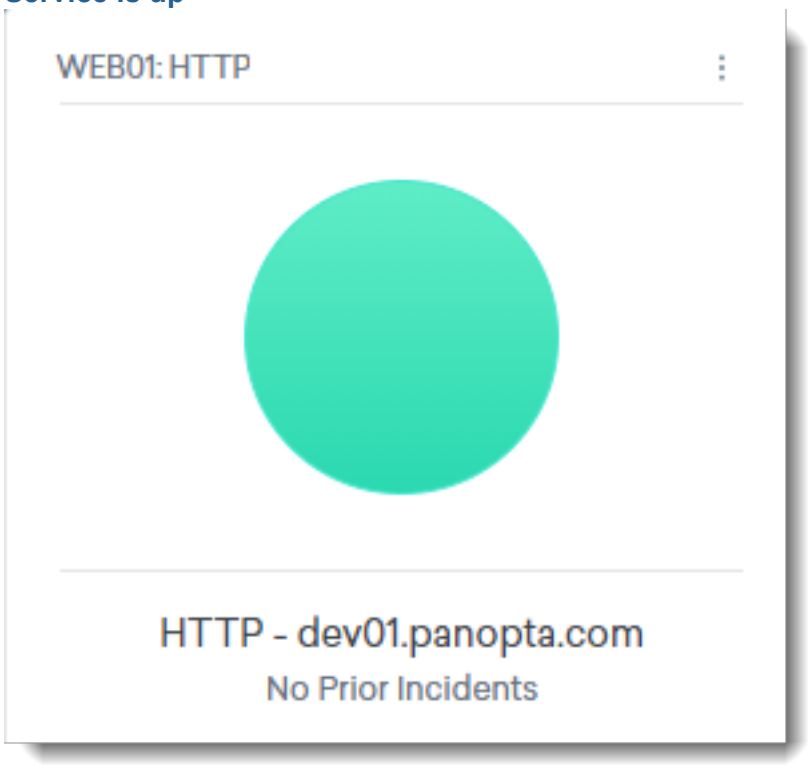
Use case

The **Stoplight** widget is useful when:

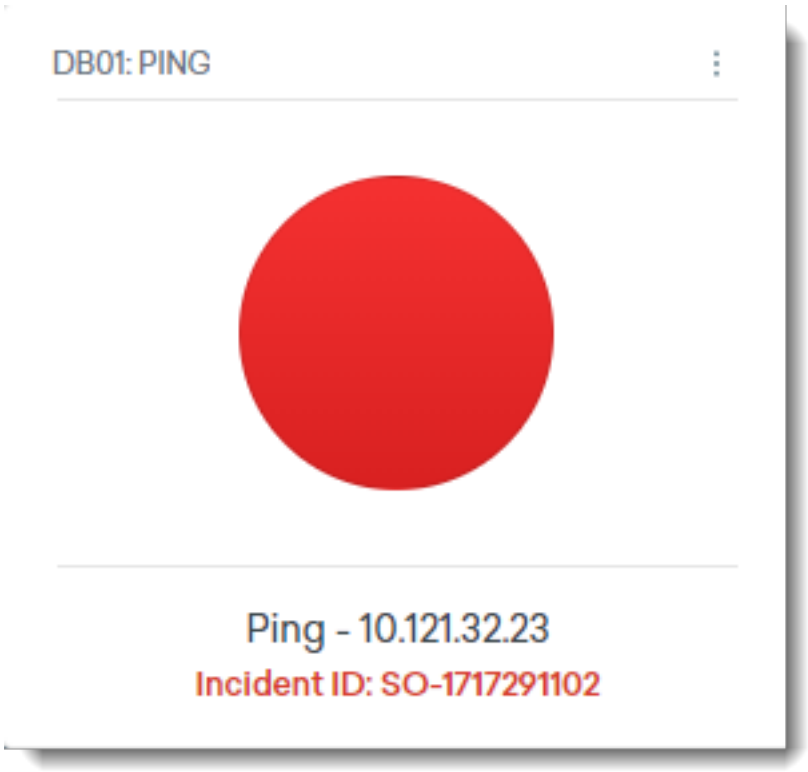
- You want a simple way to display the current status of a critical service or metric.

Examples

Service is up



Service is down



When a service is down, the Incident ID is also displayed on the widget. Selecting the **Incident ID** will redirect you to the incident's page.

Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Services	Select the service or metric that you want to display on the widget.

Up/Down

The **Up/Down** widget is a clear visual indicator of the current availability of a specific service or metric. This is commonly used to display whether a critical service is up or down. The last outage/incident involving this check/metric will be displayed below the metric label.

Use case

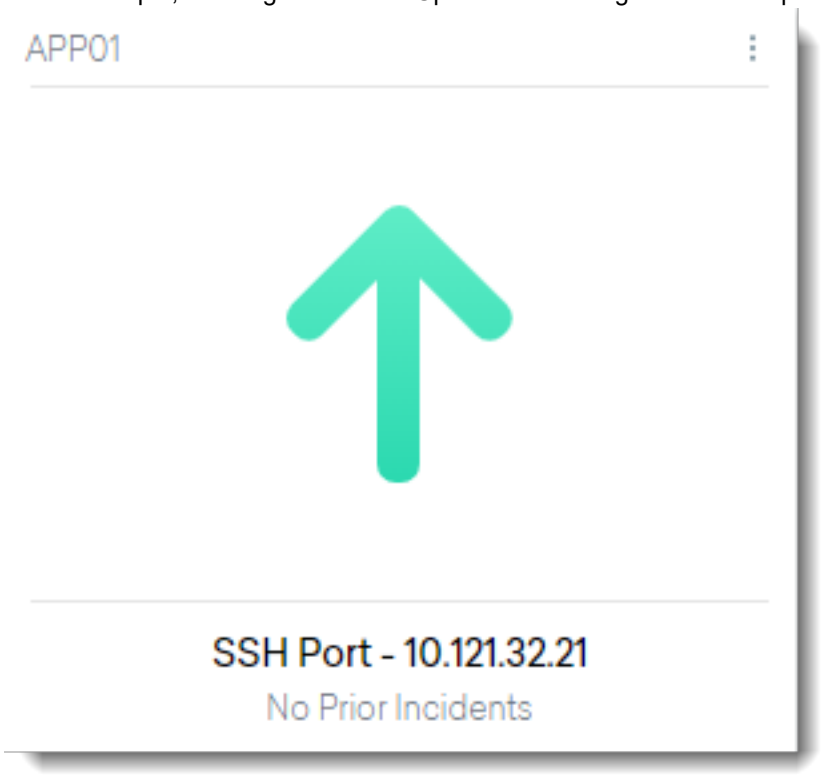
The **Up/Down** widget is useful when:

- You want to display the current status of a critical service or metric.

Examples

Service is up

In this example, the widget shows an Up arrow indicating that the SSH port is reachable.



Service is down

A red down arrow indicates that there is an incident on the service. Selecting the **Incident ID** will open the incident on a new tab.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.

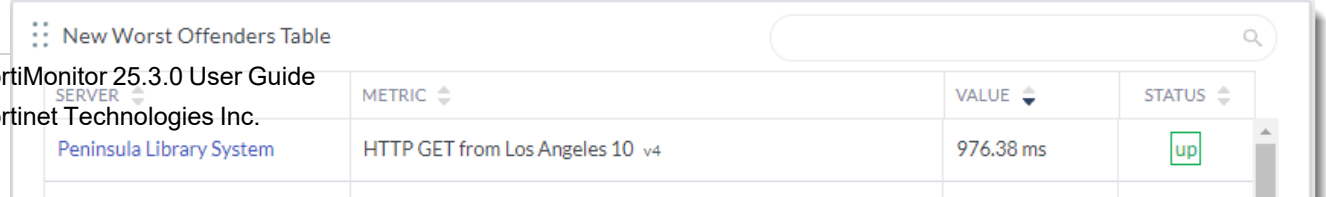
Field	Description
Widget name	Name of the widget as it appears on the dashboard. This field shows the default name of the widget. Select the pen icon to rename the widget.
Services	Select the service or metric that you want to display on the widget.
Dashboard Scoping Adherence	Enabling this option will allow you to bypass Dashboard-level scoping.

Worst Offenders

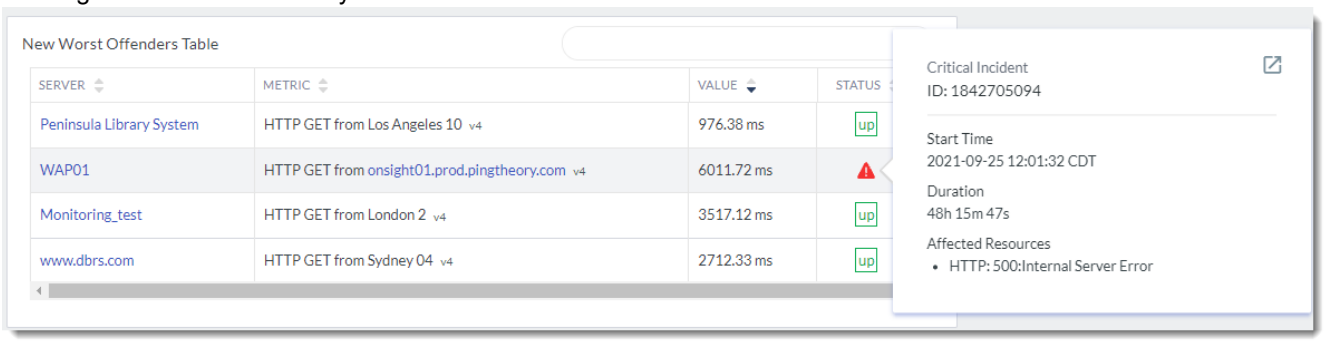
The **Worst Offenders** widget provides a table displaying the worst/top offenders for a given metric. This allows you to view the status of each instance and drill down to its problem details.

The table includes the following information:

- Instance name
- Metric
- Current or aggregated value
- Status of the metric

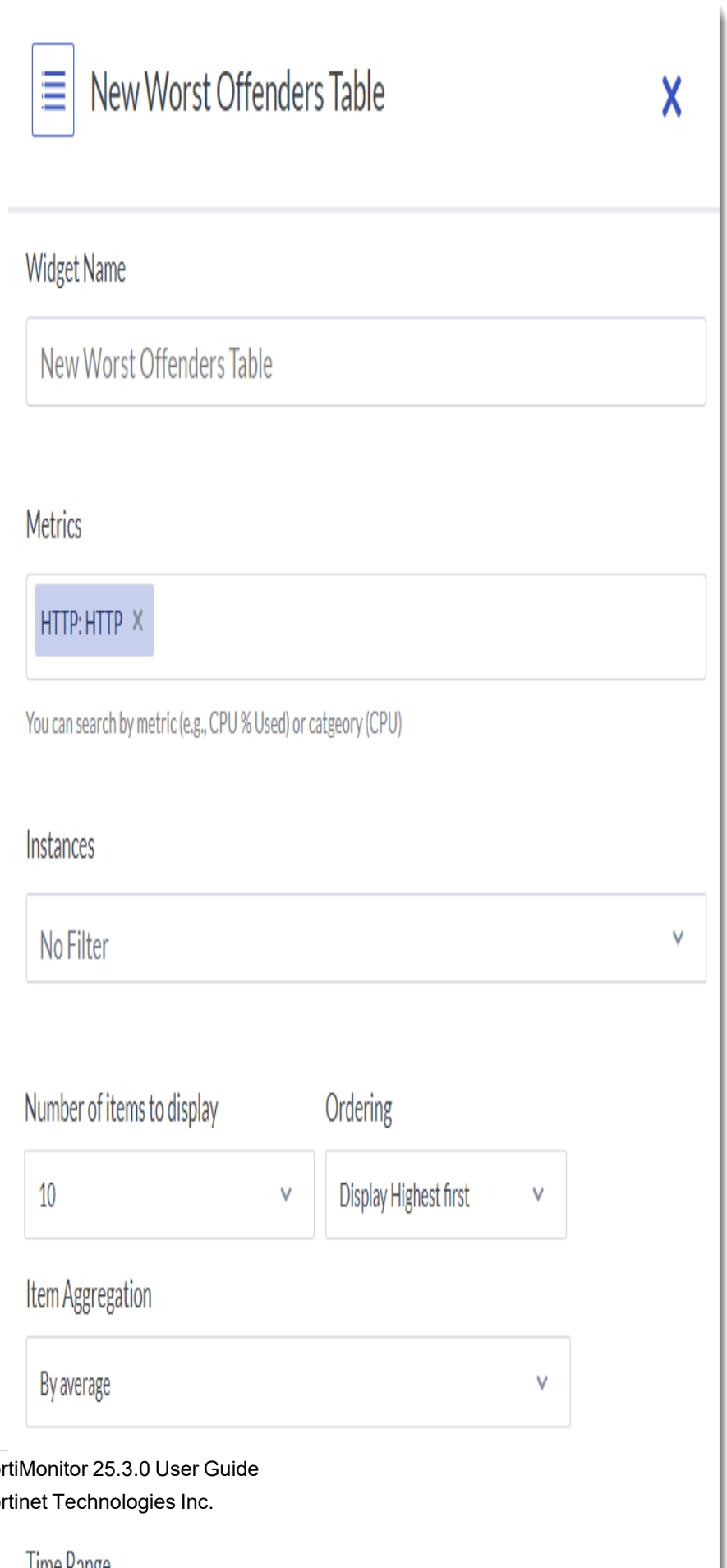


Clicking the status icon allows you to view the details of the incident.



Configuration

When creating or editing a widget, a drawer that shows the widget's configuration options slides out from the right. Each configurable option is described in the following table.


New Worst Offenders Table
Widget Name
New Worst Offenders Table
Metrics
HTTP: HTTP
You can search by metric (e.g., CPU % Used) or catgeory (CPU)
Instances
No Filter
Number of items to display
10
Ordering
Display Highest first
Item Aggregation
By average
Time Range

Field	Description
Widget name	Name of the widget as it appears on the dashboard.
Metrics	Availability, status, and outage information will be derived from the selected metrics. Use the search field to search for specific metrics.

Instances	Filter instances by:
	• No filter
	• Tags
	  • **Match all tags** – Instances that match ALL of the tags entered will be returned.
	  • **Match any tag** – Instances that match at least one of the tags entered will be returned.
	• Group
	• Name

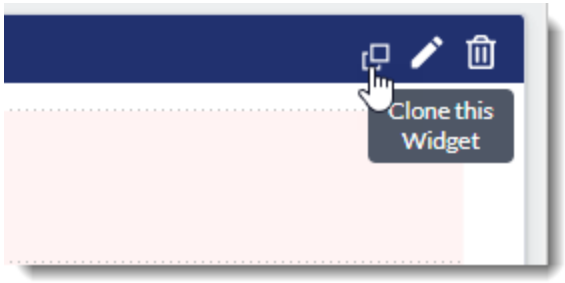
Number of items to display	Select the number of items to display on the widget.
Ordering	Select the ordering of the items. • Display highest first • Display lowest first
Item Aggregation	Aggregated data can be calculated based on the following values: **Values**: • By average • By minimum • By maximum
Time Range	Shows the aggregated availability percentage of a group of instances based on the selected time range. **Values**: • Last 15 minutes • Last hour • Last 6 hours • Last 12 hours • Last 24 hours • Last week • Last month
Background Color	Select a background color for the widget.
Data Points	The following data points are available. You can toggle each data point on or off by clicking the **eye** icon. Click the **up** or **down** arrow icons if you want to reorder them. • Server • Metric • Value • Status

Clone a widget

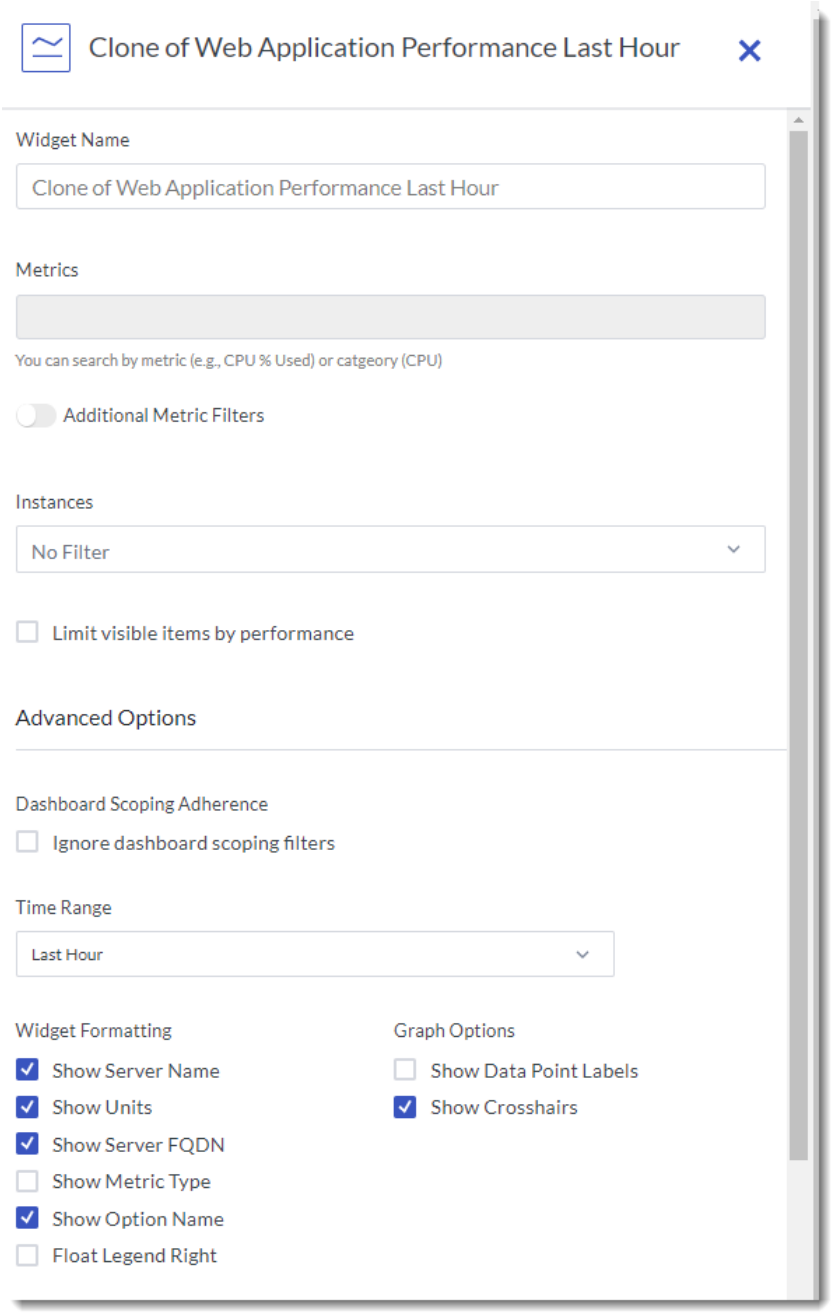
Cloning a widget allows you to duplicate an existing widget and copy all of its configuration (height, width, metrics, servers, etc).

To clone a widget, perform the following steps:

1. Go to the Dashboard where you want to duplicate a widget.
2. Click **Edit**.
3. Click the **Clone** button.



A copy of the widget is created with the title **Clone of <ORIGINAL WIDGET TITLE>**.



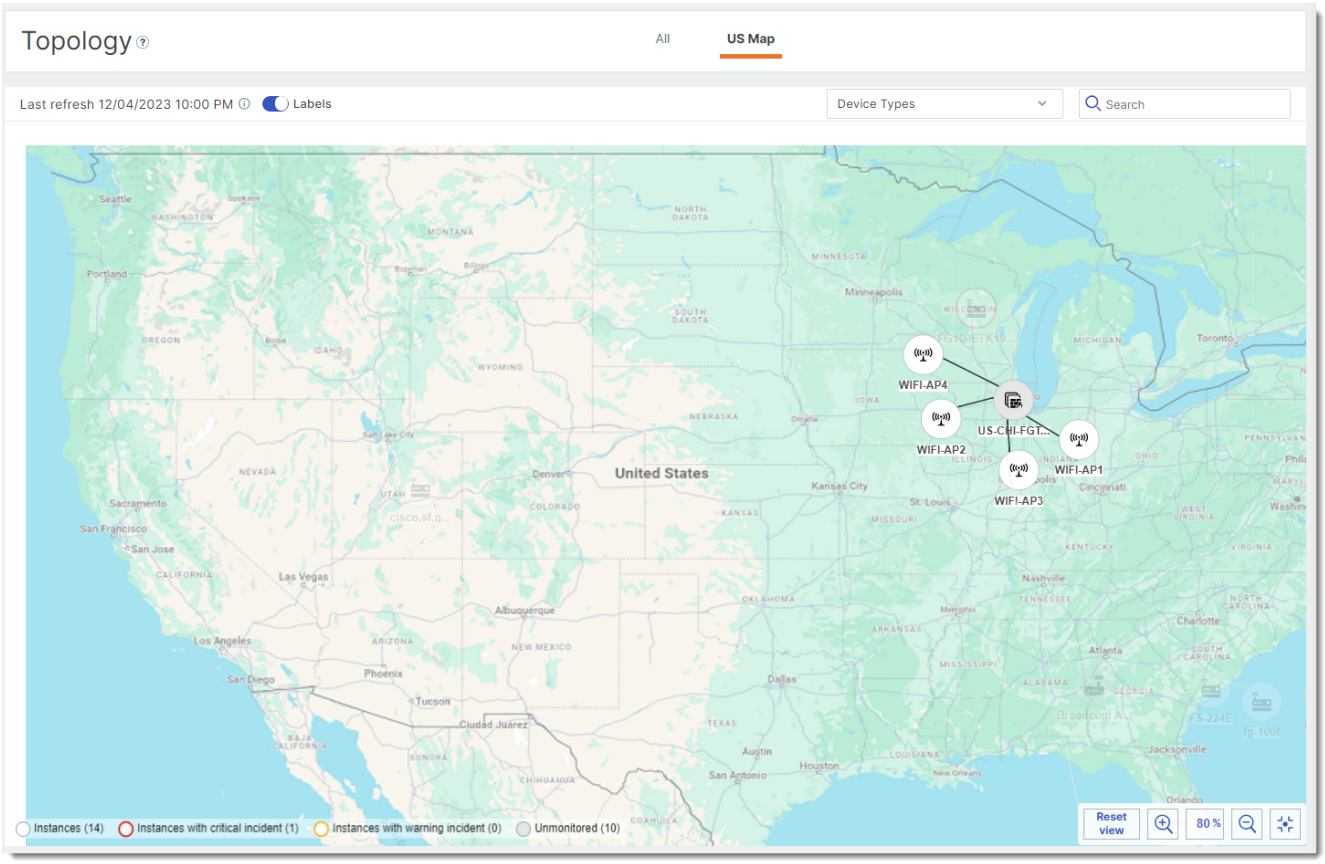
4. From here, you can modify the configuration as needed.
5. Click **Create Clone**. The cloned widget will now be displayed in the Dashboard.

Topology Maps

Topology Maps help network and IT administrators to find the interconnections and relationships between devices in an IT infrastructure. With an overall view of the network, administrators can better visualize the dependencies of devices.

Filters, search, and toggles can be used to narrow down or customize the map. Multiple topology views can also be created.

Scans of the whole IT infrastructure is performed nightly to update the map.



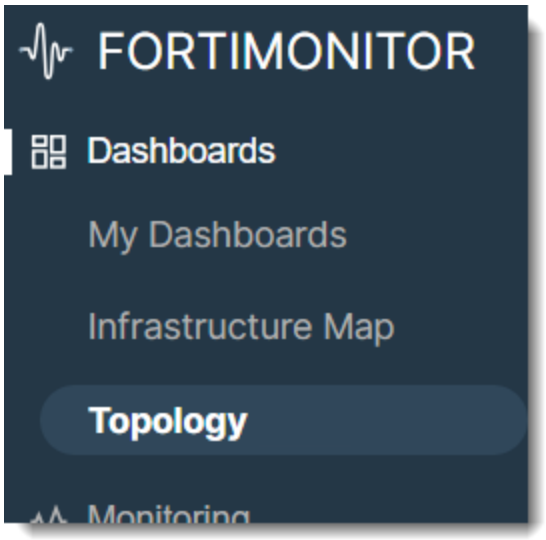
Data collection

- SNMP network devices by performing SNMP walks against the LLDP and CDP MIBs on each monitored network device. Enable LLDP or CDP on network devices to expose this data.
- Fabric network devices topology data directly from the Fabric APIs used for discovery and provisioning.
- VMware infrastructure topology data from the VMWare APIs used for discovery and provisioning.

Note: If you enable LLDP/CDP or add Fabric integrations, the topology map will not be updated with new data until after the next nightly scan.

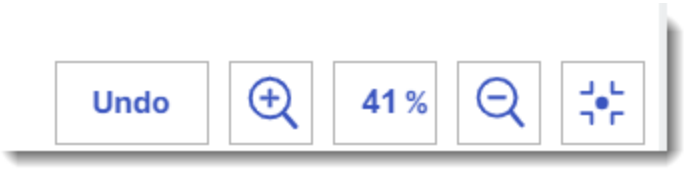
Topology page

To get started, select **Dashboards** > **Topology** from the navigation menu.



Basic controls

With the available basic controls, you can zoom in/ out the map using the magnifier icons. Click the **Fit View** button to view the entire map on your screen.



Filters and toggles

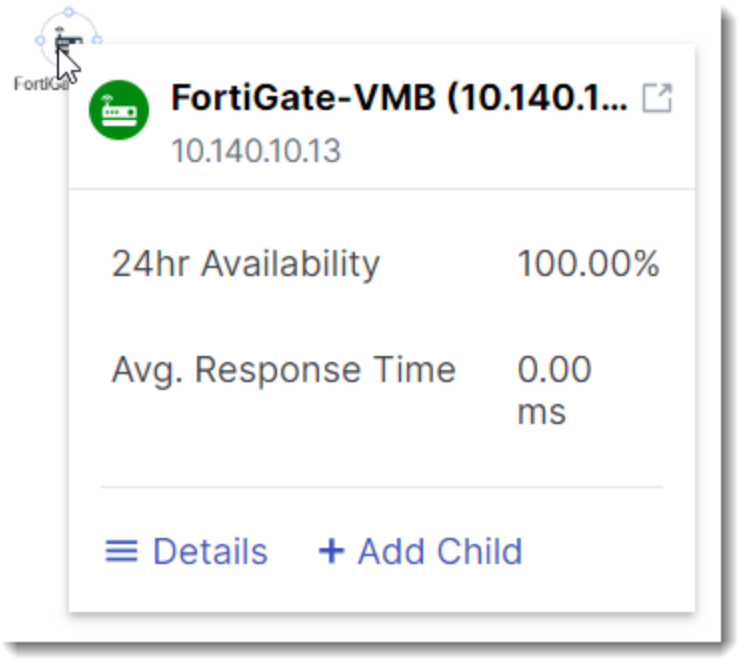
The following filters and toggles are available:

- **Labels** – Toggle to display labels
- **Device Types filter** – Select the devices you want to display from the dropdown
- **Instances** – Only show monitored instances without an incident
- **Instances with warning incident** – Only show monitored instances with an active warning incident
- **Instances with critical incident** - Only show monitored instances with an active critical incident
- **Unmonitored** – Only show unmonitored instances

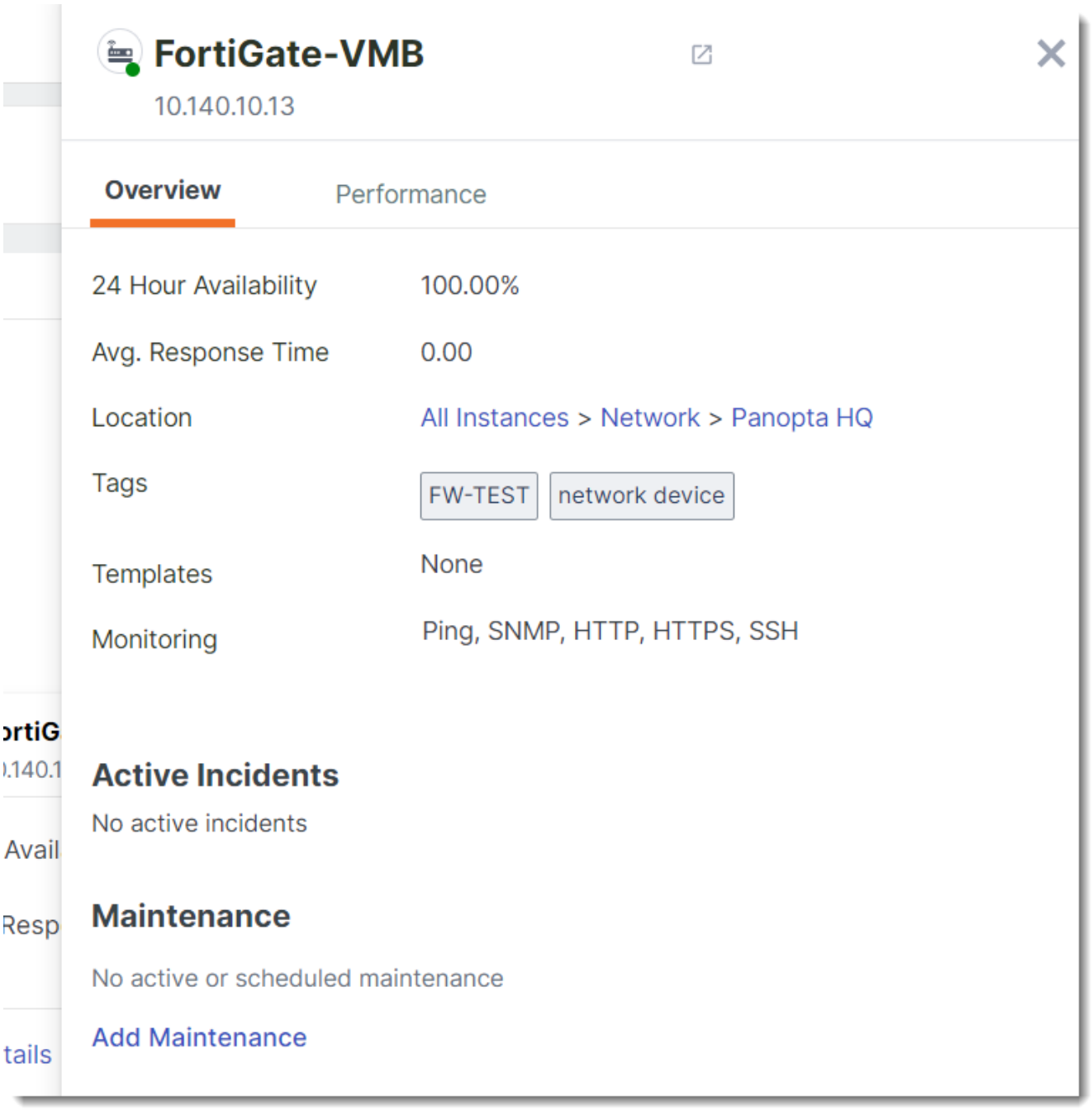
You can also use the search box to search for and display specific network devices.

View instance details

Click a monitored instance to view its availability and response times.



For a detailed view of the instance, including its maintenance schedule and incidents, click **Details**. Note that you can also add a maintenance from the instance drawer.

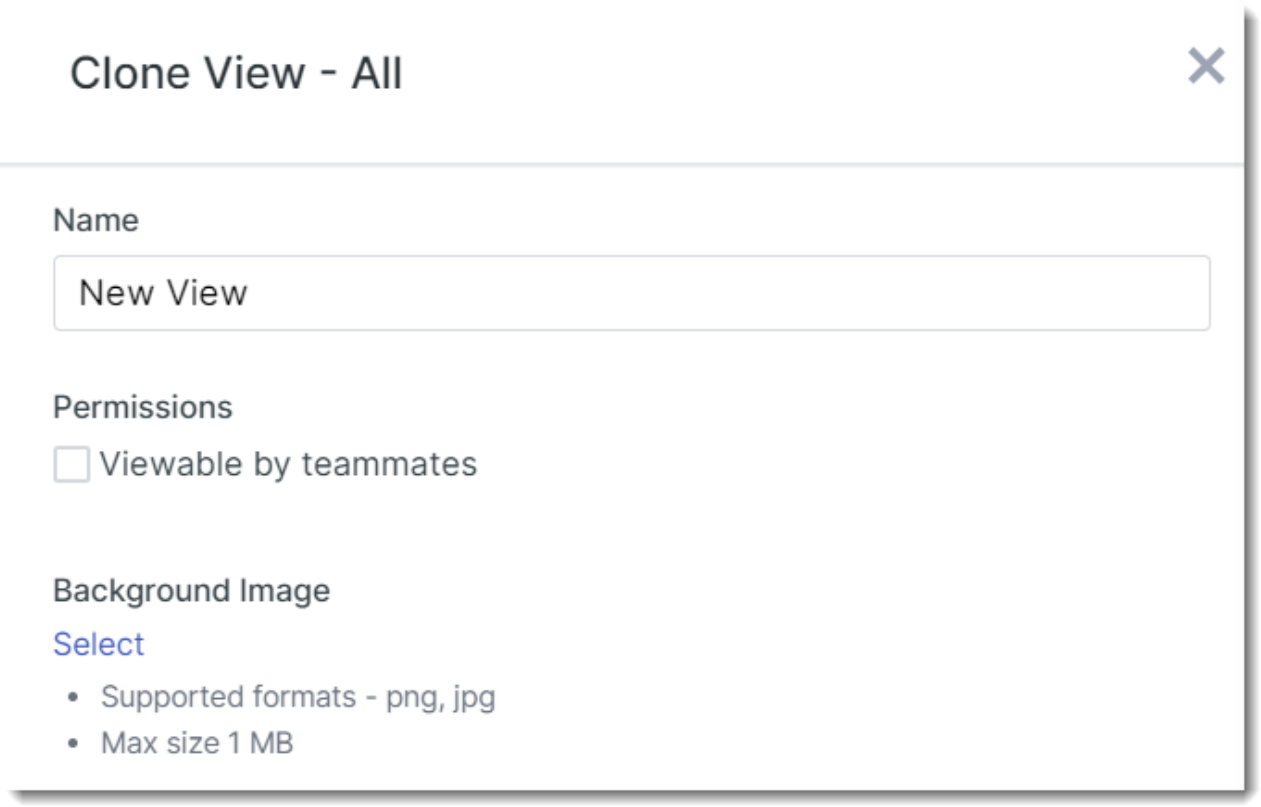


Monitor an unmonitored instance

Create and save views

Multiple topology views can be created by performing the following steps:

1. Click the **3-dot menu** beside the name of the view that you want to clone or use as a base for the new view.
2. Click **Clone**. The **Clone View** drawer slides out.

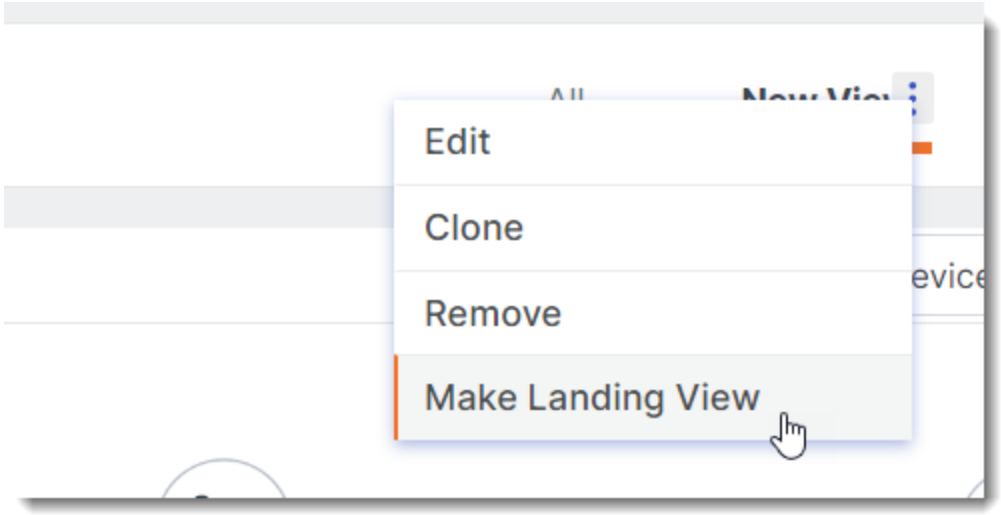


3. Enter a name for the new view.
4. Enable or disable **Viewable by teammates**.
5. (Optional) Choose a background image.
6. Click Continue. The new view is displayed.
7. Click **Save** to save the view.

Set the default view

To set the default view, perform the following steps:

1. Click the **3-dot menu** beside the name of the view that you want to set as the default view.
2. Click **Make Landing View**.



Monitoring

FortiMonitor allows you to monitor your public and private infrastructure using **three different dimensions** of monitoring. Using all three dimensions together allows you to correlate different types of events/incidents across your infrastructure and stack.

- Monitor publicly accessible servers and services using our globally distributed monitoring probe network. Test for uptime, performance and synthetic transactions from any of our 50+ PoP's.
- Use the FortiMonitor OnSight vCollector to achieve the same in-depth monitoring and secure reach servers and devices from behind your firewall.
- Install our Server Agent to collect key operating system health and application performance metrics and be alerted when different thresholds are reached.

No Unaddressed outages

FortiMonitor believes that any outage should be addressed, whether its because a server is slow, a threshold is too sensitive, or the network is finicky. For example, if a threshold is too sensitive, change it to alert when a problem is really happening. If you get an alert, it should be actionable. If you get used to seeing outages in your dashboard, you'll get alert fatigue.

No Persistent Alerts

Another aspect of the above, is that FortiMonitor won't continue to alert you on the same outage until someone responds. FortiMonitor only sends alerts that you have configured in your **Alert Timeline**. If you want to continue to

receive alerts after an outage to remind you, you'll need to factor that into your configuration.

FortiMonitor is an extension of your team

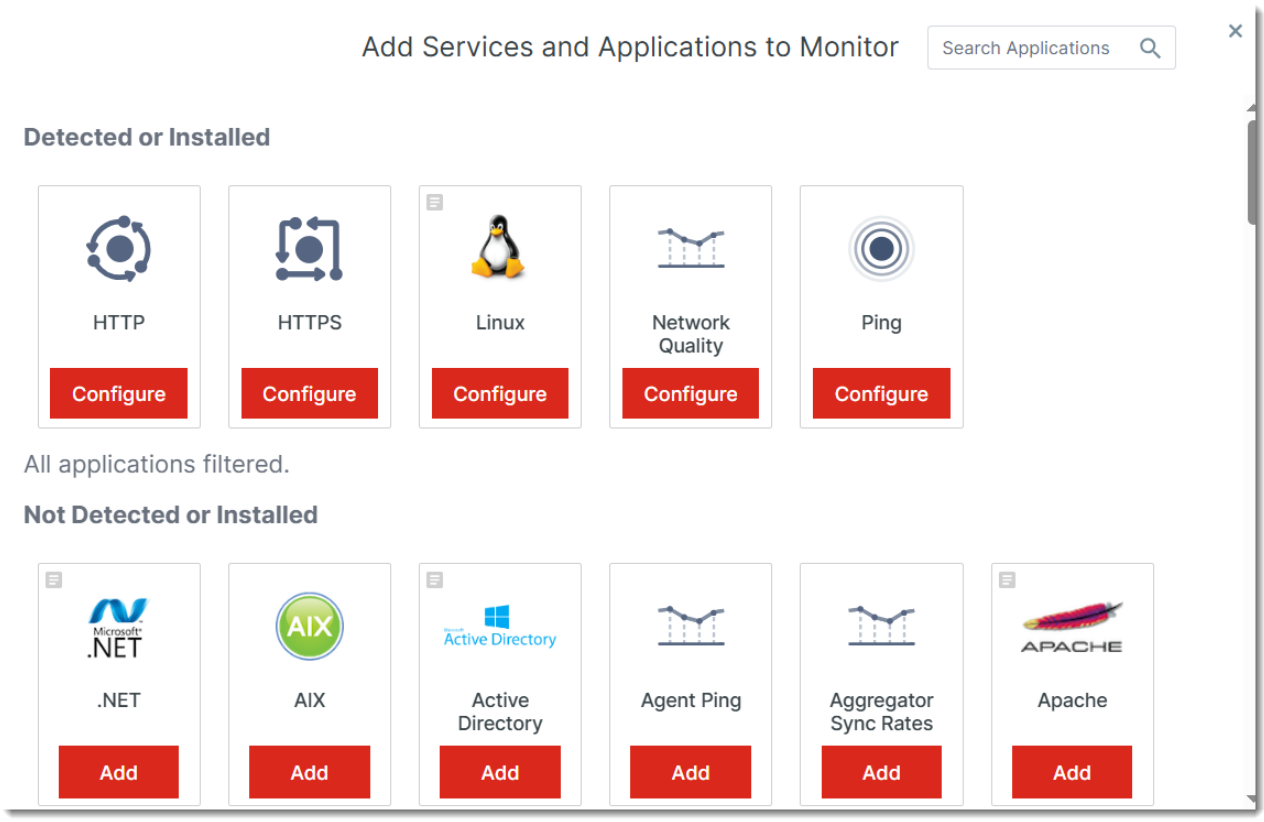
FortiMonitor wants to understand your job, your business, and your needs. Have an idea for a plugin we don't have yet? Let's build it together! Have a complex check that needs to be run and you aren't sure if we can handle it? Let's find a solution! Are we missing some functionality you think could help? Let us know. 80% of our feature development is customer-driven, so let us know and we'll give you a timeline for when we can have it for you.

Add monitoring to an existing server

Network service checks allow you to monitor external access to your servers and webpages.

To add a network service check to an existing server

1. Go to an instance's **Instance Details** page.
2. Select the **Config** tab.
3. Click **Add Monitoring**.
   The **Add Services and Applications to Monitor** modal is displayed. This modal shows you the checks you already have configured, the services detected on the server that are available to be monitored, and the rest of the catalog of available checks and plugins.



4. Choose the check type you would like to add by selecting **Add**.
5. For specific instructions on how to add, see the catalog of available checks and plugins.

Monitoring check types

Network service checks allow you to monitor your server or webpage via a network (most likely the internet). Most network service checks are simple port checks, for example an FTP port check is just checking to make sure the FTP port on your server is responsive. However there are a many more complicated options as well.

If you are new to monitoring we recommend you start by using either the ping or the simple HTTP check for your websites and servers.

Setting up a network service check

You can setup network service checks while creating a server, but you can also add a service check to a server that you have already created. This article will walk you through the latter, as well as provide links to help articles on the more complicated network service checks.

HTTP Check Types

In addition to a basic HTTP port check, there are two more in-depth checks which you can setup. See the pages for HTTP HEAD checks and standard HTTP respectively for more details.

Delivery Time Checks for mail servers

Both the POP and the IMAP service checks have delivery time checks, which allow you to measure the amount of time it takes for your e-mail server to receive an e-mail.

DNS

The DNS network service check behaves very differently from the other service checks. It is a very valuable tool that we encourage our customers to take a look at. To learn more about DNS monitoring see DNS.

Monitor Flapping Metric incidents

A Flapping Metric incident helps you track whether a particular server resource or monitor point associated with a server instance is generating incidents frequently within a specified time window. You can set the minimum number of incidents required to trigger an alert. In addition to individual metrics, Flapping Metric incidents also work for instance-level heartbeat alerts such as SNMP or Fabric.

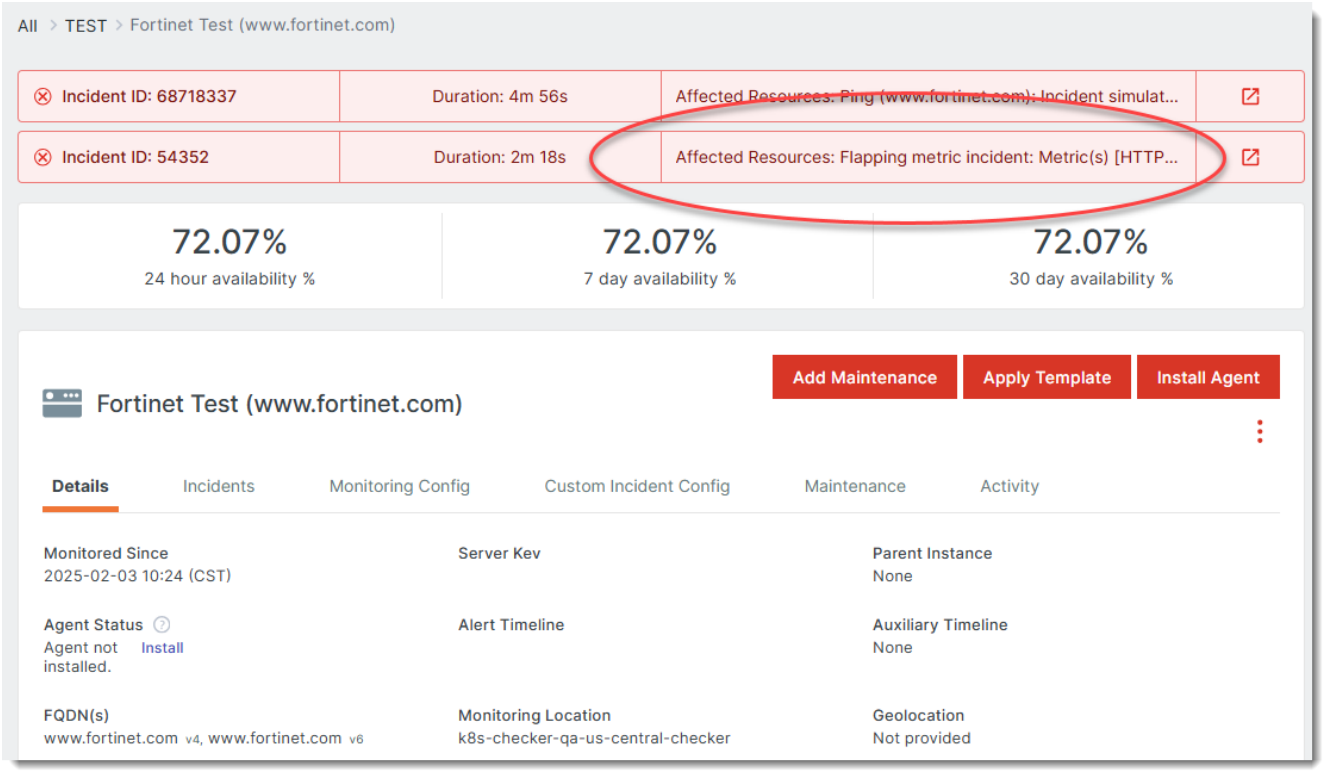
For example, a server has two resources: CPU and Memory. You might have set thresholds such as CPU usage above 80% for 5 minutes and Memory usage above 90% for 5 minutes. You can configure Flapping Metric incidents to monitor whether any of these resources are causing too many incidents. If you set a 2-hour time period with a minimum of 3 incidents, then during that time, 6 CPU incidents and 2 Memory incidents occur, then only the CPU incidents would meet the threshold. A Flapping Metric incident for the CPU will then be triggered for your server. Flapping Metric incidents will remain open indefinitely until manually closed.

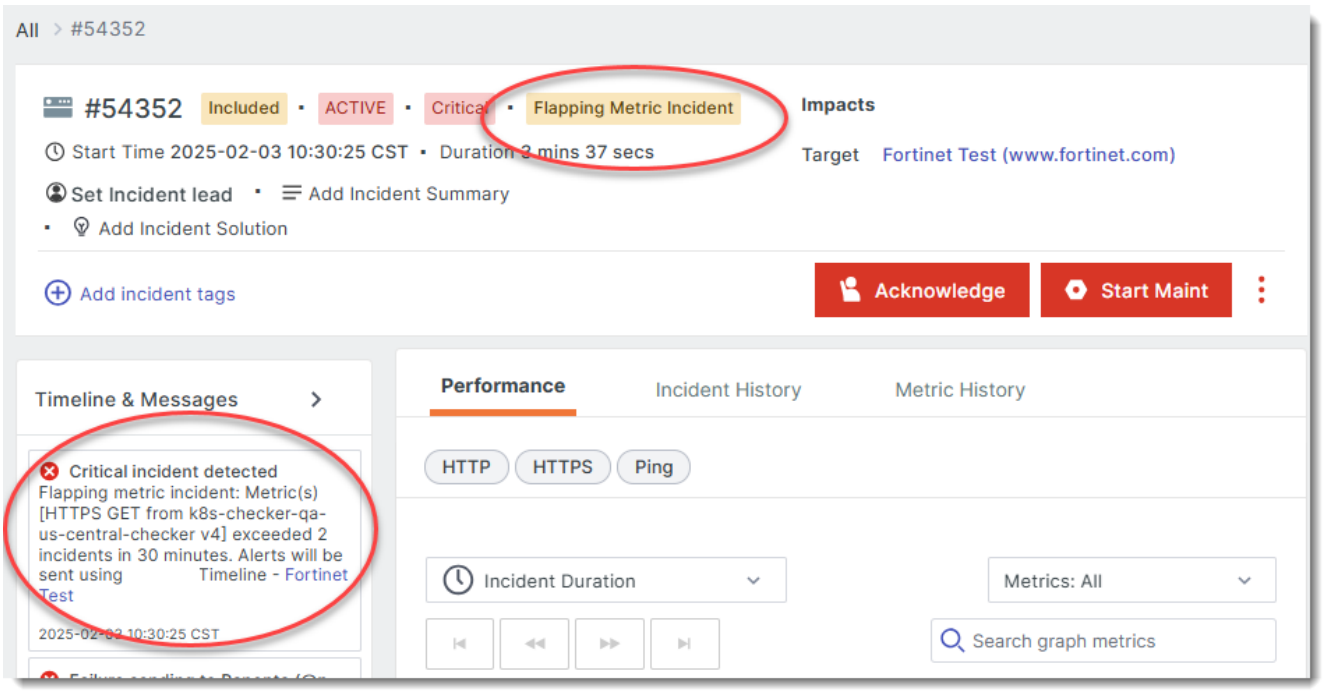
Note: Flapping Metric incidents being enabled do not suppress alerts for other metrics or alerting on the instance.

Flapping metrics calculations are performed at 30-minute intervals. Also, note that flapping metric events are only calculated after a threshold is configured on an instance. After configuring the threshold, events that are in the scope of the specified time window are included in the calculation.

For example, a flapping configuration is created at 9:42. At 10:00 it will run and check from 9:30 to 10:00 if there were enough events to trigger the condition.

Flapping Metric incidents, when they occur, are displayed in the **Instance Details page** or the page of the incident that caused the flapping condition.



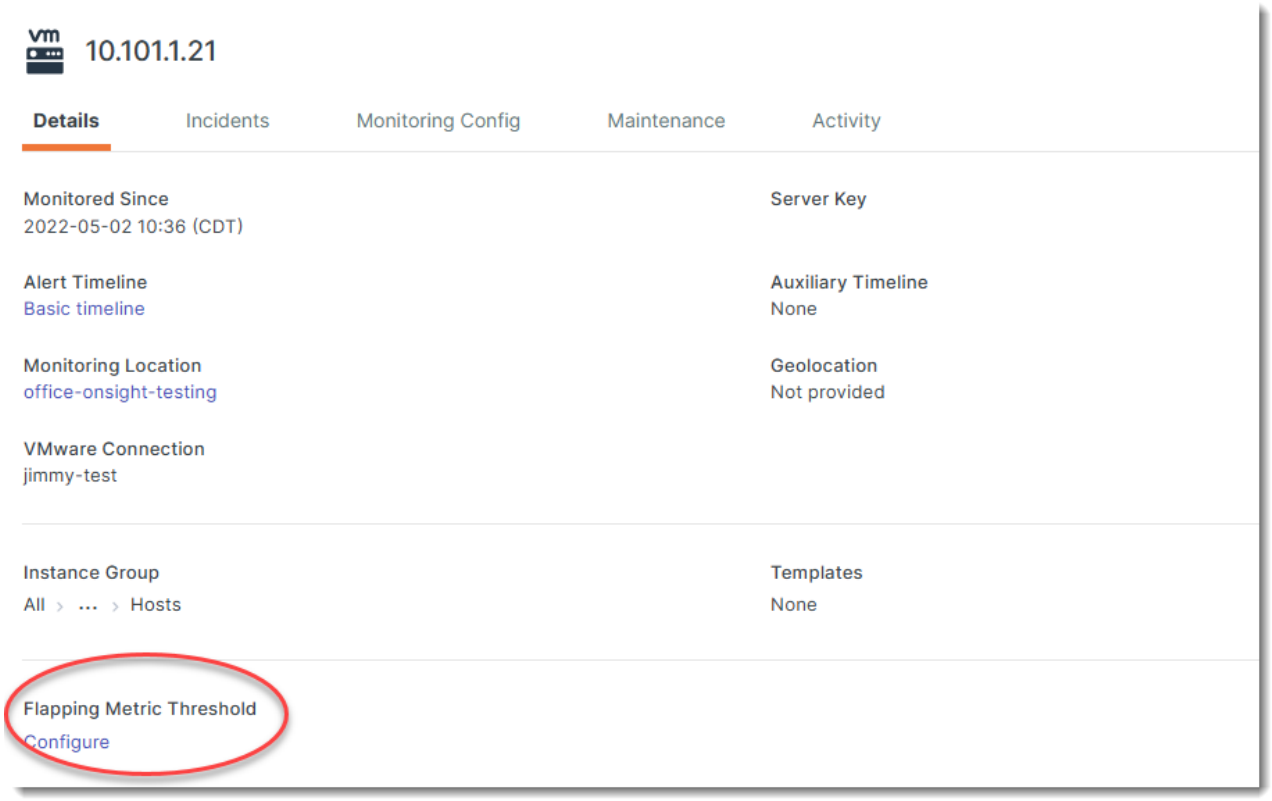


Note: If multiple metrics breach the configured Flapping Metric threshold, they will be displayed under one incident. Expanding the description will list the metrics that were above the threshold.

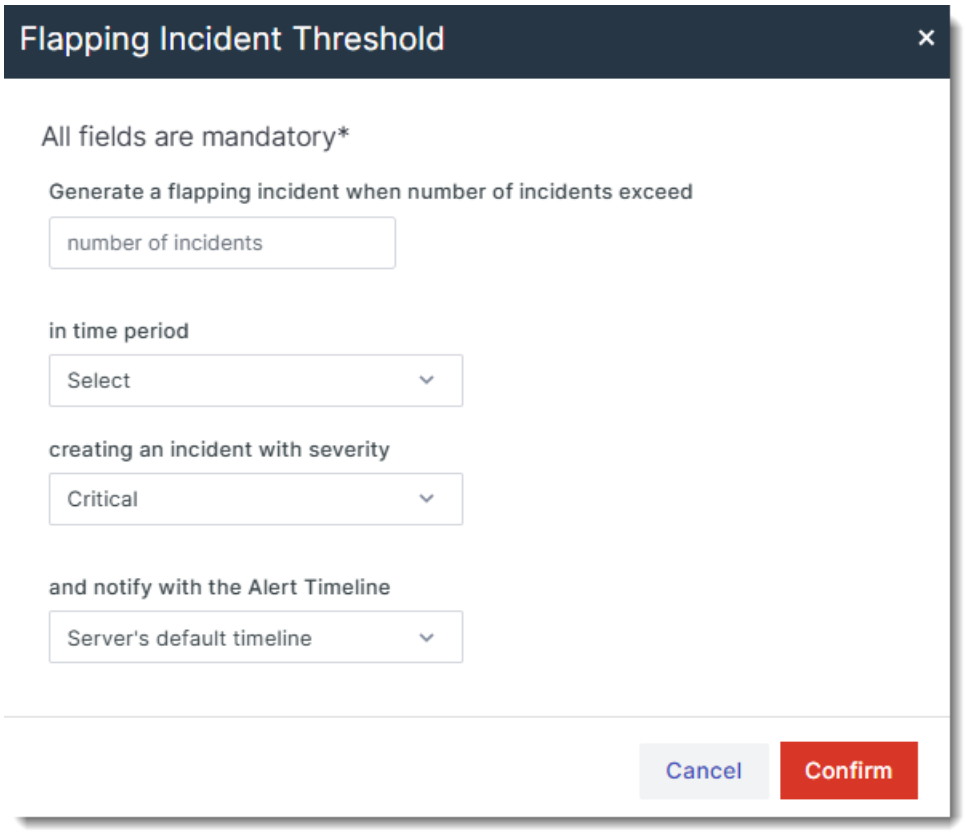
Configuration

To configure a Flapping Metric incident, perform the following steps:

1. Go to the Instance Details page of the server that you want to configure.
2. Under **Flapping Metric Threshold**, select **Configure**.



The **Flapping Incident Threshold** modal is displayed.



Each configurable option is described in the following table.

Field	Description
Generate a flapping incident when number of incidents exceed	Specifies the minimum number of incidents for any server resource or monitoring point within a server instance that must occur to trigger a Flapping Metric incident.
in time period	Defines the duration (from 30 minutes to 24 hours) during which incidents for any server resource or monitoring point will be considered.
creating an incident with severity	Indicates the severity level for the alert, either as a warning or critical.
and notify with the Alert Timeline	Determines the schedule for notifying users about the Flapping Metric incident.

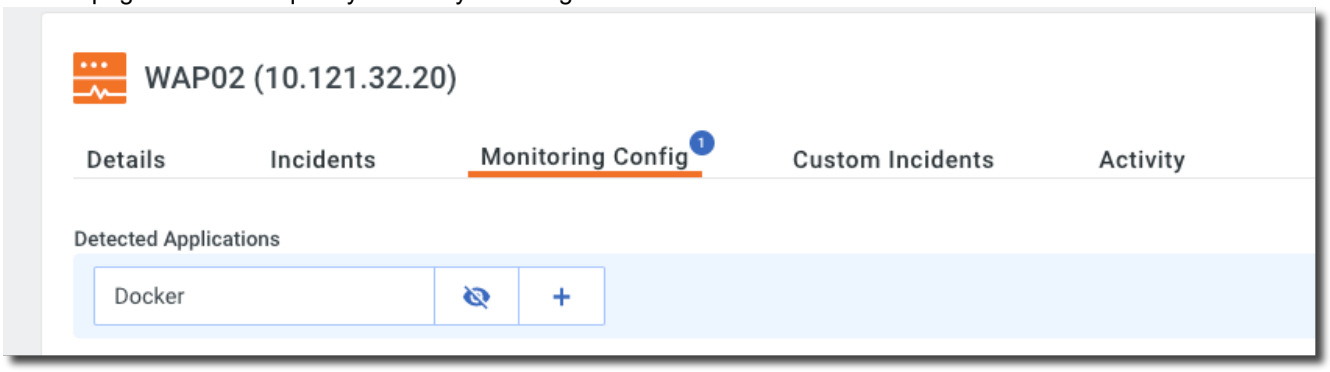
You can also use templates to configure Flapping Metric incident thresholds.

Advanced monitoring and tuning

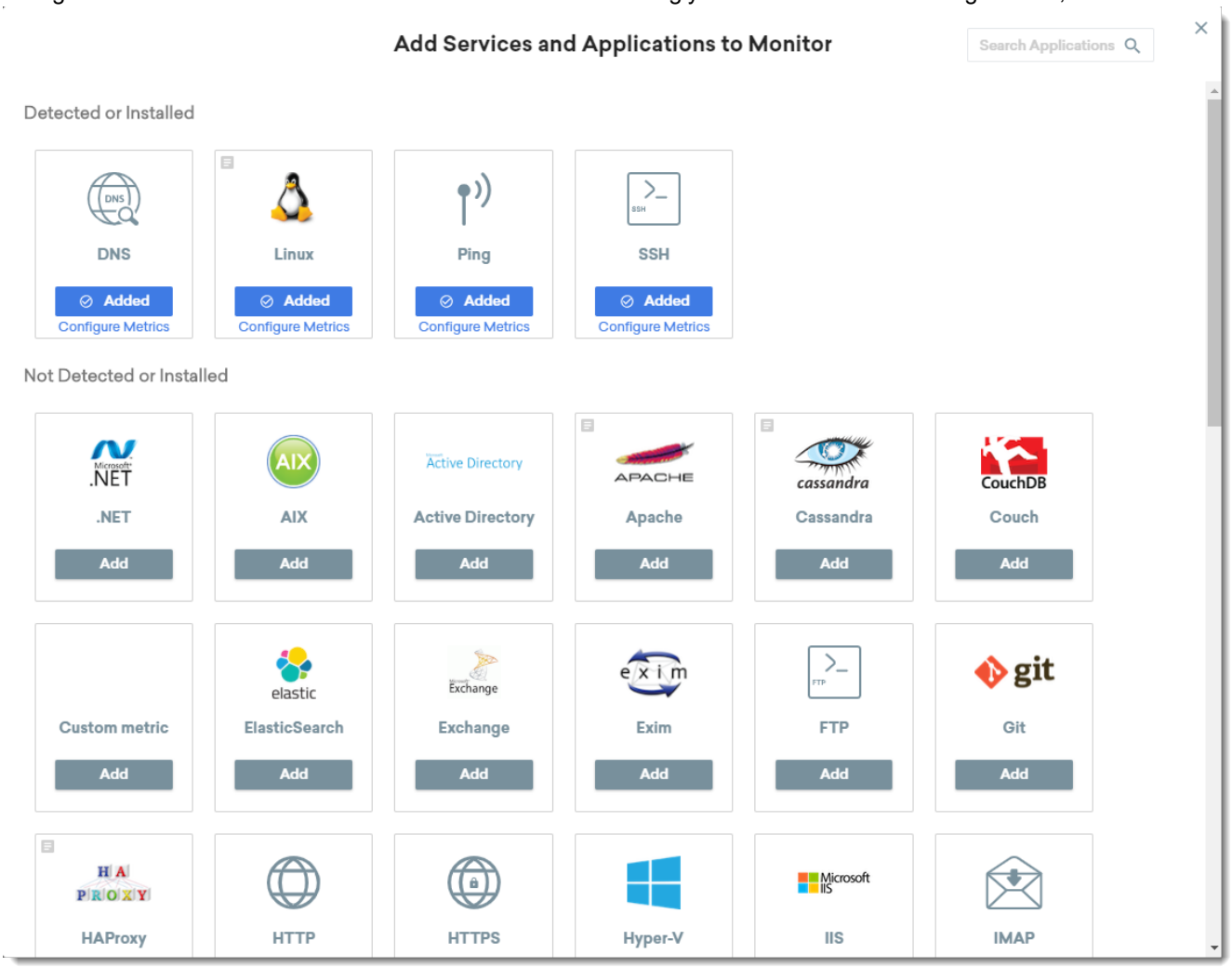
Adding Monitoring

Adding the additional enhanced monitoring can be done on the instance page itself.
On each instance page, under the monitoring tab (see below), checks and metrics which have been already added will appear under the light grey headers. Each grey header represents a different high level monitoring category. To add more monitoring within those existing categories, click on the grey “Add Metric” button to the right.

FortiMonitor also attempts to automatically discover other services and applications which are present on the server. Those discovered categories will appear under **Detected Applications** in the **Monitoring Config** tab of the **Instance Details** page and can be quickly added by selecting the **+** button.

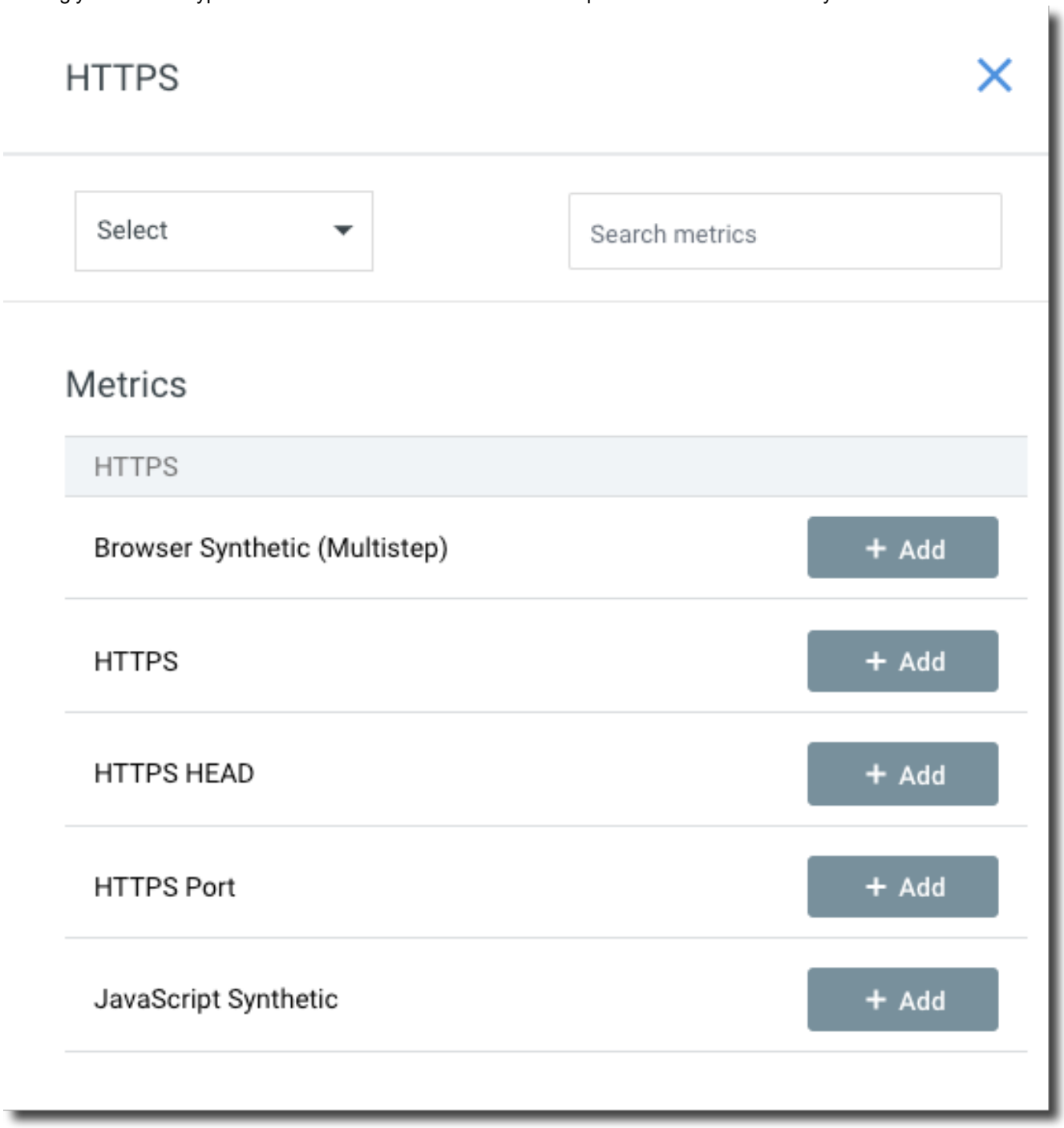


To add other categories of monitoring, select the **Add Monitoring** button above the table. This will expose a catalog of different out of the box monitoring capabilities for application stacks on both the Windows and Linux platforms. Certain applications may require additional configuration to be auto-discovered and become operation. Specifics on each plugin configuration can be found here. For more information on creating your own custom monitoring metrics, see here.

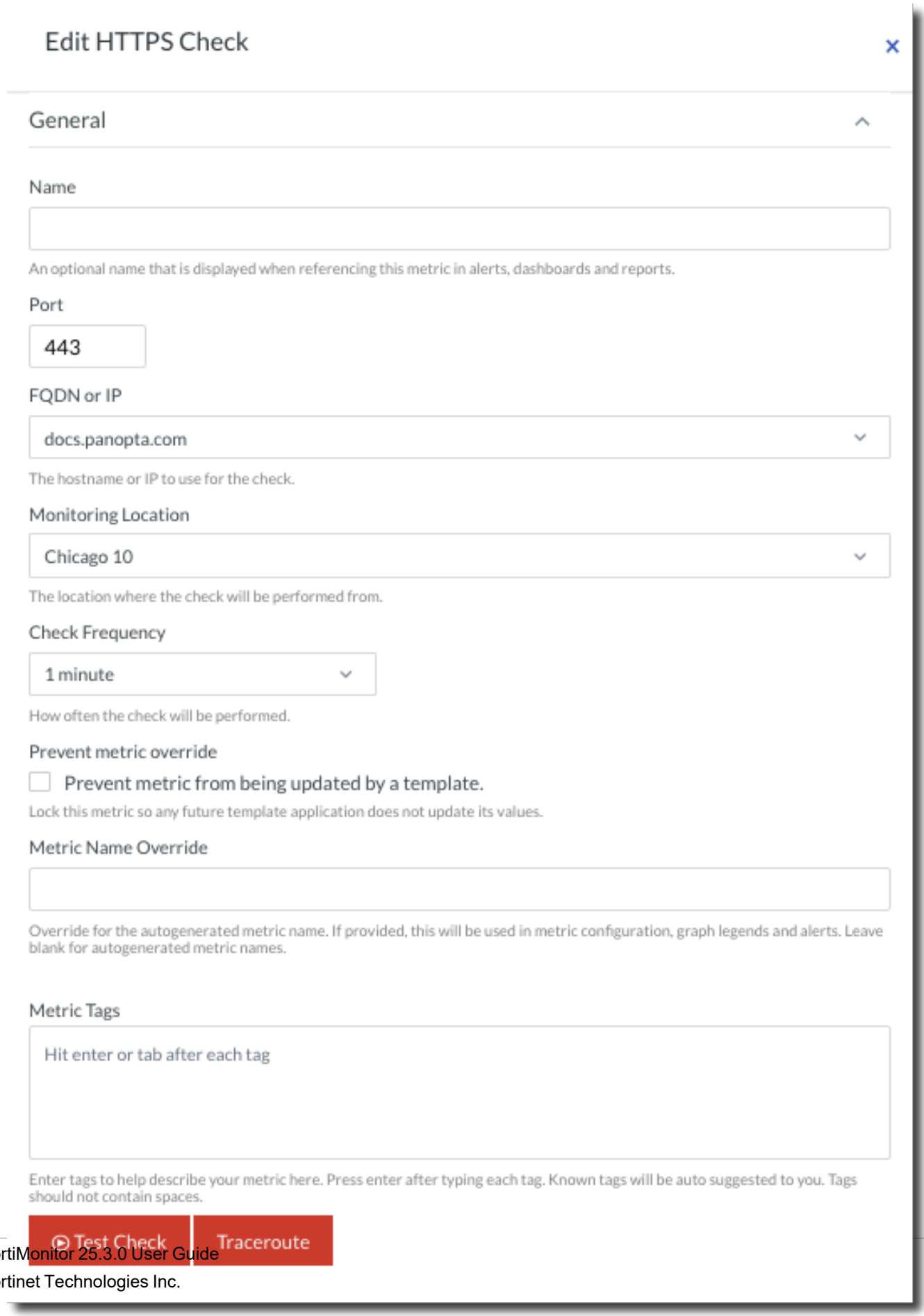


For detected metrics, you can click either the **Add** button, or the **Configure Metrics** button for already added checks.

Once you click the **Monitor** button on any of the discovered or selected categories, a drawer will slide out from the right offering you different types of checks and metrics. Here's an example of what is shown when you select HTTPS.



Choose from one of the check types by clicking **+ Add**. Doing so will open a window presenting different configuration options. Let's choose a standard HTTPS web check.


Edit HTTPS Check
General
Name
An optional name that is displayed when referencing this metric in alerts, dashboards and reports.
Port
443
FQDN or IP
docs.panopta.com
The hostname or IP to use for the check.
Monitoring Location
Chicago 10
The location where the check will be performed from.
Check Frequency
1 minute
How often the check will be performed.
Prevent metric override
Prevent metric from being updated by a template.
Lock this metric so any future template application does not update its values.
Metric Name Override
Override for the autogenerated metric name. If provided, this will be used in metric configuration, graph legends and alerts. Leave blank for autogenerated metric names.
Metric Tags
Hit enter or tab after each tag
Enter tags to help describe your metric here. Press enter after typing each tag. Known tags will be auto suggested to you. Tags should not contain spaces.
Test Check
Traceroute

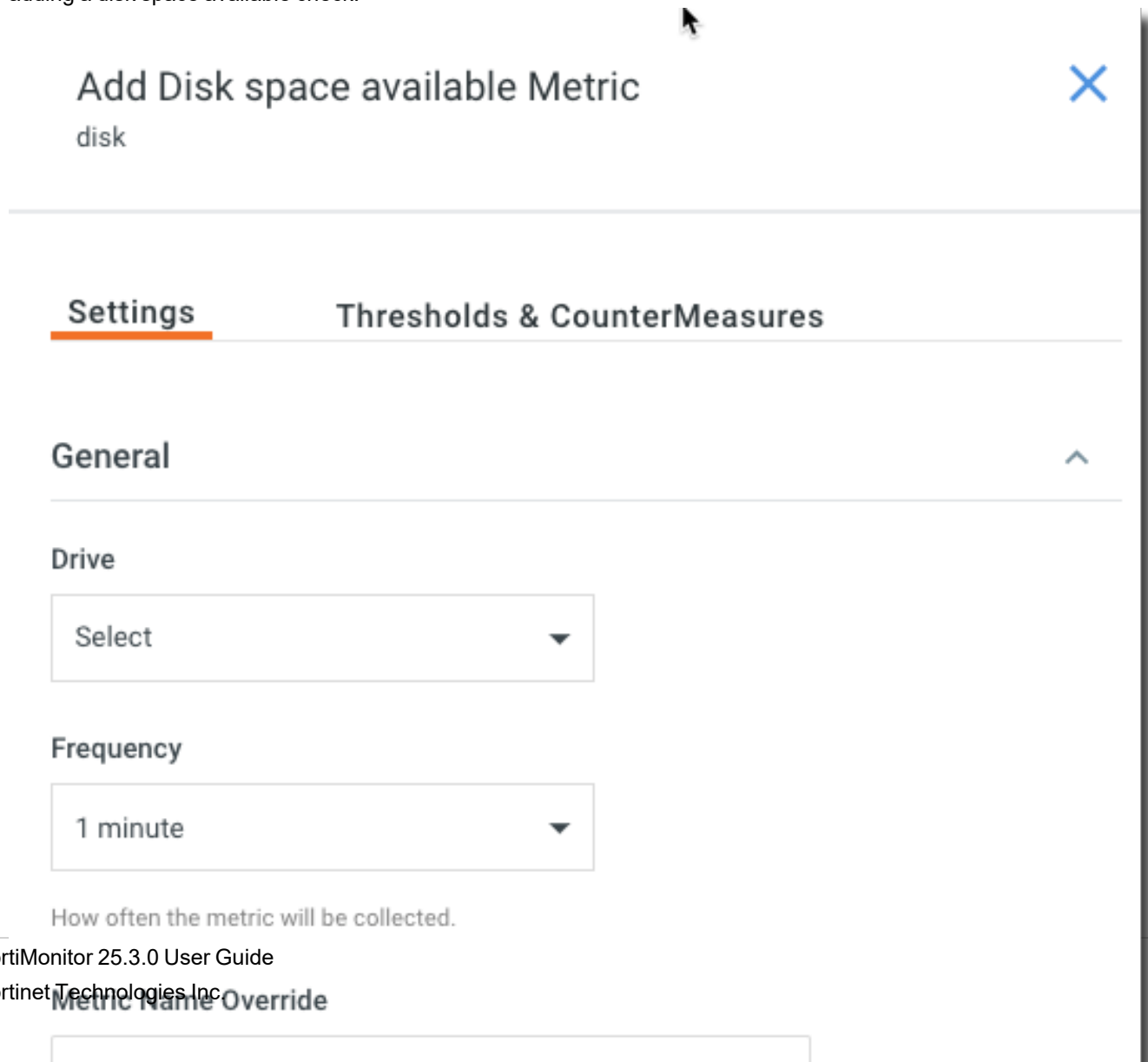
As you can see, there are a number of tabs for configuring the check, but not all of them are required. For HTTPS, there are options for authentication, simulating user requests, SSL/TLS verification and content matching.
In addition, there are additional options that will allow you to override auto-generated names and template settings:

- **Metric Name Override** – Enter a metric name to override the auto-generated name. If provided, this will be used in metric configuration, graph legends, and alerts. Leave blank to use auto-generated metric names.
- **Prevent metric override** – Prevent a metric from being updated by future template applications.
  Applying a template to the instance will not override the current setting of the metric. For example, you applied a template which sets up SSH checks for Linux servers using default port 22 but you also have a few Linux servers that have SSH on a different port. After template application, you can update the metrics to use the correct port and enable the **Prevent metric override** option. By enabling this option, you ensure that the metric will not be move back to port 22 after subsequent template applications.

Within the **Thresholds** tab, you'll have the option to choose a specific alert timeline if different from the server's default alert timeline. This can be useful for sending different types of alerts to different teams or to tune the alerting aggressiveness of specific failures.

Adding Agent Metrics

Adding a metric which is monitored by the FortiMonitor Agent follows roughly the same steps. Here is an example of adding a disk space available check.



Under the **Settings** tab, select the appropriate options for the metric along with the frequency of collection. Then, set any number of thresholds on the **Thresholds & CounterMeasures** tab which can each use a different alert timeline for notification.

Tags

Adding infrastructure tags at this point is optional, however if you'd like to get started, tags can be set at the check level, as well as the server level. In this way, tags give you a flexible and powerful method to dynamically "query" your infrastructure within the different systems in FortiMonitor. Use tags to:

- Have dashboard widgets dynamically show the right metrics and infrastructure
- Run scheduled and ad-hoc reports across different segments of your infrastructure
- Schedule maintenance windows for certain slices of your environment

Now that you have your base set-up for a few servers finished, the next step is automating the process using templates to make adding instances efficient and standardized.

Automation

Now that you've added a few different kinds of servers, let's cover how to automate the onboarding of new infrastructure and streamlining your ongoing maintenance. There are two main methods to achieve this: templates and the FortiMonitor Agent manifest file. Templates allow you to capture all aspects of a server's configuration (monitoring configuration, thresholds, alert timelines, tags, etc.) and maintain it in one place. You can then apply one or more templates to a server to stamp out that configuration.

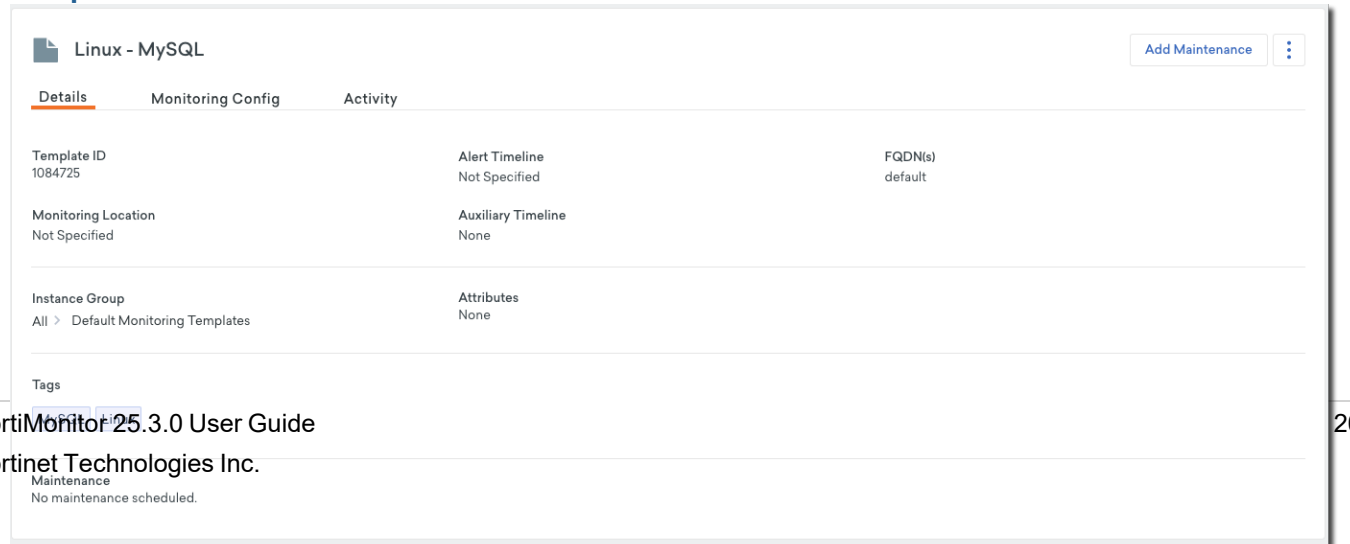
FortiMonitor Agent manifest file

The FortiMonitor Agent manifest file is a method by which you can apply certain features to a server, such as: server group, templates, tags, attributes, FQDN, server name, and plugin configuration. Read more about the agent manifest file in our help docs.

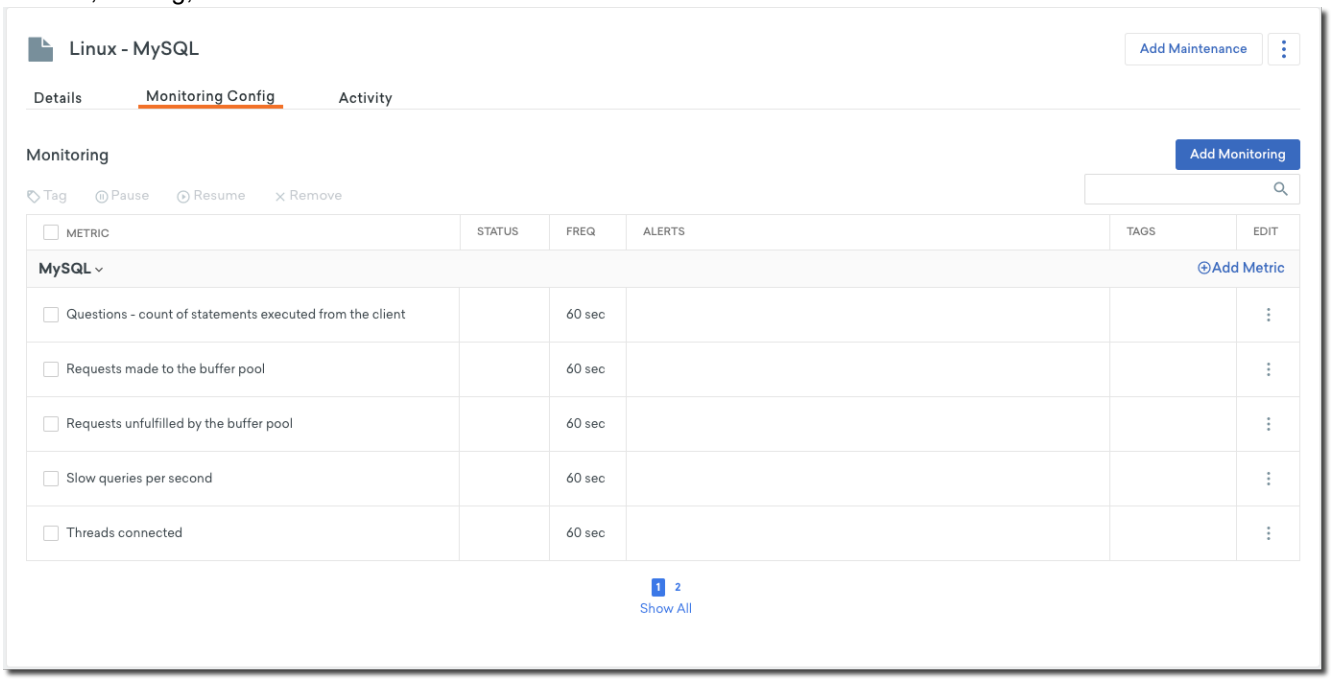
Template Creation

You can build a template from scratch or clone an existing server into a reusable template. If you're backing into using templates after your initial setup and already have monitoring in place, the templates will automatically match metrics which are of the same type to avoid any duplication.

Template View



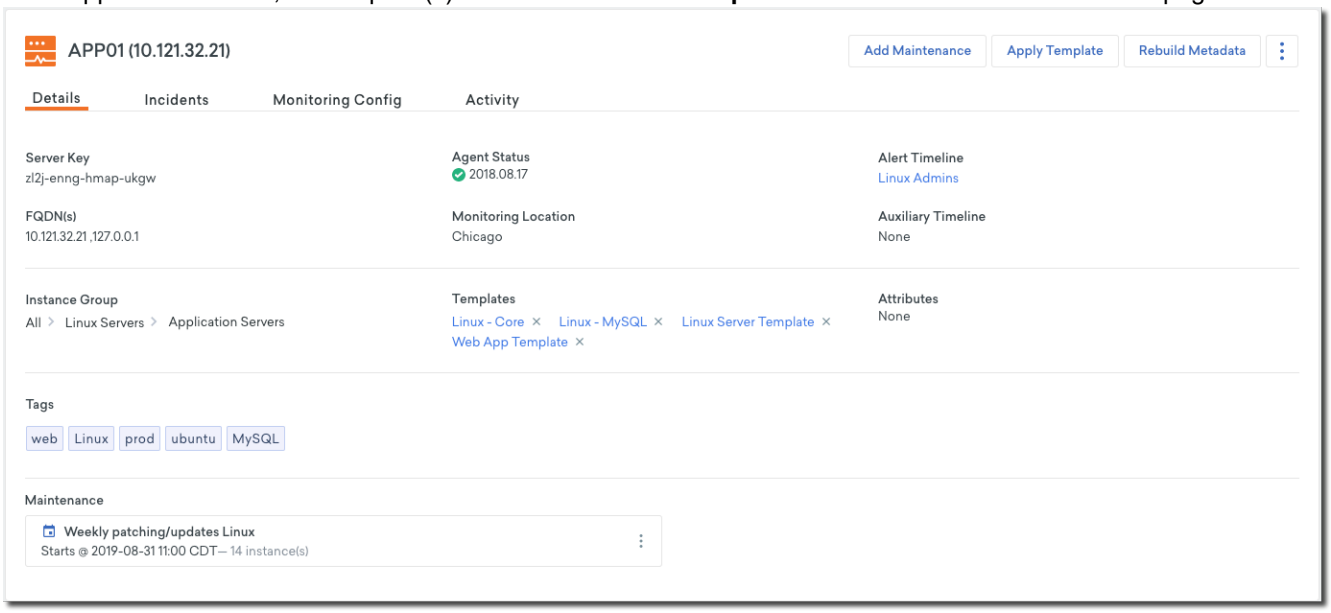
A template looks similar to a server but has a different icon to help you differentiate between the two. Checks and thresholds can be set up on templates the same way they're added to servers including setting options like monitoring location, alerting, etc.



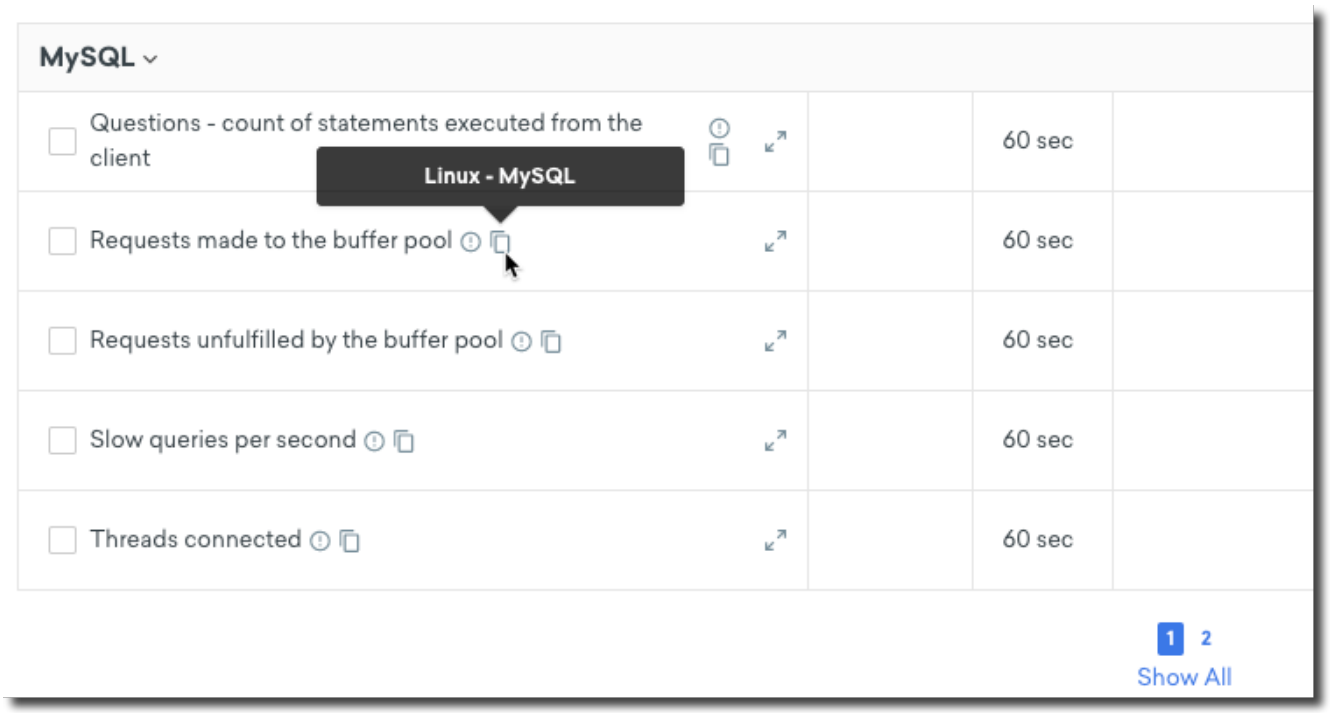
Templates can also be applied to network devices using SNMP.

Server View

Once applied to a server, the template(s) will be shown in the **Templates** section of the **Instance Details** page.



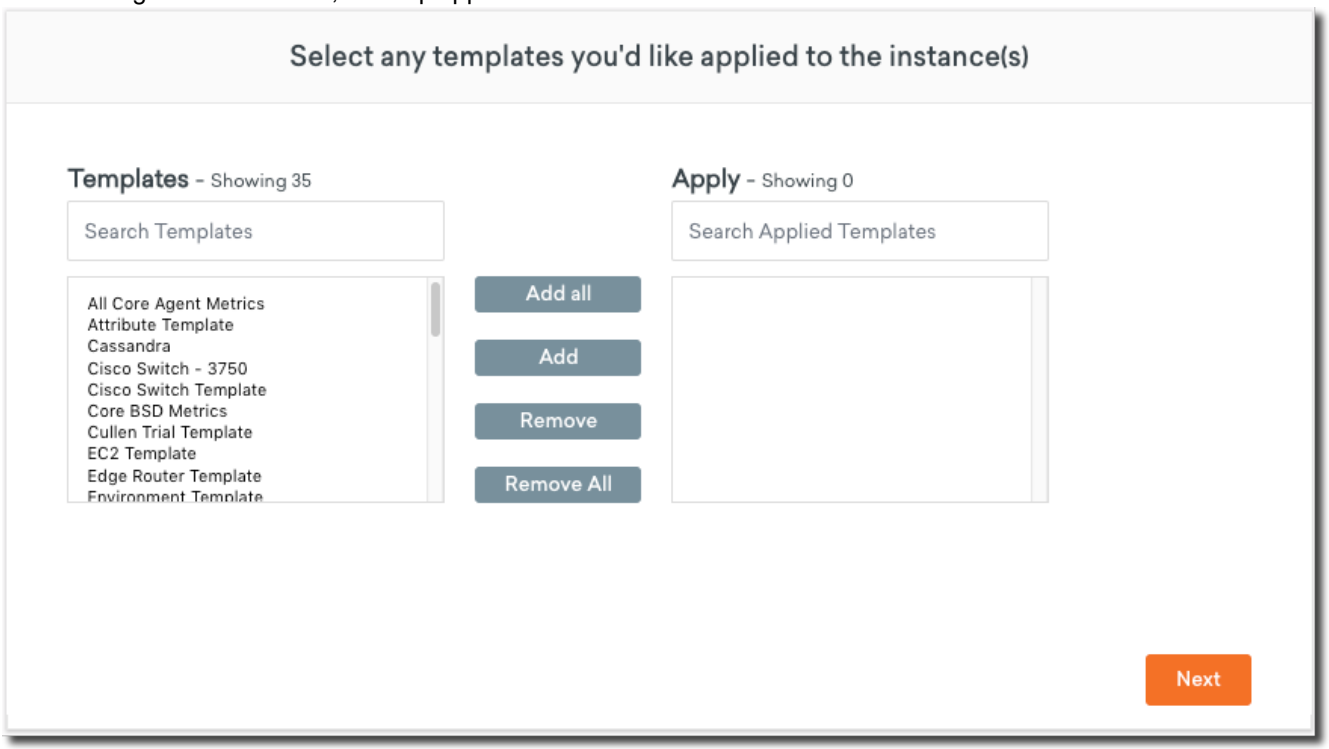
You can see which of the checks, tags, and attributes were applied to the server by the template icon next to each.



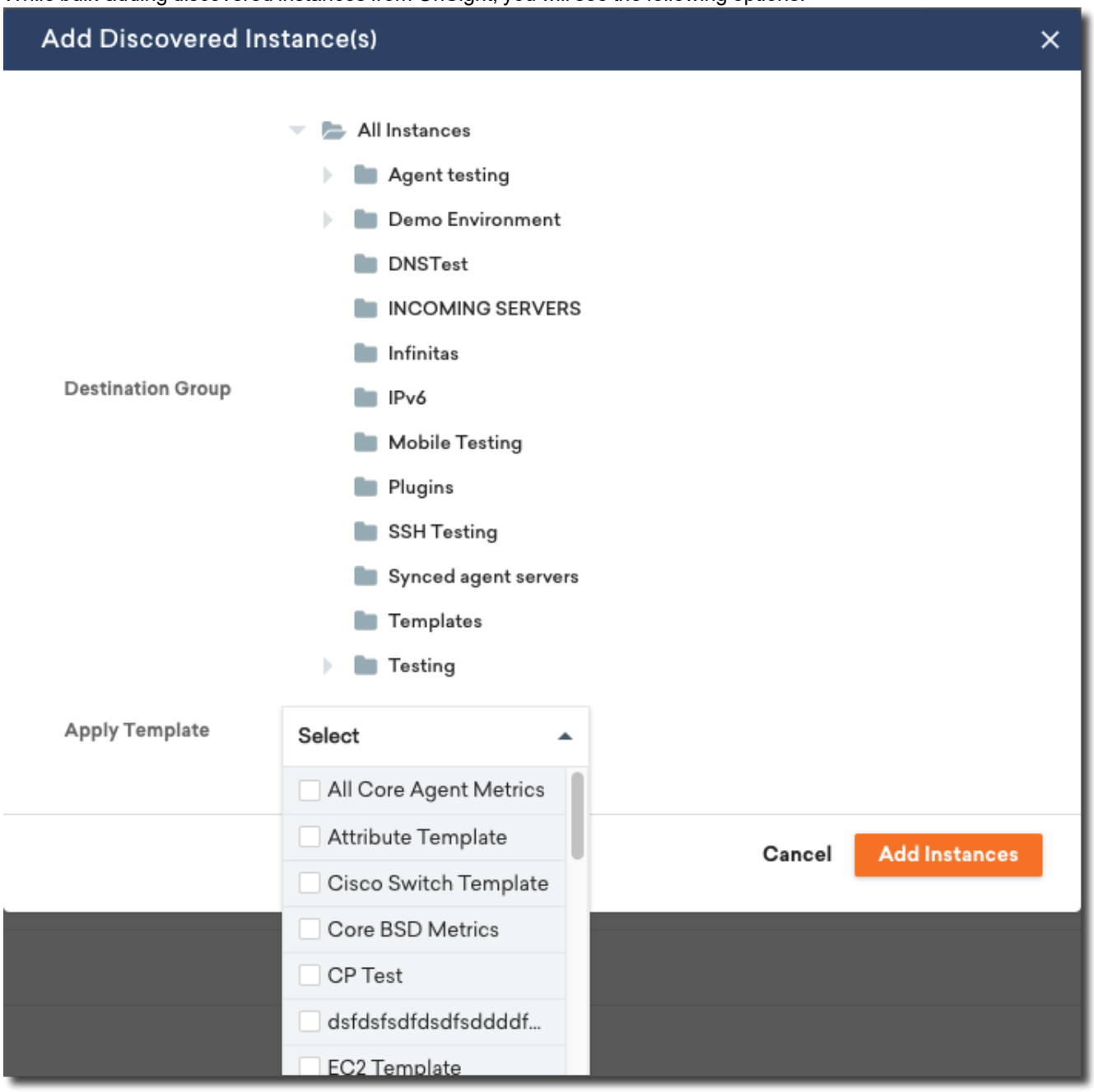
Other Methods of Using Templates

Even if you're not fully automating your deployment, you can still benefit from templates while manually adding instances or while adding discovered instances from OnSight vCollectors.

While adding a basic instance, the step appears like so:



While bulk-adding discovered instances from OnSight, you will see the following options.

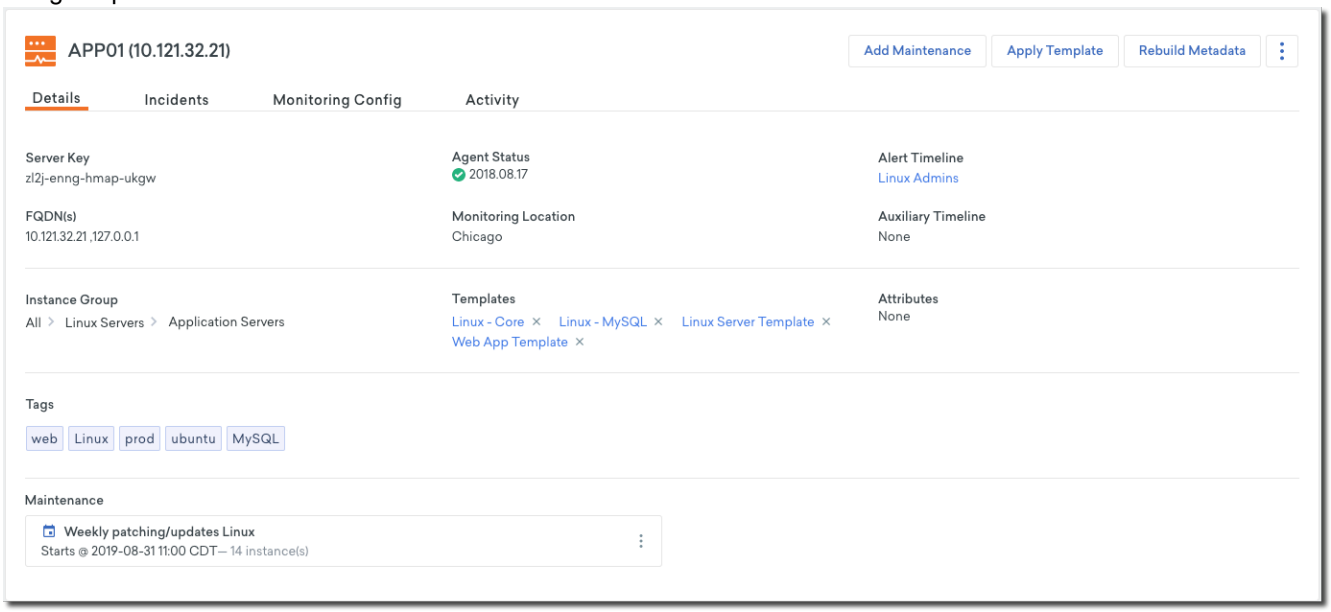


Easy Template Management

If you have many variations of servers, you may find it beneficial to create a modular structure with your templates. This will allow you to layer templates based on function. One common use case for this is creating a baseline template for all OS/system related metrics and then layering on application-related metrics. This makes your ongoing maintenance of templates easier and less error-prone.

Tags

As mentioned earlier in this guide, tags can provide a dynamic method for you to query your infrastructure. You can create any number of tags to capture things like: environment, server type, application type, responsible team, or any other internal taxonomy. For example, we've tagged this server with **web** for the function of the server, **Linux** for the OS, and **prod** to indicate that it is in our production environment. Tags can also be automatically applied to your infrastructure using templates.



You can centrally view and manage all tags when you got to **Settings** > **Tags**. From there, you can merge and delete tags, which will update all the places they have been used. Selecting **View applied locations** on each tag will show you where it is applied.

What can I tag?

- **Servers** – Helpful for organization and powerful server selection for starting maintenance windows, running reports, or generating dashboards. Heartbeat tags can also be added to Agent and OnSight-based instances.
- **Individual checks/metrics** – This is helpful when you want to pull reports or build dashboards based on checks with specific tags.
- **Network device ports** – This is helpful when you want to isolate and modify ports based on their function. For example, filtering ports with an **HTTP** tag will allow you to see only the ports associated with the tag. To tag network device ports, see Tag a network device port.

Tags and Attributes

- Tags
- Attributes
- Bulk upload

Tags

Tags provide a dynamic method for you to query your infrastructure. You can create any number of tags to capture things like:

- Environment
- Server type
- Application type
- Responsible team
- Other internal classification

Tags can be applied to several segments of your infrastructure:

- Instances (physical servers, VMs, cloud instances)
- Individual metrics
- Agent and OnSight heartbeats
- Network device ports
- Users and dashboards

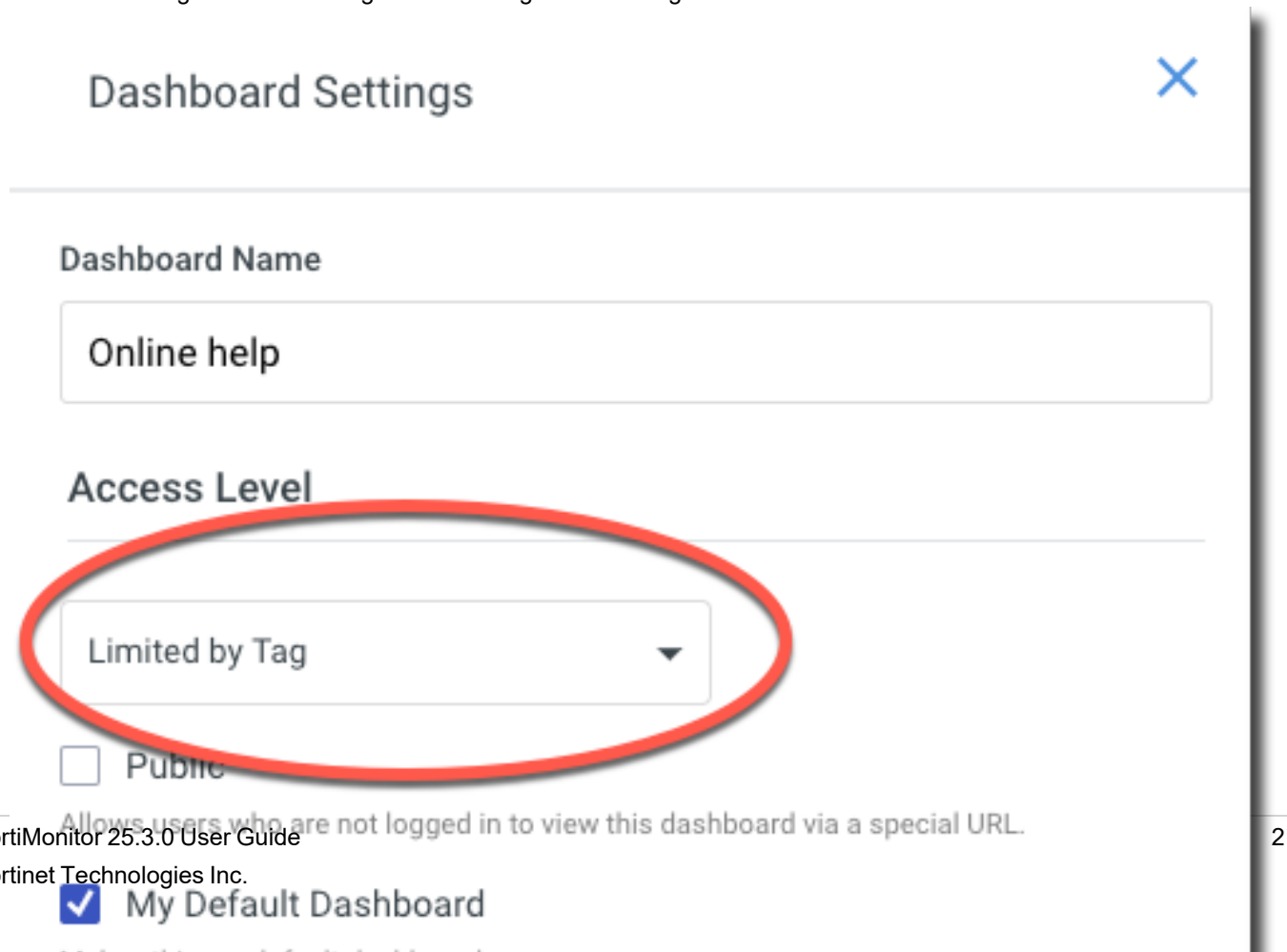
These are detailed in Apply tags.

You can manage and view all of your tags by going to the **Tags** page. For more information, see Manage and view tags.

Using tags

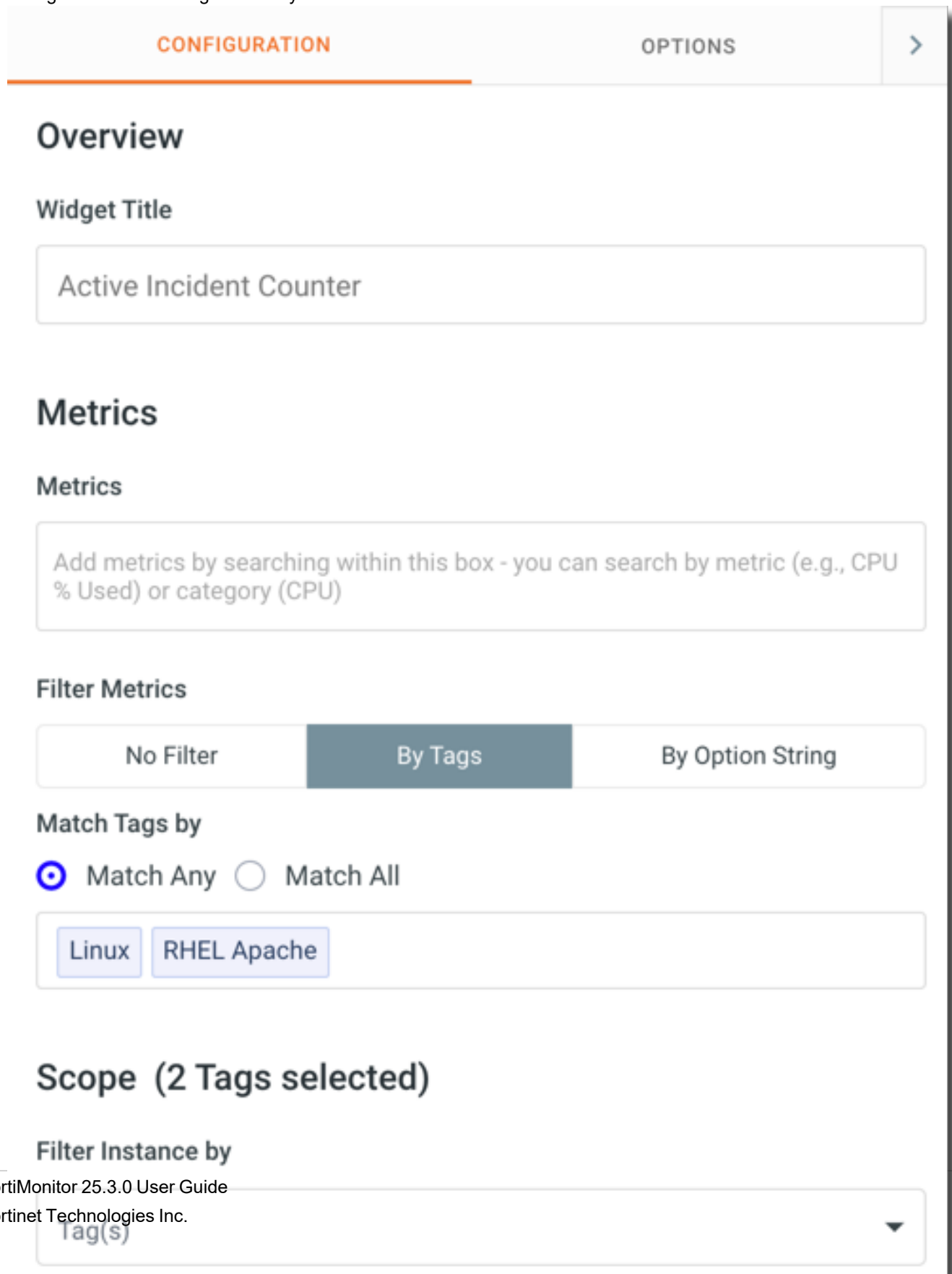
Limit access to dashboards

The **Access Level** options allow you to limit which of your users can see specific dashboards. If you select the **Limited by tag** option, dashboard access will be limited by which tags are set on this dashboard and tags set on a user. The tag filter uses OR logic when matching dashboard tags and user tags.



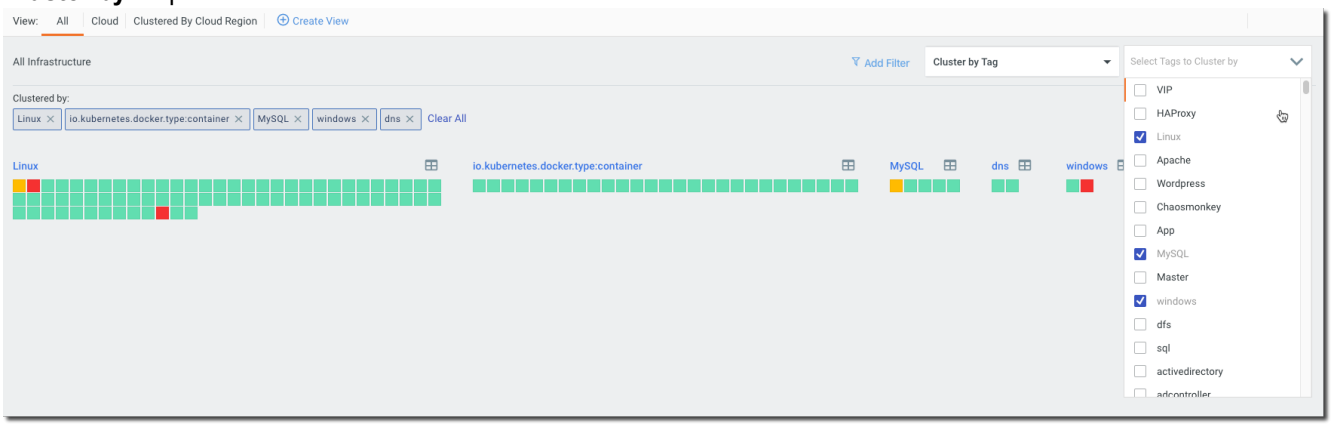
Dashboard widget filtering

Every widget provides options for dynamically displaying the appropriate infrastructure and metrics by tag, rather than hard-coded selections. This allows your infrastructure to expand and contract as needed without requiring constant changes on what is being shown in your live dashboards.



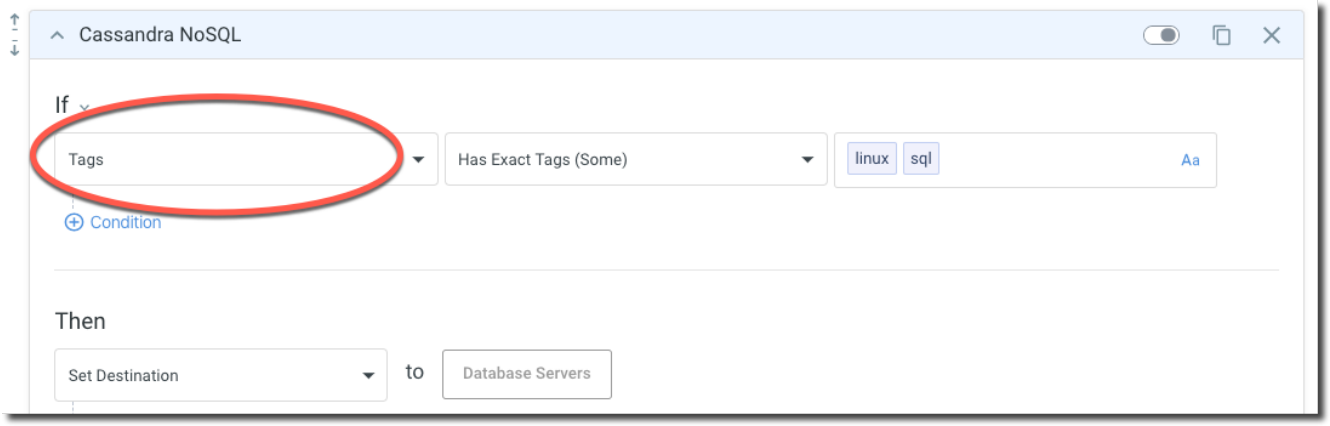
Infrastructure Map – Cluster by tag

Clustering by tag groups instances by the tags applied to each instance. Each tag you select from the **Select Tags to Cluster by** dropdown menu will become its own cluster.



Monitoring policies

Using tags as a filterable dimension in Monitoring policies allows you to set tags as a condition in a ruleset. For example, based on the following screenshot, if an instance added to Panopta has the **Linux** and **SQL** tags, the instance will be added to the **Database Servers** group. For more details on how to use tags in monitoring policies, see Tag matching logic.



You can also use monitoring policies to apply tags to an instance.

Network device ports

This is helpful when you want to isolate and modify ports based on their function. For example, filtering ports with an HTTP tag will allow you to see only the ports associated with the tag. To tag network device ports, see Tag a network device port.

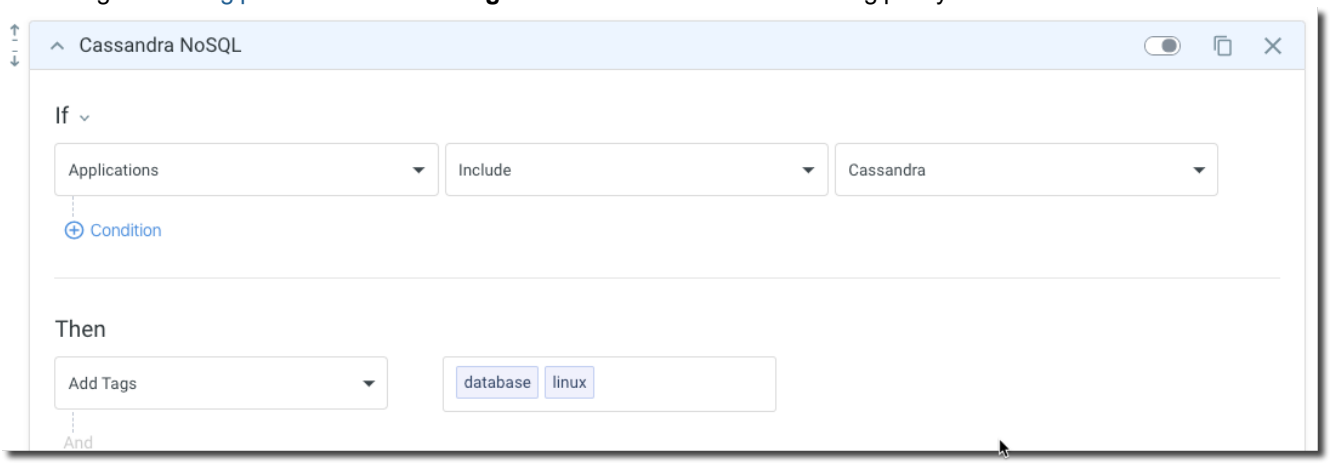
Apply tags

Apply tags to an instance

Instance tagging is helpful for organization and instance selection - starting maintenance windows, running reports, or generating dashboards. Heartbeat tags can also be added to Agent and OnSight-based instances.

There are several ways to tag an instance:

- During Agent installation – Using the Agent oneliner during installation, the **tags** parameter can be used to apply tags to an instance.
  ```
  curl https://packages.panopta.com/install/agent_install.sh | bash /dev/stdin --customer-key 13232-32342-2323 --tags linux,database,prod
  ```
- Using monitoring policies – The **Add Tags** action can be used in monitoring policy rulesets.



- Using the manifest file – Using the tags field will allow you to add tags to your server. This can be set to any value but would need to be comma-separated to indicate different tags.
  ```
  [agent]
  customer_key = afsdyngoaeppmfqefa
  server_key = ashe-pokf-bfhb-eabn
  tags = tag1, tag2, tag3
  ```

Apply tags to metrics

This is helpful when you want to pull reports or build dashboards based on checks with specific tags.

User dashboard tags

User and dashboard tags are used to restrict users from accessing certain dashboards. When creating a dashboard, use the **Limit by tag** access level. Dashboard access will be limited by which tags are set on this dashboard and tags set on a user. The tag filter uses **OR** logic when matching dashboard tags and user tags.

To create dashboards, see Build live Dashboards.

To add users, see Add users to your account.

Templates

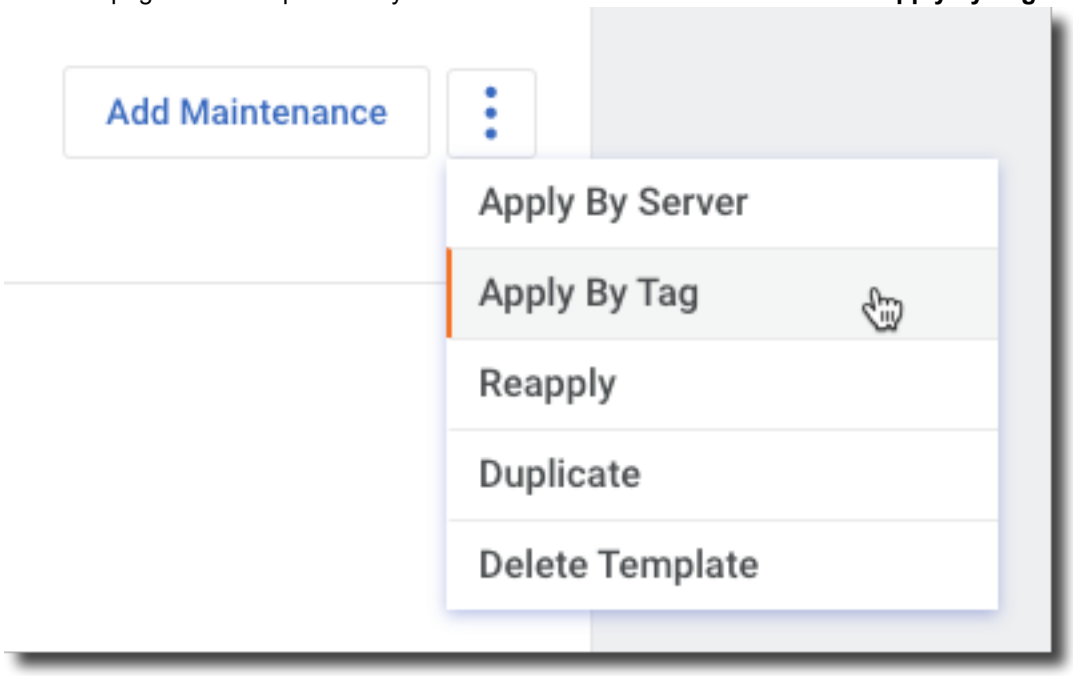
Add tags to templates

Tags added to a template will automatically be applied to instances that use the template. To add tags to a template, configure the template by performing the steps detailed in Configure a new or existing template.

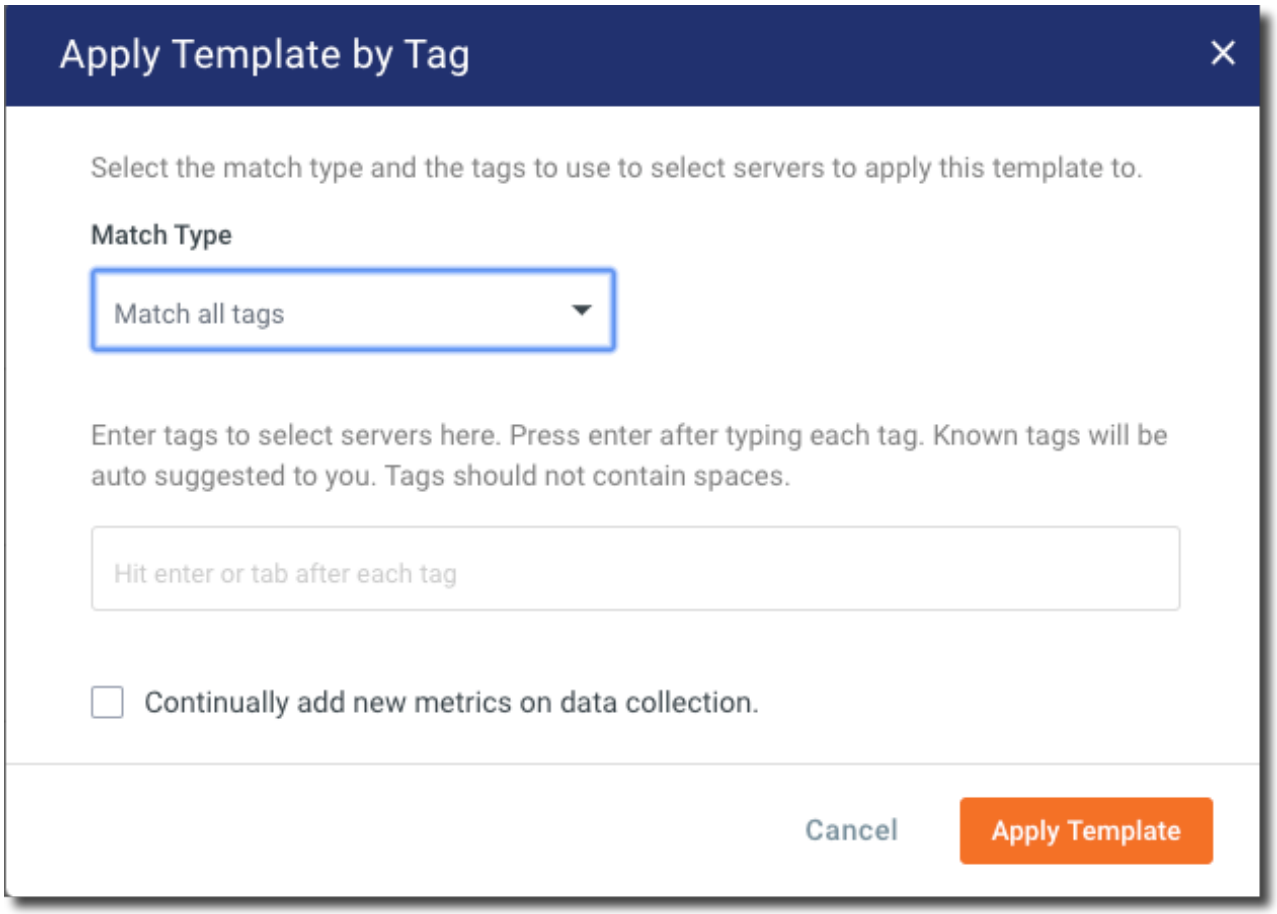
Apply templates using tags

You can apply a template to multiple instances by using the **Apply by Tag** option from a template's details page. To do so, perform the following steps:

1. Go to the page of the template that you want to use and click the **3-dot menu** > **Apply by Tag**.



A modal will open where you can select a **Match Type** and enter tags that you want to use.



2. Select a **Match Type**.
   a. **Match all tags** – The template will be applied to instances that match ALL of the tags entered.
   b. **Match any tag** – The template will be applied to instances that match at least one of the tags entered.
3. Enabling the option **Continually add new metrics on data collection** will enable this option for all of the instances where the template will be applied.
4. Select **Apply Template**.

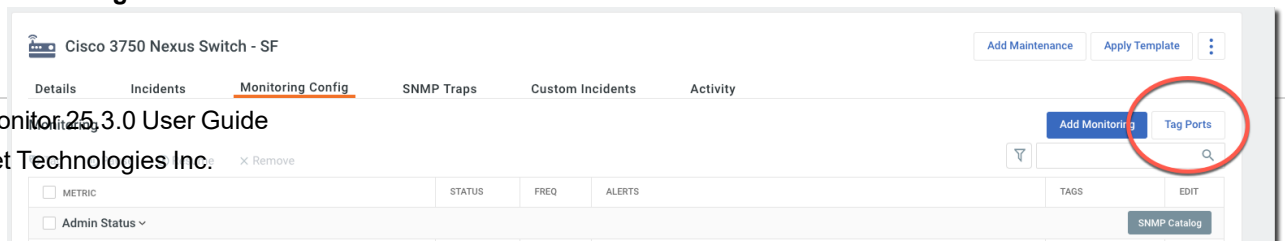
Apply tags to network device ports

Tagging individual ports can provide a dynamic way to query or filter ports in a network device. For example, you can add the **HTTP** tag to all of the ports dedicated to HTTP to make the ports searchable from the **Monitoring Config** tab. When you use the tag (in this case, **HTTP**) to query the ports, only the ports tagged with **HTTP** will be listed. It's also convenient for limiting and grouping series in dashboards. To tag network device ports, see Tag a network device port.

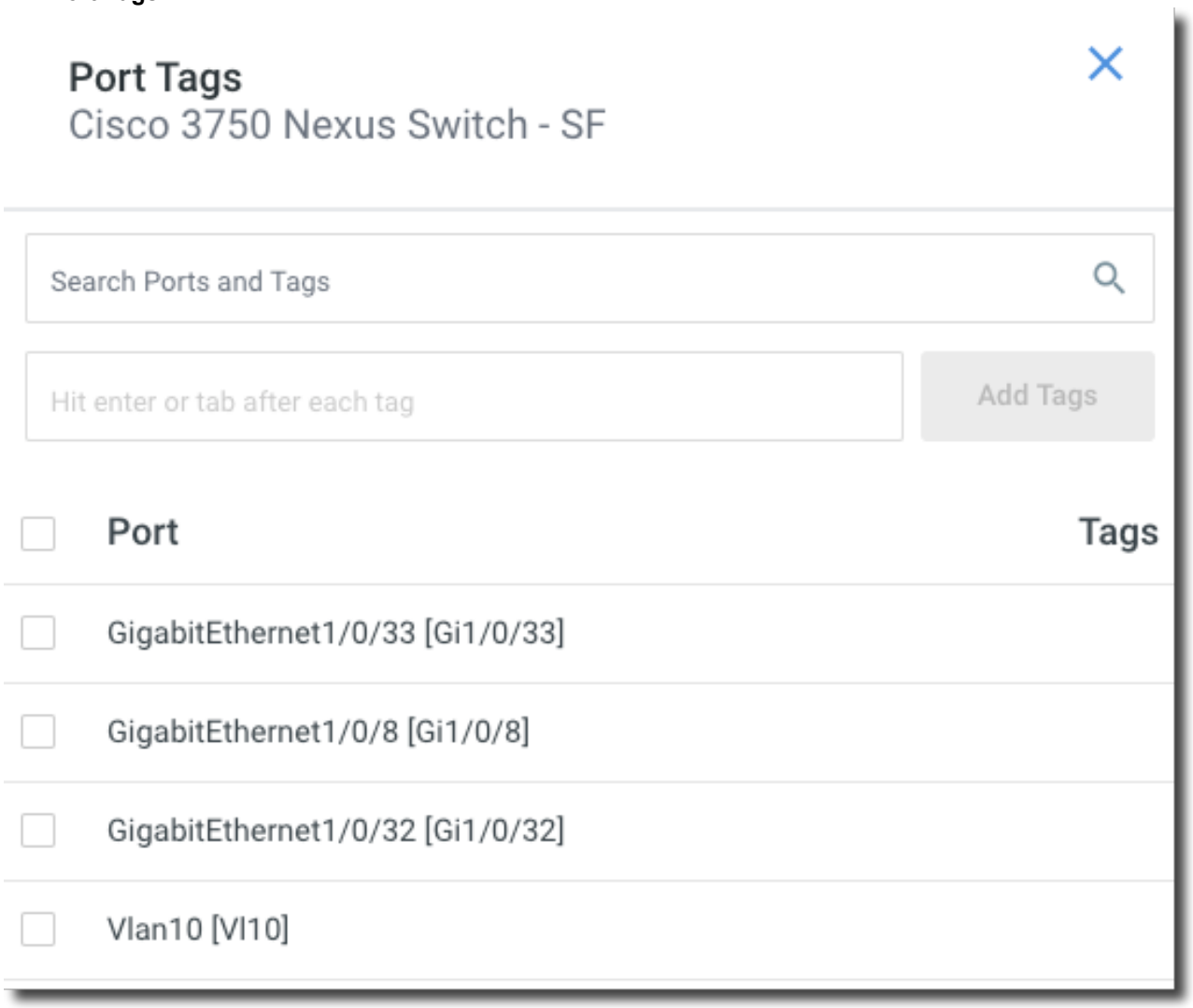
Tag a network device port

To tag a network device port, perform the following steps:

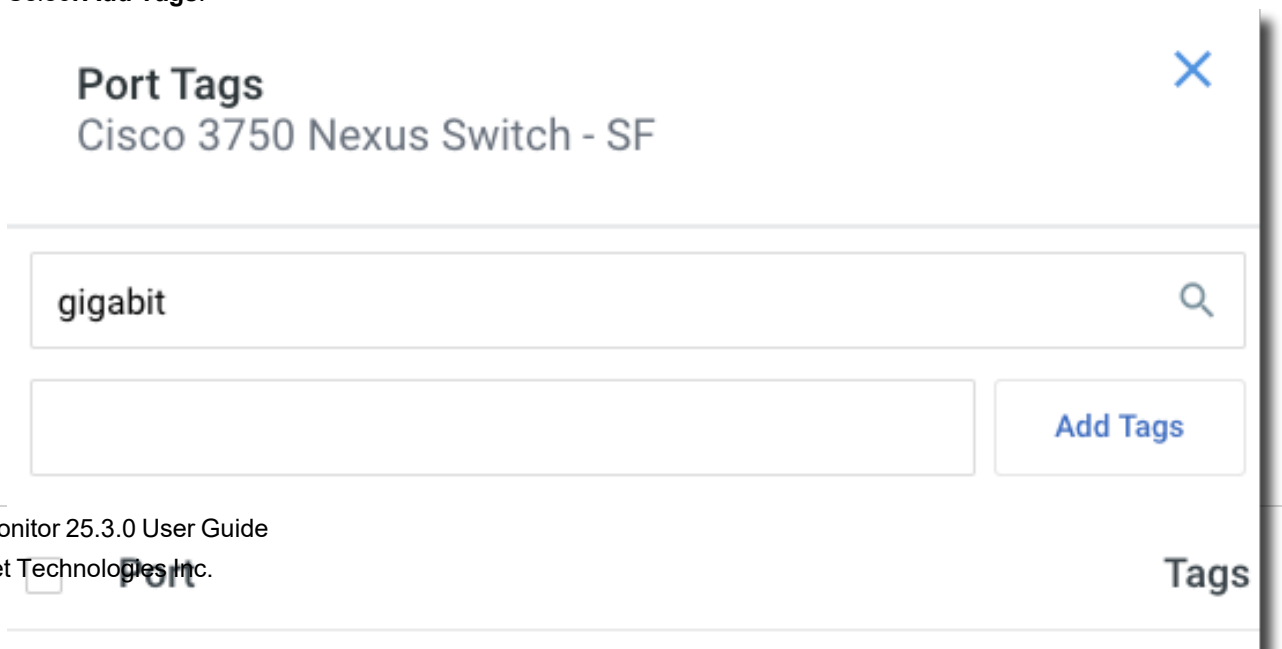
1. Go to a network device's Instance Details page.
2. Select the **Monitoring Config** tab.
   Select **Tag Ports**.



The **Port Tags** drawer will slide out.



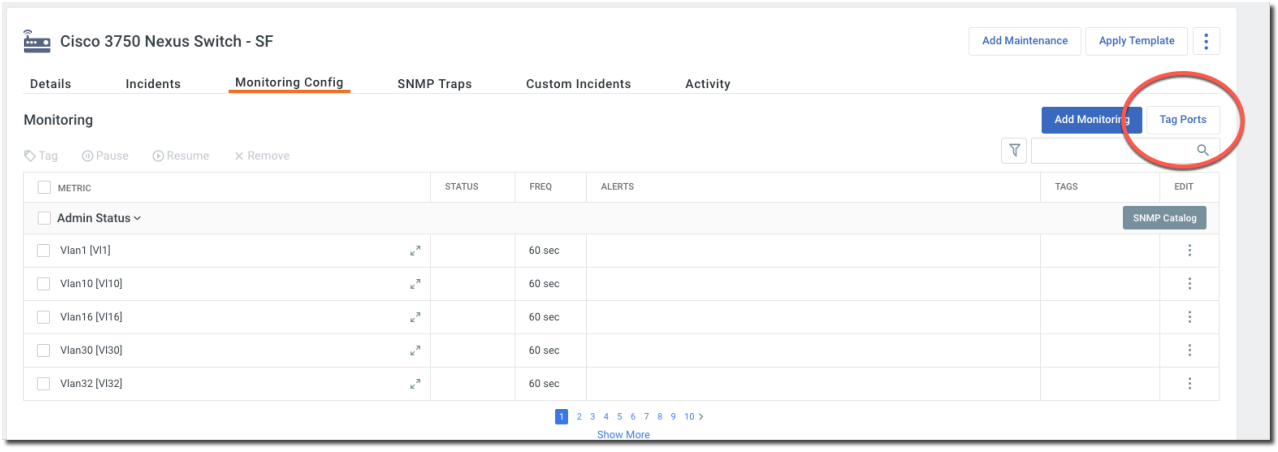
4. Click the checkbox or checkboxes of the port(s) that you want to tag. You can also use the search field to narrow down the list of ports.
5. Enter a tag then press **Tab** or **Enter** after each tag.
   Select **Add Tags**.



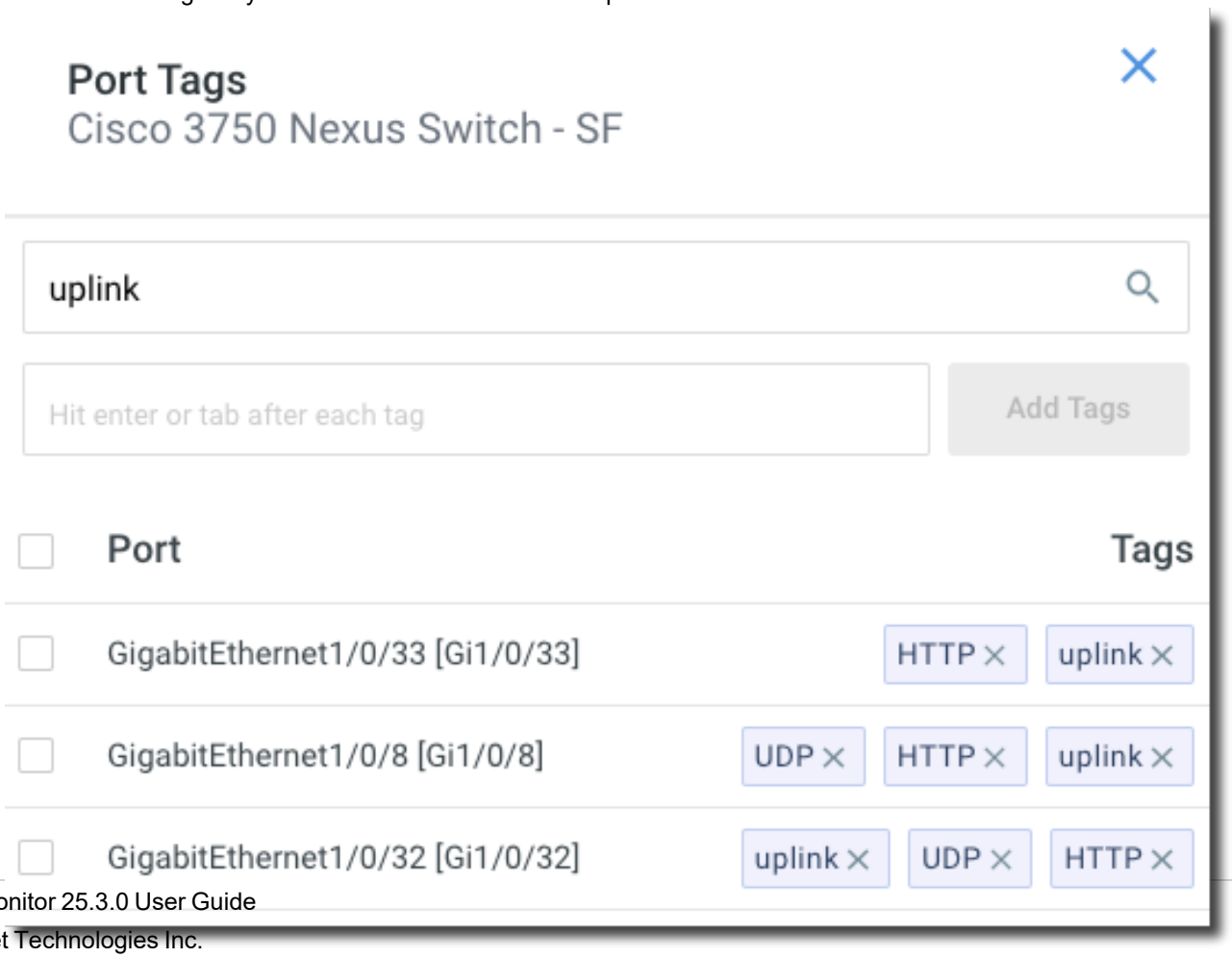
7. Select **Save**. The tag will now be applied to the selected ports.

Remove tags from ports

To remove the tags applied the network device ports, perform the following steps:

1. Go to a network device's Instance Details page.
2. Select the **Monitoring Config** tab.
   Select **Tag Ports**.
3. 

   The **Port Tags** drawer will slide out.
   Search for the tag that you want to disassociate from the port.



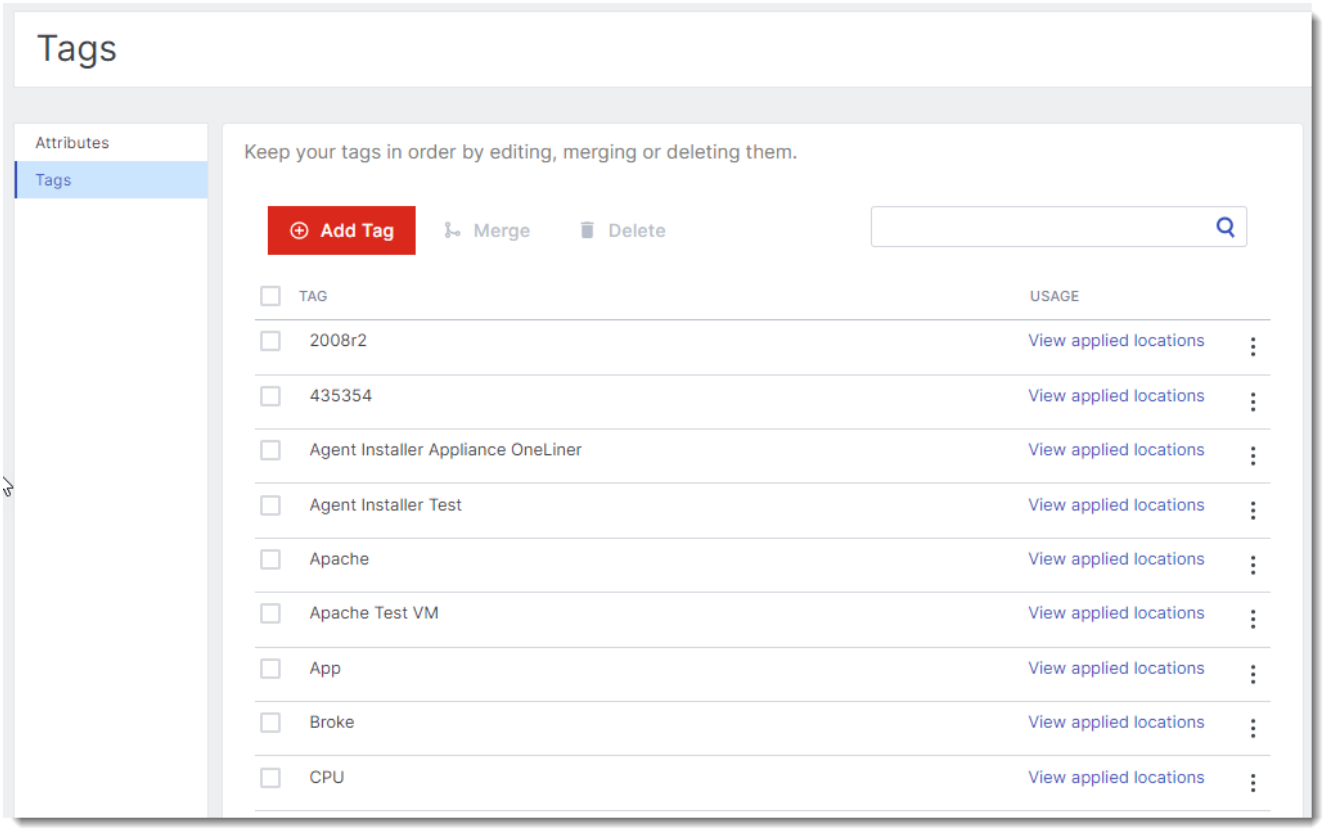
4.

5. Click **X** on the tag that you want to remove.
6. Select **Save**. The tag will now be removed from the port.

Manage and view tags

Tags page

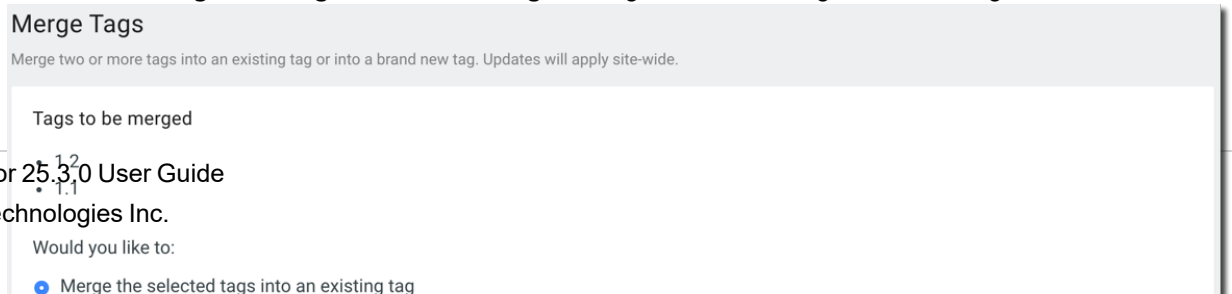
You can centrally view and manage all tags by selecting **Monitoring** > **Attributes & Tags** from the navigation menu. From there, you can merge and delete tags, which will update all the places they have been used. Selecting **View applied locations** on each tag will show you where it is applied. From this page, you can add tags by simply clicking **Add Tag**. You can also rename and delete tags and edit a tag's name by clicking the **3-dot menu** of the tag.



Merge tags

To merge tags, perform the following steps:

1. Select the tags that you want to merge.
2. Click **Merge**.
3. Select how you want to merge the tags
   a. **Merge the selected tags into an existing tag** – Merge two or more tags into an existing tag.
   Create a new tag and merge the selected tags – Merge two or more tags into a new tag.



4. Click **Merge Tags**. The places where the tags have been used will be updated.

Tag Audit report

The Tag Audit Report allows you to see which tags are in use, and to whom and what they are applied.

Attributes

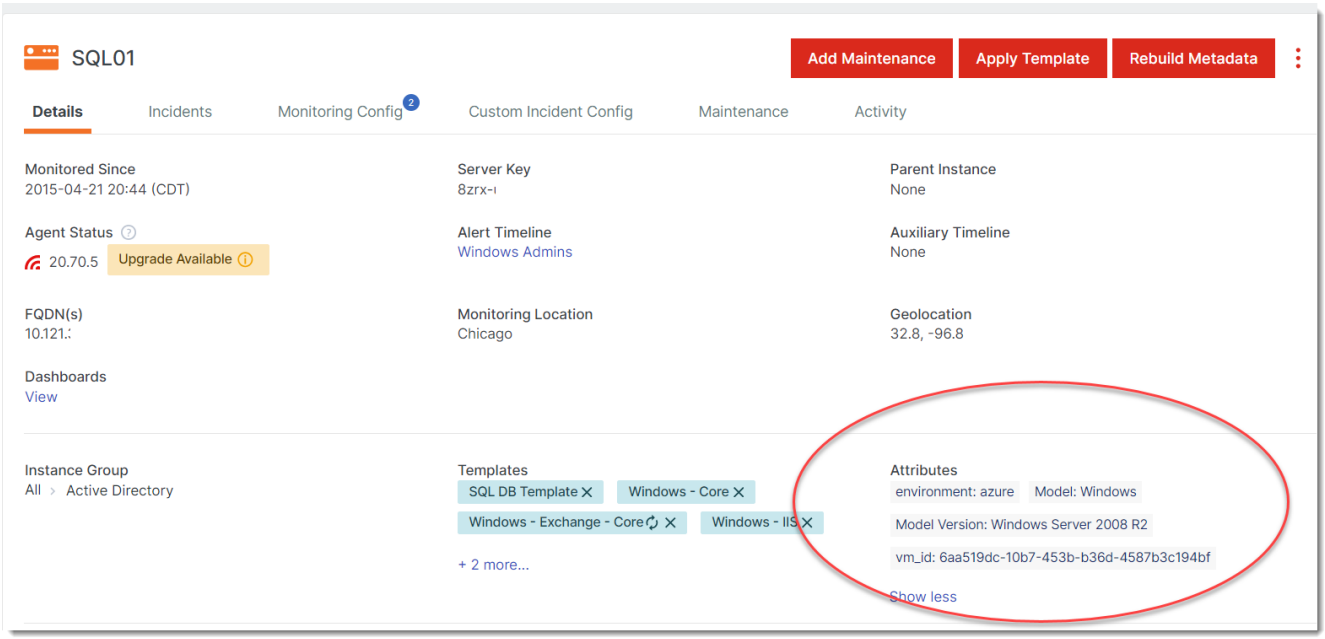
Attributes are values associated with instances that can be leveraged to filter results when needed.

While **Attributes** are similar to **Tags** there are some key differences.

Tags are standalone values while **Attributes** consist of a key-value pair that provides additional context for the stored value.

Each attribute key can also have multiple associated values.

For example, a **Tag** value could be 'Windows' while an **Attribute** key-value pair must consist of the key and at least one paired value example a key of **Operating System** and the value **Windows** as well as the value **Linu'**. Thus allowing to filter for results where the **Attribute key** of **Operating System** has a value of **Windows**.



Use Cases

Attributes can be leveraged for the following tasks:

1. Monitoring Policy Framework automation.
2. Dashboard level scope filtering.
3. Filtering Instance Group instance results.
4. Filtering Incident Hub incident results.

Creation

Attributes can be created from the **Instance Details** page.

Attributes can also be included as part of a manifest file during agent installation.

The **Monitoring Policy Framework** can also be leveraged to assign and utilize Attributes.

Lastly, **Attributes** can also be automatically created for AWS or Azure cloud integrations.

The cloud integration setup has the option to import AWS or Azure tags as FortiMonitor tags storing only the value portion or the key-value pair as a single string or import them as a Attribute key-value pair.

Examples

Below are some example use cases for Attributes:

Environment

- Dev
- Stage
- Prod

Operating System

- Windows Server 2019
- Windows Server 2022
- Ubuntu 22.04 LTS
- Ubuntu 20.04 LTS
- Ubuntu 18.04.6 LTS

Location

- Indianapolis
- Chicago
- Columbus
- AWS
- Azure

Status

- Primary
- Backup
- Standby

Bulk Upload

The **Bulk Upload** feature allows you to add tags and attributes to a large number of instances during or after the onboarding process. To do so, a template in XLSX format is available for download. You can use this template to assign tags, attributes, or both to your instances. After completing the template, you can upload the file directly to FortiMonitor.

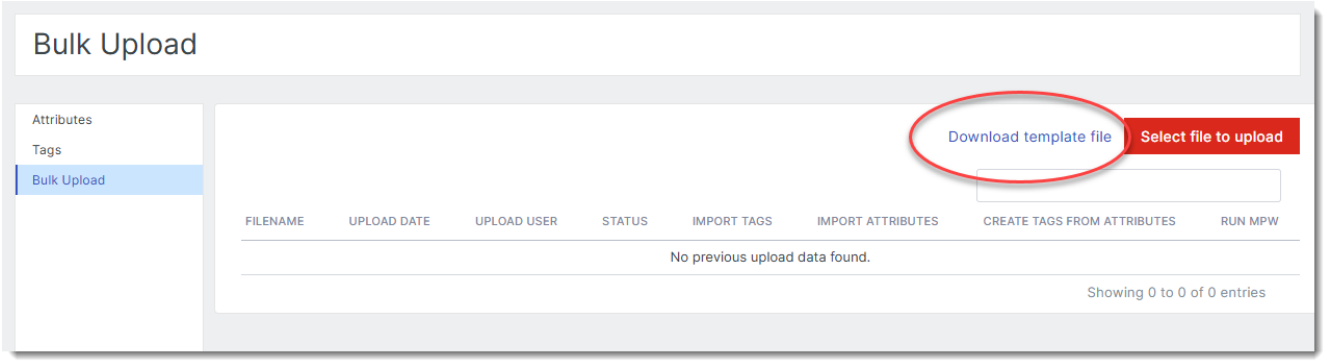
Once uploaded, FortiMonitor will automatically validate the file. If the file is valid and contains no errors, the import process will be performed in the background. Only users with the appropriate permissions to view and edit tags and attributes are able to use this feature.

Note: When using the template, tags and attributes used in the template must already exist in your account. The template does not have the ability to create new ones. Additionally, the template will not generate duplicate tags or attributes.

Template

A template file in **XLSX** format is available for download. To download the file:

1. Go to **Monitoring** > **Attributes & Tags** > **Bulk Upload**.
2. Click **Download template file**.



Using the template

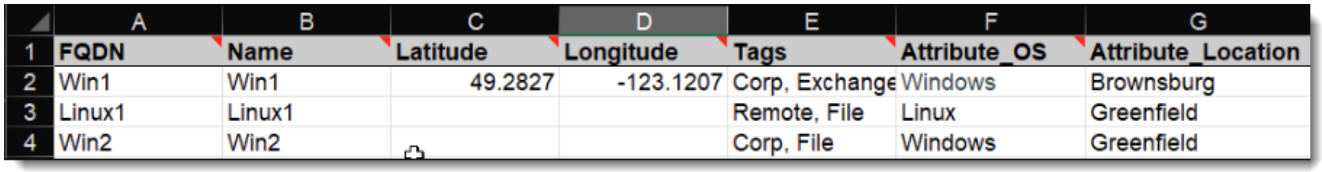
The following table describes the fields that are available in the template:

Field	Description
FQDN	Enter the FQDN or IP address of the instance. The instance must match the FQDN in your account. If it does not, this row will be skipped. Ensure that only one entry is filled out per instance in the template to avoid errors.

Field	Description
Name	(Optional) Enter a name for the instance.
Latitude and Longitude	(Optional) Enter the coordinates of the instance. For example: 41.8781 87.6298 If the location coordinates are valid, the geo-location attributes will be added to the instance.
Tags	Enter the tags to assign to the instance. To assign multiple tags, see the following section.
Attribute Name 1	Attribute key-value pair. For more information, see Attributes.
Attribute Name 2	Add additional columns for attributes to the right of the column as needed. Populate the top cell of the column with the attribute key and the cells below with the attribute value for each instance.

Assigning multiple tags

To assign multiple tags to an instance, add the tags to the row of the instance in comma-delimited format. For example:



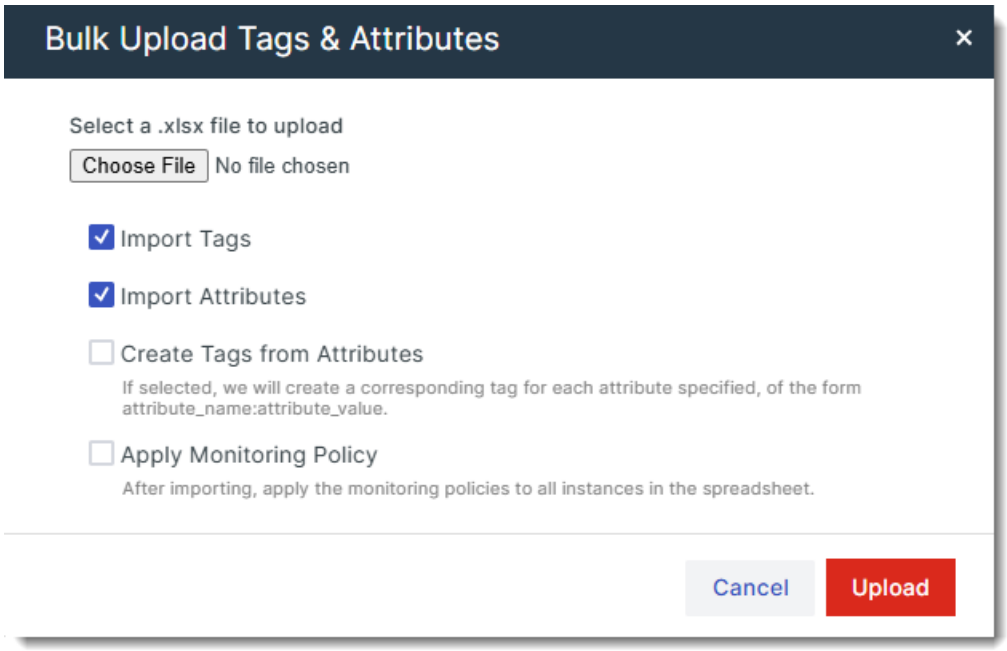
	A	B	C	D	E	F	G
1	FQDN	Name	Latitude	Longitude	Tags	Attribute_OS	Attribute_Location
2	Win1	Win1	49.2827	-123.1207	Corp, Exchange	Windows	Brownsburg
3	Linux1	Linux1			Remote, File	Linux	Greenfield
4	Win2	Win2			Corp, File	Windows	Greenfield

Creating multiple rows with the same instance will result to the file being invalid when uploaded.

Upload the template

To upload the template:

1. Click **Select file to upload**.
2. Use the modal that appears to choose the appropriate options for this template.
3. Click **Upload**.



Option	Description
Choose file	Select the template file to be uploaded.
Import Tags	Enable this option to import tags.
Import Attributes	Enable this option to import attributes.
Create Tags from Attributes	If enabled, a corresponding tag will be created for each specified attribute, following the format **attribute_name:attribute_value**.
Apply Monitoring Policy	If enabled, the monitoring policy is applied to all instances in the template.

Monitoring locations

The FortiMonitor Monitoring System is based on a global network of monitoring nodes designed to track the availability of your servers 24 hours a day. This page is updated regularly as new nodes are added or when IP addresses change.

All of our servers are hosted with trusted partners using a fully state-of-the-art datacenter infrastructure. Our entire monitoring infrastructure is designed to withstand the failure of any component. If one monitoring node is inaccessible, checks begin immediately from an alternate node and continue until the original node returns to service so that your servers never go unmonitored.

IP addresses are listed for each monitoring node for customers that want to either restrict access to only our nodes or to separate out monitoring traffic from customer traffic in weblogs, etc. Enable the option Panopta Monitoring Network Updates in your Email preferences to subscribe to our monitoring network announcement list.

Lastly, if you are scripting firewall rules, you can hit http://bit.ly/pan-ips to get a plaintext version of our latest IP addresses.

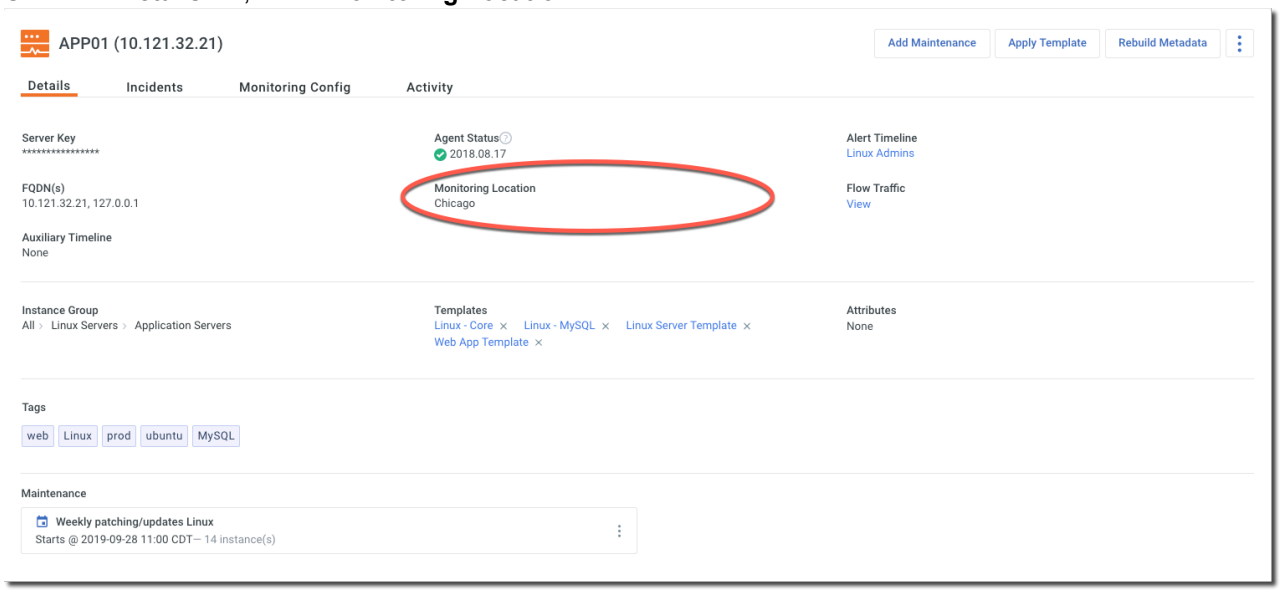
You can see a map and a list of our IP addresses here.

Specifying confirmation check locations

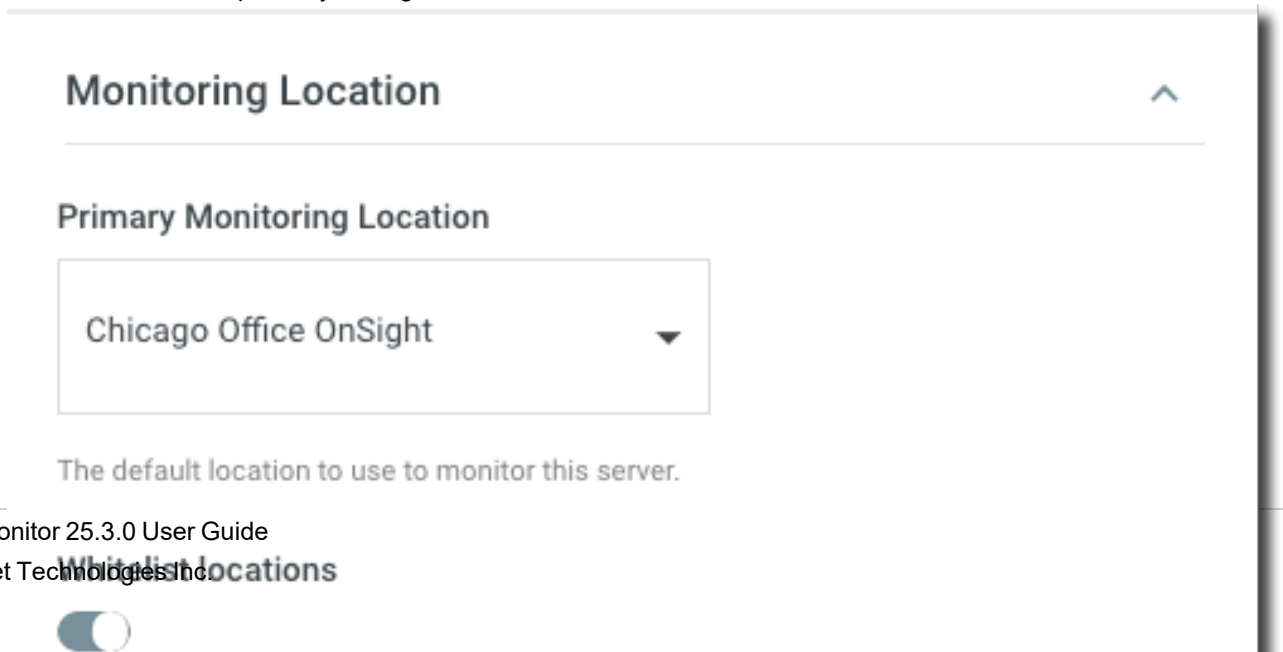
Whenever a service has an outage, we perform several test checks from different locations to ensure that it is not a false positive. You can choose which monitoring locations you would like to perform these confirmation checks from through a feature we call whitelist. This feature can be very useful for a number of reasons. Say, for example, you would only like your confirmation checks to come from monitoring nodes very close to your primary monitoring location, which would allow you to keep track of local network outages. Additionally, If you are explicitly opening firewall rules for a service to be checked, this allows you to keep that list controlled and manageable despite how many new nodes we add.

To create a whitelist, perform the following steps:

1. Go to the server's **Instance Details** page.
   Under the **Details** tab, select **Monitoring Location**.



2. The **Monitoring Configuration** drawers slides out.
   Enable **Whitelist locations**.
   Your confirmation node should be set to default. From here you can select to use either a whitelist. A whitelist will let you make a list of all of the monitoring locations you would like to include. Deselecting a location will exclude the location. When you enable **Whitelist locations**, you will see a list of all of our monitoring nodes to either include or exclude. You will need to leave at least seven monitoring nodes to run these confirmations checks, which is the minimum number required by our algorithms.



4. Once you have selected all of the monitoring nodes you would like, click **Save Changes** to save this configuration. From now on all confirmation checks for this server will be restricted according to your configuration.

Global monitoring network

Americas

Location	IPv4 Addresses	IPv6 Addresses
Atlanta	• 38.121.43.199 • 198.74.52.10 • 45.32.222.5	• 2600:3c02::f03c:91ff:fe96:42d3 • 2001:19f0:5401:24:5400:ff:fe33:5a21 • 2604:86c0:7000:2::a
Chicago	• 146.71.78.119 • 45.63.67.141 • 185.33.85.38 • 162.251.61.122 • 64.187.238.242	• 2604:bc0:1::4:103f • 2604:86c0:0:8::a • 2a01:8640:4:1::bcfe:b5f4 • 2001:19f0:5c00:9415:5400:ff:fe20:6f0 • fe80::225:90ff:fea8:612e
Dallas	• 96.126.124.184 • 108.161.134.98 • 38.68.135.103	• 2600:3c00::f03c:91ff:fedf:9d63 • 2604:86c0:4000:3::a • 2606:4300:0:4::10f4
Denver	• 104.219.169.116	• 2606:1880:3:1b::3e2c
Fremont, CA	• 50.116.5.62	• 2600:3c01::f03c:91ff:fe96:a9f6
Kansas City, MO	• 69.30.216.234	• fe80::6ef0:49ff:fea0:51c8
Los Angeles	• 45.32.77.110 • 45.66.249.212 • 162.251.164.116	• 2001:19f0:6001:c4:5400:ff:fe30:10d9 • 2a10:1fc0:6::1952:b46d • 2604:86c0:2000:8::a
San Jose	• 104.238.180.233	• 2001:19f0:ac01:121:5400:ff:fe33:5a38
Santiago	• 37.235.52.158	• 2a03:f80:56:37:235:52:158:1
Sao Paulo	• 54.94.193.233	x
Newark, NJ	• 207.192.69.191	• 2600:3c03::f03c:91ff:fe96:a92d
New York	• 198.211.117.64 • 52.129.44.146	• 2604:a880:400:d0::7:8001
Phoenix	• 162.251.160.214	• 2605:9d00:0:7::2
Seattle	• 23.226.225.34 • 108.61.194.48 • 144.48.107.157	• 2604:180:1::5452:d9e • 2001:19f0:8001:b3a:5400:2ff:fe41:ee74
Toronto	• 159.203.6.127	• 2604:a880:cad:d0::132:1001
Washington, DC	• 192.96.206.34 • 158.106.140.78 • 147.135.10.81	• 2600:4c00:80:141:: • 2604:2dc0:100:151:: • 2604:9a00:2100:af05:3::1

EMEA

Location	IPv4 Addresses	IPv6 Addresses
Amsterdam	• 45.32.234.28 • 198.211.119.192	• 2001:19f0:5001:1dc:5400:ff:fe49:72bb • 2a03:b0c0:0:1010::3c1:b001
Frankfurt	• 138.68.80.87 • 108.61.211.127	• 2a03:b0c0:3:d0::1b8a:2001 • 2001:19f0:6c00:8681:5400:ff:fe05:d934
Johannesburg	• 165.73.248.123	
London	• 54.36.164.191 • 176.58.122.143	• 2001:41d0:800:ebf:: • 2a01:7e00::f03c:91ff:fe70:bd4f
Milan	• 172.232.203.17	• 2600:3c0b::f03c:94ff:fe22:4349
Nairobi	• 102.68.86.48	• 2c0f:f598:6::53f:de0a
Paris	• 107.191.46.31	• 2001:19f0:6800:85e1:5400:ff:fe08:5094
Tel Aviv	• 193.182.144.207	• 2a03:f80:972:193:182:144:207:1

ANZ

Location	IPv4 Addresses	IPv6 Addresses
Melbourne	• 113.52.144.53 • 113.52.144.191	x
Sydney	• 103.25.58.106 • 103.4.16.254	• 2407:500::2:9794:a374 • 2406:d501::de92:8614
Adelaide	• 103.25.56.152	• 2401:1400:1:1201:0:1:9814:9d4

APAC

Location	IPv4 Addresses	IPv6 Addresses
Bangalore	• 139.59.22.193	• 2400:6180:100:d0::18d:c001
Hong Kong	• 158.255.208.184	• 2a03:f80:852:158:255:208:184:1
Jakarta	• 38.68.135.103 • 38.68.135.224 • 103.243.44.88	x
Singapore	• 128.199.70.67 • 103.25.203.234	• 2400:6180:0:d0::3a1:b001 • 2400:c980:0:2:4bf:aeff:fe01:13e7
Tokyo	• 172.105.194.147	• 2400:8900::f03c:91ff:fedf:4015

Change log

July 22, 2024

- Updated the Los Angeles IPv4 and IPv6 addresses.
- Updated the Milan IPv4 and IPv6 addresses.

September 25, 2023

- Removed Moscow node.

May 12, 2022

- Removed Accra node.
- Updated the Fremont IPv4 address.
- Updated the Johannesburg IPv4 address.
- Removed the Johannesburg IPv6 address.

June 18, 2021

- Added chicago04, chicago11, and kc10.

June 10, 2021

- Removed Vienna node. All checks moved to Paris.

Checker IP changes

March 17, 2025 changes

We are making an IP change to our public checker **newark01.monitorengine.com (Newark, NJ)**. The planned time for this update will be **March 17, 2025**. Provided your firewall rules are updated to allow the new IP addresses during this window, you should not notice any changes from these updates. If you do not apply firewall rules to any of your public checks, there is nothing you need to do during this time.

Control Panel Name	Location	Server Name	Old IPv4	New IPv4	Old IPv6	New IPv6
Newark, NJ	Newark, NJ	newark01.monitorengine.com	207.192.69.19	45.33.88.25	n/a	n/a

February 14, 2025 changes

We are making changes to our public checker **lisbon01.monitorengine.com (Lisbon 01)**. Checks monitored by this public probe will be moved to a newly deployed public checker **lisbon02.monitorengine.com (Lisbon 02)**. When the update is complete, you will see no longer see **Lisbon 01** and its checks will be moved to **Lisbon 02**.

The planned time for this update will be **February 14th, 2025**. Provided your firewall rules are updated to allow the new IP addresses during this window, you should not notice any changes from these updates. If you do not apply firewall rules to any of your public checks, there is nothing you need to do during this time.

Control Panel Name	Location	Server Name	Old IPv4	New IPv4	Old IPv6	New IPv6
Lisbon 01	Lisbon, Portugal	lisbon01.monitorengine.com	81.90.54.130	n/a	n/a	n/a
Lisbon 02	Lisbon, Portugal	lisbon02.monitorengine.com	n/a	94.131.10.93	n/a	n/a

February 15, 2024 changes

Control Panel Name	Location	Server Name	Old IPv4	New IPv4	Old IPv6	New IPv6
Adelaide, Australia	Adelaide, Australia	adelaide01.monitorengine.com	103.25.56.152	16.50.12.213	2401:1400:1:1201:0:1:9814:9d4	TBD
Amsterdam, NL 2	Amsterdam, NL	amsterdam02.monitorengine.com	198.211.119.192	157.245.72.251	2a03:b0c0:0:1010::3c1:b001	2a03:b0c0:2:d0::f79:2001
Bangalore	Bangalore	bangalore01.monitorengine.com	139.59.22.193	165.232.186.183	2400:6180:100:d0::18d:c001	2400:6180:100:d0::876:d001
Dallas 2	Dallas	dallas06.monitorengine.com	108.161.134.98	209.58.154.60	2606:4300:0:4::10f4	2606:9880:2100:a007:2::1
Frankfurt 1	Frankfurt, Germany	frankfurt01.monitorengine.com	138.68.80.87	64.226.90.108	2a03:b0c0:3:d0::1b8a:2001	2a03:b0c0:3:d0::149d:3001
Hong Kong	Hong Kong	hongkong01.monitorengine.com	158.255.208.184	16.162.153.177	2a03:f80:852:158:255:208:184:1	TBD
Los Angeles 3	Los Angeles	la02.monitorengine.com	162.251.164.116	172.233.154.229	none	2a01:7e03::f03c:93ff:fe08:b893
Melbourne, Australia	Melbourne, Australia	melbourne03.monitorengine.com	113.52.144.53	139.84.192.227	none	2401:c080:2000:18cc::/64
Nairobi	Nairobi	nairobi01.monitoren	102.68.86	102.68.86	2c0f:f598:6::53f:de	2c0f:f598:6::b339:91c8

Control Panel Name	Location	Server Name	Old IPv4	New IPv4	Old IPv6	New IPv6
		gine.com	.48	.95	0a	
New York	New York	nyc01.monitorengine.com	198.211.117.64	159.65.236.124	2604:a880:400:d0::7:8001	2604:a880:400:d0::209d:b001
Santiago, Chile	Santiago, Chile	santiago01.monitorengine.com	37.235.52.158	64.176.11.33	2a03:f80:56:37:235:52:158:1	2001:19f0:c800:2ffd:5400:04ff:fea6:4047
Seattle	Seattle	seattle06.monitorengine.com	23.226.225.34	172.232.171.116	2604:180:1::5452:d9e	2600:3c0a::f03c:93ff:fe08:0ba7
Seattle 3	Seattle	seattle03.monitorengine.com	144.48.107.157	108.62.194.170	none	2606:9880:2100:a007:2::1
Singapore 1	Singapore, Indonesia	singapore01.monitorengine.com	128.199.70.67	206.189.89.152	2400:6180:0:d0::3a1:b001	2400:6180:0:d0::81a:9001
Sydney 04	Sydney	sydney04.monitorengine.com	103.4.16.254	45.76.122.249	2407:500::2:9794:a374	2401:c080:1800:47a1:5400:04ff:fea6:3fbf
Sydney, Australia 2	Sydney, Australia 2	sydney02.monitorengine.com	103.25.58.106	172.105.162.128	2406:d501::de92:8614	2400:8907::f03c:93ff:fe08:0bf2
Telaviv 1	Telaviv, Israel	telaviv01.monitorengine.com	193.182.144.207	51.17.146.118	2a03:f80:972:193:182:144:207:1	TBD
Toronto	Toronto	toronto01.monitorengine.com	159.203.6.127	159.203.8.100	2604:a880:cad:d0::132:1001	2604:a880:cad:d0::d6b:5001
Washington, DC	Washington, DC	wdc02.monitorengine.com	158.106.140.78	172.234.43.102	2600:4c00:80:141::	2600:3c05::f03c:93ff:fe08:0bcb

Advanced monitoring features

Compound Services

Allow you to connect servers by relationship or cluster so of any % or related services become unavailable, you will only receive one alert. This is also helpful in the case of servers in a cluster, you may not need to know about if one server in a 10 server cluster goes down, but you're going to start to worry when 5 of those 10 become unavailable. To learn how to configure a compound service, click here.

Server Templates

Give you the ability to bulk add checks to similar servers and bulk update should the need arise. You can apply multiple templates to one server, for example, you may have a set template for network checks and another template for monitoring server metrics. You can dictate thresholds, alert schedules, and tags with templates. To learn how to set a server template, read Templates.

Forcing DNS Updates

Should you update your server's IP addresses, you'll want to flush the DNS we have cached so you don't get any failing checks. Read here to learn how to Force DNS updates before your TTL expires.

SNMP

We give you the ability to monitor your network devices via SNMP, read here to learn how to set up SNMP monitoring once enabled.

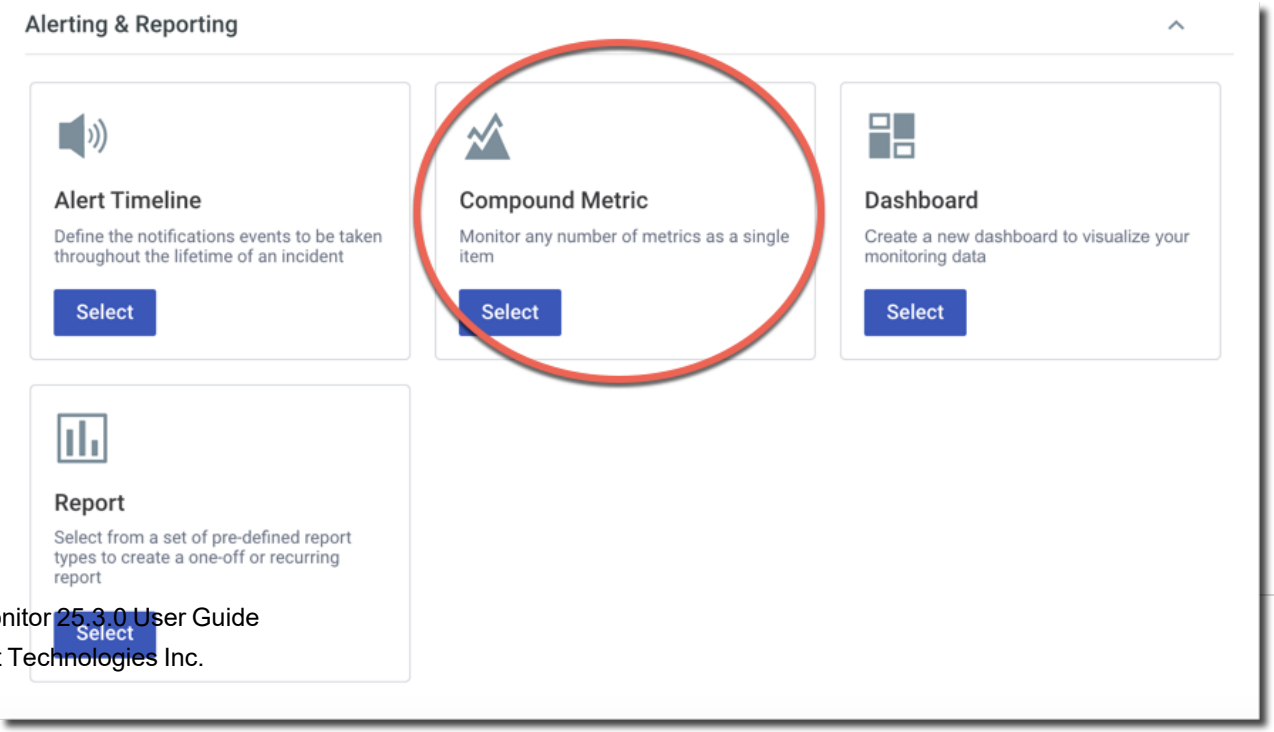
Compound metrics

Often, such as with clusters and HA configurations, you're not concerned with the degradation of a single instance or metric. You may be concerned, however, when a given portion of the cluster begins to experience issues. That's where **compound metrics** come in handy. They allow you to create incidents when a certain percentage of a collection of metric thresholds is experiencing issues. Note that metrics must have configured thresholds in order to be monitored by a compound metric.

Configuration

To create a compound metric, perform the following steps:

1. From the navigation menu, click **Add**. The **Infrastructure and Resource Catalog** will be displayed. Locate the **Compound Metric** card then click **Select**.



2.

3. Click **+Add** at the top of the control panel to open the Infrastructure Catalog. From there, select **Compound Metric**.

   You will be redirected to the top-level-group page, where the **Create Compound Metric** pane will open.

   In the **General** section of the slide-out pane, you can name the compound metric, select which instance group it will reside in, and add tags.

   In the **Services, Thresholds, and Alerts** section, you will define what the compound metric is actually comprised of.

Option	Description	Required
Metric Selection Type	Method for selecting metric that comprise the Compound Metric. **Metric Tag - Any**: use metrics which have at least one of the provided tags to comprise the Compound Metric **Metric Tag - All**: use metrics which have all of the provided tags to comprise the Compound Metric **By Metric**: select metrics individually that will comprise the Compound Metric	Required
Metrics	**Metric Tags**: If one of the two tag-based options was selected in the Metric Selection Type, enter 1-to-many metric tags **Metrics**: if By Metric was selected, a tree will be present to select the individual metric thresholds to comprise the Compound Metric	Required
Threshold %	The % of metric thresholds that are required to be crossed before creating an incident. For instance, if all of the selected metric thresholds must be crossed, you'd enter 100.	Required
Alert Timeline	The Alert Timeline to use when an incident is created.	Optional
Auxiliary Alert Timeline	The Auxiliary Alert Timeline to use when an incident is created.	Optional
Severity	The severity which the compound metric incident is assigned when an incident is created	Optional

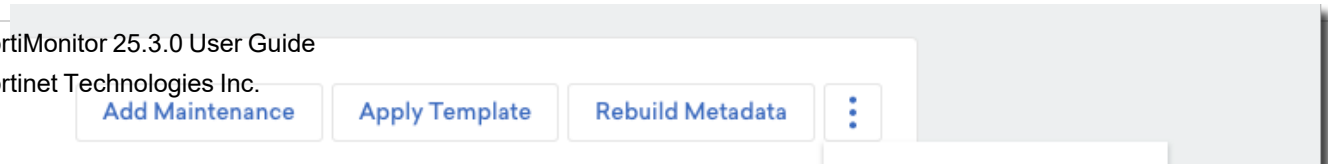
   Compound metrics support the following:

   - External network checks (HTTP, HTTPs, Ping, etc)
   - Agent-based metrics (CPU, Disk, Apache, MySQL, etc)
   - SNMP
   - Cloud metrics

Force a DNS update after an IP address change

Changing an IP address can be a delicate process and we want to make it as easy for you as possible. The last thing you want if for your server to have a false outage for the duration of your TTL because the cached IP address is no longer accurate. Fortunately, we have you covered.

You will need to go to that server's **Instance Details** page on your FortiMonitor account. Select **Flush DNS** from the menu as seen below:

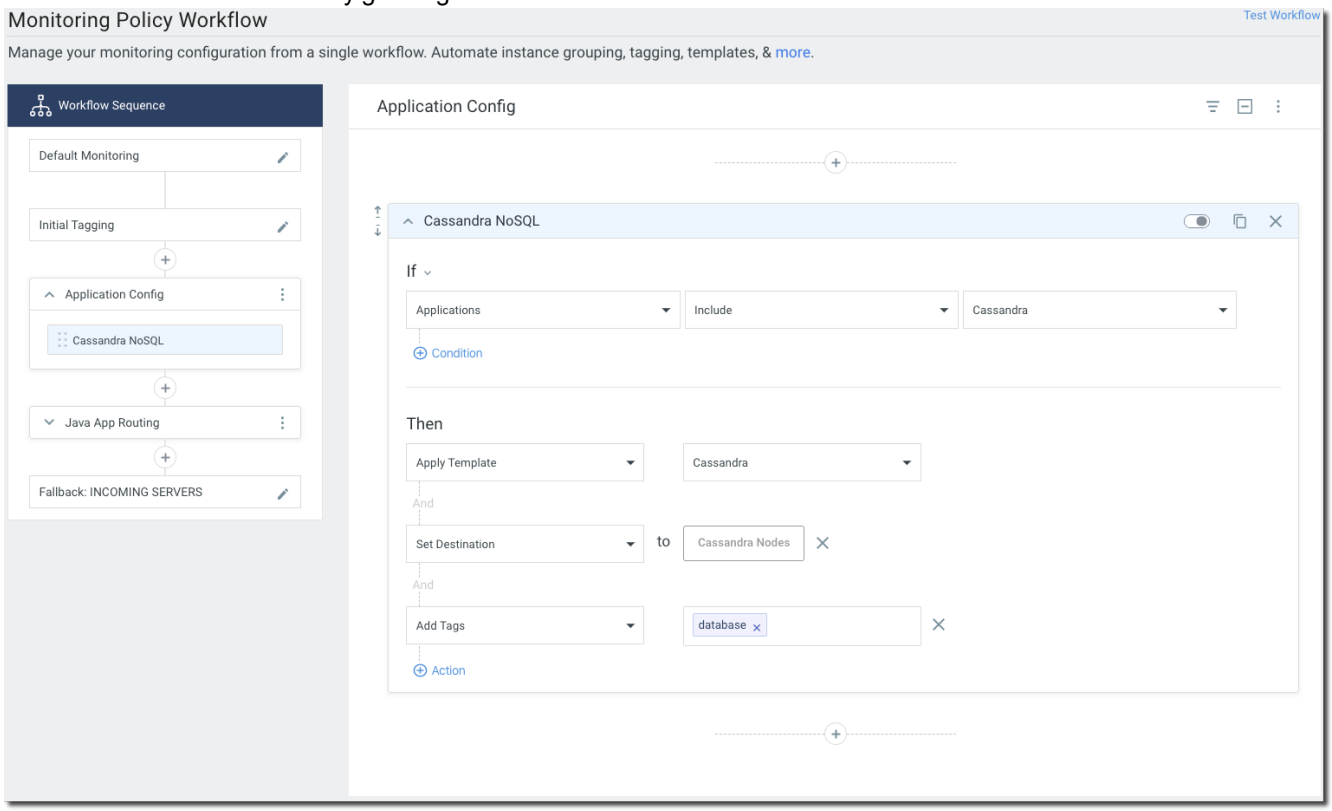


This will flush the local DNS resolver's cache on our monitoring nodes, forcing our servers to contact the authoritative DNS and cache the correct IP address.

Monitoring policies

Managing monitoring configuration at scale is often onerous, especially in diverse environments with many types of applications and technologies. To handle this, managing monitoring configuration as part of the build and release process is often viewed as the end-goal. However, many companies are far from ready for this, for various reasons. Thankfully, your Monitoring Policy enables you to manage your monitoring configuration from a central location using a workflow-like interface.

A Monitoring Policy is a series of rules that are applied to your instances as the instances are added to FortiMonitor. They usually include a condition - e.g. if Apache is on the instance - and an outcome - apply a template, apply a tag, etc. The policy conditions can be either a collection of **OR** or **AND** predicates, forming an **IF...THEN** statement. The conditions utilize our continually growing list of dimensions.



The rulesets are executed sequentially and will override any settings that were previously set. For example, if you set the instance group for an instance and then in the next ruleset it to a different group, the latter will be utilized.

Default rulesets

Out of the box, we supply default rulesets which help to enrich your instances. As the feature matures, more default rulesets will be introduced.

Default monitoring

For each application identified in an instance, a monitoring template will be applied so that metric collection will begin immediately. Note that most applications require at least some level of configuration (such as credentials) to enable monitoring. We'll still apply the appropriate template but the metric collection can not begin in its entirety until the configuration is completed. See our Application plugins section for more information.

Initial tagging

Based on the instance-type (cloud, agent, etc), we'll apply several tags based on the characteristics of the instance - cloud region, cloud service, OS, etc. These are helpful when building dashboards, creating maintenance periods, running reporting, and more.

Fallback location

If an instance group is not set via a custom ruleset, it will be placed in the group selected in the fallback location policy. While acting as the fallback group, a group cannot be deleted - a different fallback will need to be selected first.

Custom rulesets

To get started, click the + icon from the workflow sidebar. From there, you'll enter the ruleset builder where you can add any number of rules.

Always rules

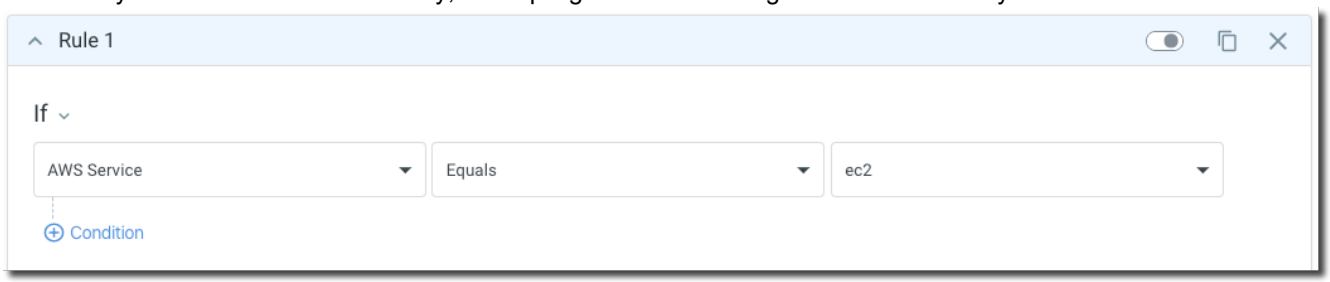
`Always rules` allow you to take action on every instance that is processed by a Monitoring Policy. It has no requirements - it will always be executed when it's encountered. This is great for things such as applying foundational Templates or tags.

When configuring your rule, simply select `Always` from the first dropdown. Whatever is selected in the resulting action step will always be applied.

Conditional rules

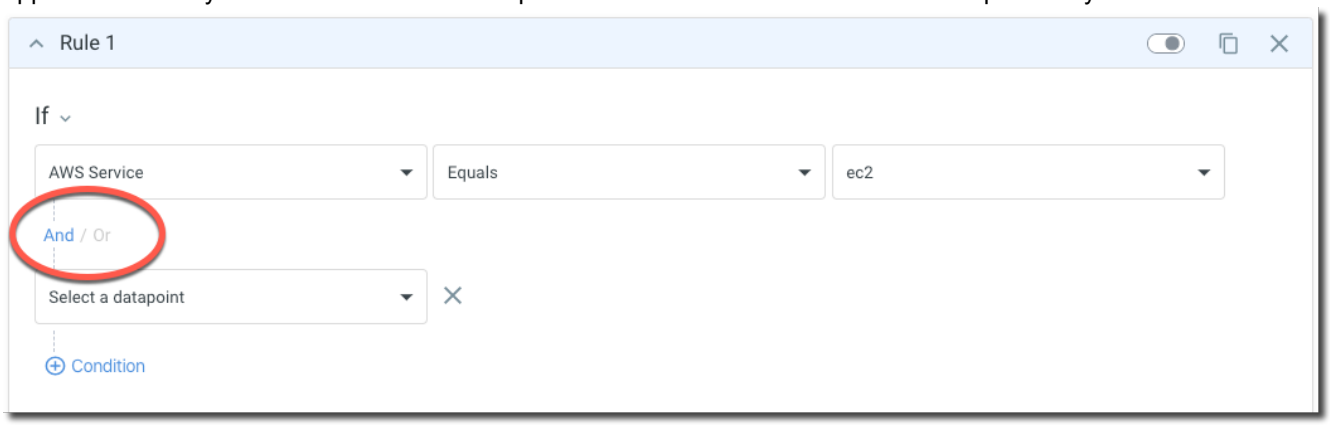
`Conditional rules` are the type you'll utilize most often. As the name suggests, an action is performed assuming a given condition is met. If the condition is not met, it merely skips over it.

The example below checks to see if an instance is an EC2 instance. You can add as many condition requirements or actions to your ruleset rule as necessary; decoupling them is encouraged for maintainability.



AND and OR

You can toggle your `Conditional rule` between an `AND` requirement to an `OR` requirement by clicking the applicable term in your filter condition. `AND` requires all conditions to be met while `OR` requires only one of `n`.



Exiting Rulesets

Maintaining small, decoupled `rulesets` is encouraged for the sake of scalability and maintainability. One tool to aid in that is the ability to exit a ruleset. When a condition is met and the action is `Exit Ruleset`, processing of that particular ruleset will cease and the workflow will move on to the ruleset. This makes it really easy to do things like "If not an EC2 instance, move on to the next ruleset."

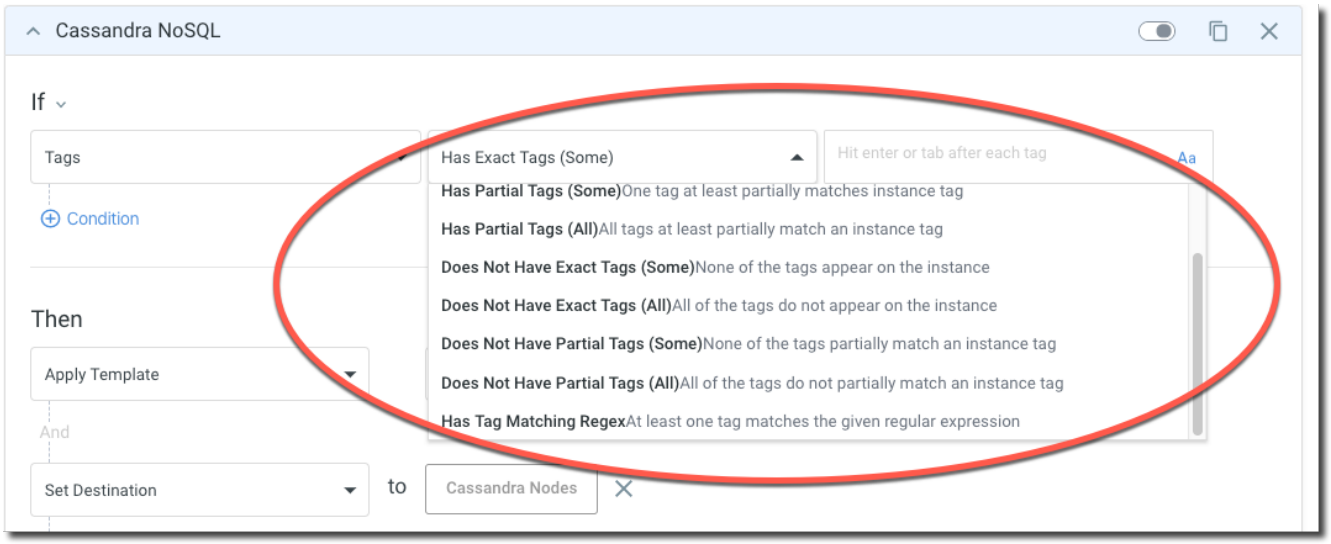
Filterable Dimensions

Dimension	Description
Always	When encountered, always evaluates to true and causes the action to always be executed
FQDN	Any known public or private IP or domain name. FQDN also supports filtering using regular expressions.
Tags	Existing tags on an instance, from the cloud provider, or agent manifest file. See Tag matching logic.
Applications	Name(s) of applications discovered to be running on the instance by the FortiMonitor monitoring agent
Instance Source	FortiMonitor Agent, Cloud, OnSight Discovery, or API
Instance Name	Name of the instance
CPU Architecture	CPU Architecture of the instance
CPU Core Count	number of cores on the instance
Kernel Version	OS kernel

Operating System	Windows, Linux
Operating System Distribution	Ubuntu, Red Hat, CoreOS, etc
Operating System Distribution Version	e.g., Ubuntu 16.04.4 LTS
AWS Availability Zone	Availability Zone the instance resides in
AWS Image ID	Instance's Image ID
AWS Instance ID	Instance's Instance ID
AWS Instance Type	Size of the instance e.g., t2.micro
AWS Region	The region the instance resides in
AWS Service	EC2, RDS, DynamoDB, etc.
Azure Availability Zone	Availability Zone the instance resides in
Azure Instance Type	Dsv3, Dv3, Fsv2, etc.
Azure Region	The region the instance resides in
Azure Service	Virtual Machine, SQL Server Database, etc.
Cloud Provider	AWS, Azure

Tag matching logic

There are several conditional rules that you can select when using Tags as a filterable dimension. These conditions are described in the table below.

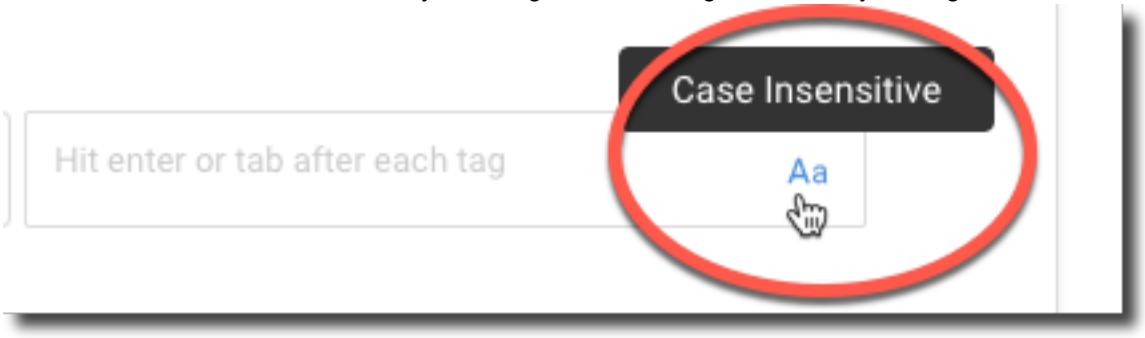


Condition	Description
Has Exact Tags (Some)	Instances that match at least one tag will be included in the ruleset.

Has Exact Tags (All)	Instances with tags that exactly match the tags entered will be added to the ruleset. For example, if the tags **linux**, **prod**, **app** are entered, only instances with the **linux**, **prod**, **app** tags will be included in the ruleset.
Has Partial Tags (Some)	Instances that have at least one tag that partially matches the entered tags will be included in the ruleset. For example, if the tag **EC2** is entered, instances with the **EC2** tag (**EC2 Redis**, **EC2 db**) tag will be included in the ruleset.
Has Partial Tags (All)	Instances with tags that partially match all the tags entered are included in the ruleset. If the tags **prod** and **database** are entered, instances with tags that partially match all of the tags are included in the ruleset. For example, instances with the **EC2 prod** and **linux database** tags.
Does Not Have Exact Tags (Some)	Instances that do not match at least one tag will be included in the ruleset.
Does Not Have Exact Tags (All)	Instances with tags that do not match the tags entered will be added to the ruleset. For example, if the tags **linux**, **prod**, **app** are entered, only instances without the **linux**, **prod**, **app** tags will be included in the ruleset. Instances with only the **linux** tag will still be included.
Does Not Have Partial Tags (Some)	Instances that do not have at least one tag that partially matches the entered tags will be included in the ruleset. For example, if the tag **EC2** is entered, instances without the **EC2** tag will be included in the ruleset.
Does Not Have Partial Tags (All)	Instances with tags that do not partially match all the tags entered are included in the ruleset. If the tags **EC2** and **database** are entered, instances with tags that partially match all of the tags are not included in the ruleset. For example, instances with the **EC2 prod** and **linux database** tags.
Has Tag Matching Regex	Instances with tags that match the given regular expression will be included in the ruleset.

Case selection

You can also select the case sensitivity of the tags when creating the ruleset by clicking **Aa**.



Actions

Action	Description
Exit Ruleset	Immediately exits the current ruleset and moves onto the next one, if present.
Exit Entire Workflow	Immediately exits the workflow and bypasses any succeeding rulesets.
Set Destination	Sets the **Instance Group** where the instance will reside in.
Add Tags	Adds additional tags to the instance - it does not remove any that are already present.
Apply Template	Applies a template to an instance - previously applied templates are not removed. Enabling the **Continually add new metrics on data collection** allows continuous discovery of new resources. For more information, see Continual matching.
Set Alert Timeline	Sets the instance's Alert Timeline, replacing what was previously set.
Set Monitoring Location	Sets the instance's Monitoring Location, replacing what was previously set.
Set Attribute	Sets the instance's attribute.
Set Name	Sets/ replaces an instance's name.
Set Destination - Computed	See Set Destination - Computed action.
Apply Template - Computed	See Apply Template - Computed.
Add Tags - Computed	Allows you to assign tags using default computed attributes. • fabric_device_serial • fabric_parent_device_name • fabric_parent_device_serial • fabric_root_device_name • fabric_root_device_serial Each attribute must be enclosed in {{}}. For example, {{fabric_device_serial}}.

Set Destination - Computed action

The **Set Destination - Computed** action allows you to assign instances to any **Instance Group** in the instance tree using a combination of string interpolation and attributes. Before using the action, attributes must be set either through a ruleset step (refer to the **Actions** table above) or through an Agent manifest file.

To use the action, perform the following steps:

1. Create a step in a ruleset by selecting **+**.
2. Set the ruleset condition to **Always**.
3. Set the action to **Set Destination - Computed**.
4. In the **Path** field, enter the target destination using an interpolated string format. Refer to the example in the following section.
   Instances matching the ruleset will automatically be assigned to the **Instance Group**. Note that the **Instance Group** will be created if it does not already exist.

Example

The following attributes are set using either the ruleset or the Agent manifest file.

Commas are not supported.

- **Attribute 1**
  - **Key**: env
  - **Value**: prod
- **Attribute 2**
  - **Key**: app
  - **Value**: web

If the path entered in the **Path** field is, **All Servers/{{env}}/{{app}}**, instances will be added to group **All Servers** > **prod** > **web**.

Where:

{ } – Curly braces are used for syntax.

/ – The slash is used to walk down the tree structure.

Apply Template - Computed action

The **Apply Template - Computed** action allows you to apply templates to any instance in your account using a combination of string interpolation and attributes. Before using the action, attributes must be set either through a ruleset step (refer to the **Actions** table above) or through an Agent manifest file.

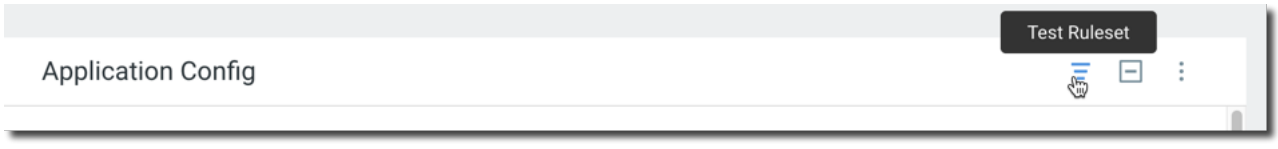
To use the action, perform the following steps:

1. Create a step in a ruleset by selecting **+**.
2. Set the ruleset condition to **Always**.
3. Set the action to **Apply Template - Computed**.
4. In the **Path** field, enter the target template using an interpolated string format. Refer to the example in the previous section.
   The template will automatically be applied to instances matching the ruleset. Note that the template must be existing.

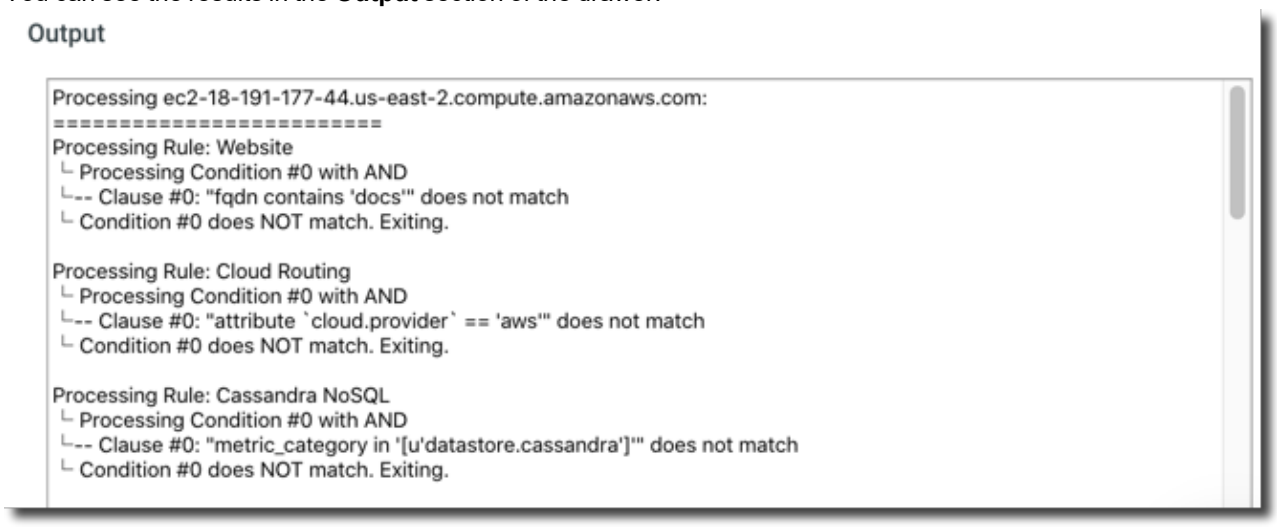
Test a ruleset

To test a ruleset, perform the following:

1. In the **Workflow Sequence** section of the **Monitoring Policy Workflow** page, select the ruleset that you want to test.
   Click the **Test Ruleset** icon.



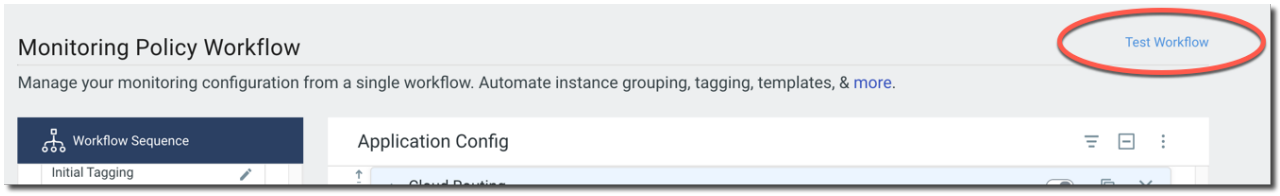
2. The **Apply Policy Workflow** drawer will slide out.

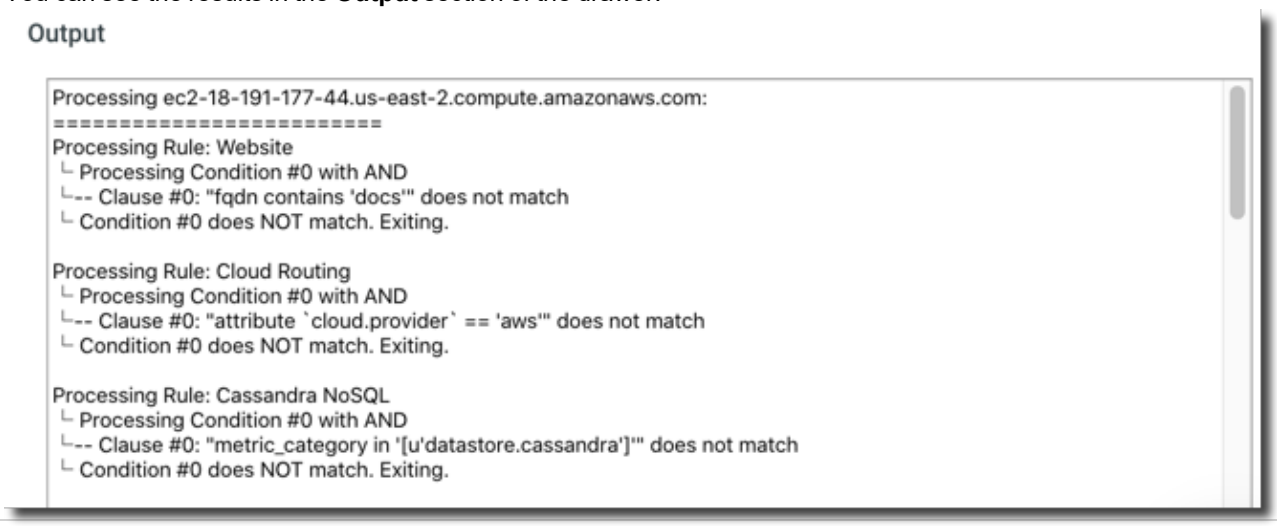
3. Select the instance or instances where you want to test the ruleset.
4. Click **Apply**.
   You can see the results in the **Output** section of the drawer.
5. 

   Note: Enable the **Commit Changes** option to apply the ruleset to the selected instances.

Test the Monitoring Policy Workflow

To test the workflow, perform the following:

1. Click **Test Workflow**.
   

   The **Apply Policy Workflow** drawer will slide out.
2. Select the instance or instances where you want to test the workflow.
3. Click **Apply**.
   You can see the results in the **Output** section of the drawer.
4. 

Note: Enable the **Commit Changes** option to apply the workflow to the selected instances.

Monitoring guides

- MySQL monitoring
- Redis monitoring

MySQL monitoring

MySQL includes several metrics that you can use to monitor connection, query, and overall database health. Regularly monitoring MySQL metrics can help you gather information that you can use when troubleshooting MySQL-related issues, such as:

- **Connectivity** - Monitor your MySQL server's availability, connections, and latency.
- **Performance** - Investigate and pinpoint performance issues.

For certain metrics, it is important to set up alerts when thresholds are breached. In the following sections, key MySQL metrics are described in detail to help you develop an alerting strategy suitable to your infrastructure.

MySQL metrics and alerts

- Synthetic checks
- MySQL application metrics

For a list of all available metrics, see MySQL.

Synthetic checks

Synthetic checks allow you to monitor servers, endpoints, and other network-based services from behind your firewall with an OnSight vCollector or from a Monitoring location. Synthetically monitoring your MySQL instance provides insights into its availability, ensuring that the instance is reachable and properly responding to requests.

To perform synthetic checks on your MySQL instance, add the following metric:

Metric	Description	Monitor	Alert
MySQL session	Checks if a session can be initiated and terminated from an OnSight vCollector or an external monitoring location.	✓	✓

MySQL application metrics

Configuring monitoring and alerting for MySQL can help you identify and investigate potential problems on the MySQL application itself. The metrics described in the following sections gather data to help you answer questions such as:

- *Is the database available and effectively providing service to clients?*
- *Are there queries performing poorly?*

- *What is the connection load?*

In addition, monitoring metrics like caches, threads, and connections can help you identify if your proper resources are sufficient. Click the following links to review the recommended monitoring and alerting strategy for each metric:

- Availability metrics
- Query performance metrics
- Other metrics

Availability metrics

The following metrics validate that your MySQL instance is working and can be reached by clients.

Metric	Description	Monitor	Alert
MySQL connections	The total number of processes (threads) connected to the MySQL server. This metric can help you assess the number of connections your instance will support. When the configured maximum number of connections has been reached, new client connections will be refused. Consider increasing the maximum connections limit or adding more resources if client connections are frequently refused.	✓	
MySQL slave server is connected to the master	The status of the slave-master servers connection, when applicable. MySQL replication spreads out the load among multiple slaves to improve query performance. Performance issues may occur if replication threads fail to run on slave instances.	✓	
MySQL slave server is replicating	Checks if the master server is replicating to the slave server. MySQL replication spreads out the load among multiple slaves to improve query performance. Performance issues may occur if replication threads fail to run on slave instances.	✓	✓
Uptime	The number of seconds since the MySQL server was started. This metric can be used to detect restarts. A consistent low uptime value suggests that the instance is frequently being restarted, causing service interruptions.	✓	✓

Query performance metrics

The following metrics measure the performance of your MySQL instance. Monitoring metrics that track queries, threads, and server latency can be added to the instance.

Metric	Description	Monitor	Alert
Slow queries/second	The average number of queries per second that took more than the configured query time to execute. A high metric value suggests that queries are not being executed optimally. Check the slow query log and identify the slow queries for optimization.	✓	
Queries/second	The average number of queries per second. If you are getting consistently high values and experiencing performance degradation, consider scaling up your infrastructure to handle more queries.	✓	✓
MySQL slave server latency (seconds)	The latency between the slave and the master servers.	✓	✓
Threads connected	The current number of open connections. This metric provides real-time insight into the number of clients connected to the instance. Monitoring the metric can help you analyze traffic or evaluate when to do a restart.	✓	
Threads created	The total number of processes (threads) created to handle MySQL connections.	✓	
Threads running	The total number of processes (threads) running on the MySQL instance.	✓	
Questions - count of statements executed from the client	The number of statements executed by the server from the client.	✓	

Other metrics

InnoDB locks and cache status metrics are described in the following table.

Metric	Description	Monitor	Alert
InnoDB row lock current waits	The current number of times operations on the InnoDB tables had to wait for a row lock. To improve the performance of MySQL read and write operations, tune the size of the buffer pool used by InnoDB to cache table and index data.	✓	
InnoDB row lock time avg	The average time, in milliseconds, to get a row lock for InnoDB tables.	✓	
MySQL query cache amount free (KB)	The amount of free space (in KB) in the MySQL query cache.	✓	

MySQL query cache percent free	The percentage of free space in the MySQL query cache.	✓	

Redis monitoring

This article describes all the available Redis metrics that you can monitor and alert on in FortiMonitor. For certain metrics, it is important to set up alerts when thresholds are breached. In the following sections, key Redis metrics are described in detail. Configure monitoring and/or alerting thresholds based on your application requirements.

For a list of all available metrics, see Redis.

Redis application metrics

Monitoring and setting up alerts for Redis-specific metrics can help you identify and investigate potential problems on the Redis application itself. The metrics described in the following sections gather data to help you plan for future infrastructure upgrades or troubleshoot issues related to availability and performance.

Availability metrics

The following metrics validate that your Redis instance can be reached by clients.

Metric	Description	Monitor	Alert
Connected clients	The number of clients connected to your Redis application node. Once Redis reaches its configured **maxclients**, it stops accepting new connections and produces an error.	✓	
Uptime in seconds	The number of seconds since the Redis instance was started. Use this metric to ensure that the Redis instance is not restarting frequently.	✓	✓
Connected slaves	The number of slaves connected to the master. Latency may occur if the slave is unable to connect to the master. Also, changes to the number of connected slaves may indicate issues with the master or the slave instances.	✓	
Replication: role (master=1, slave=0)	The role of the Redis instance in terms of replication.	✓	

Query performance metrics

The following metrics measure the performance of your Redis instance. Metrics that track used memory, expired and evicted keys, and blocked clients can be added to the instance.

Metric	Description	Monitor	Alert
Used memory	The amount of memory in bytes allocated by the Redis instance. If the used memory exceeds the system memory of your Redis instance, the system will start swapping causing performance issues.	✓	
Used memory peak	The peak memory used by the Redis instance.	✓	
Used memory rss	The amount of memory in bytes allocated by the Redis instance as seen by the OS.	✓	
Expired keys	Redis supports the time-to-live (TTL) property for each key. The key is deleted once the TTL expires. Expired keys can pile up in the memory unless evicted.	✓	
Evicted keys	The number of keys removed when the **maxmemory** limit is reached. Evicting a large number of keys may cause latency.	✓	✓
Blocked clients	The number of clients currently blocked while waiting on a blocking request.	✓	
Keyspace hits	The number of successful key lookups in the main dictionary. A low hits-to-misses ratio means that clients are looking for keys that have expired or evicted	✓	
Keyspace misses	The number of failed key lookups in the main dictionary A low hits-to-misses ratio means that clients are looking for keys that have expired or evicted	✓	✓
Persistence: background save in progress	The number of background save processes. These processes causes degraded performance. You can use this metric to diagnose whether performance issues are caused by background save processes.	✓	

Other metrics

Other metrics such as length of lists, total commands processed, and more are described in the following table.

Metric	Description	Monitor	Alert
Length of list	The number of entries in a Redis list.	✓	
Changes since last save	The number of changes to the Redis database since the last dump. This is the data that will be lost when the Redis instance restarts.	✓	

Pub/sub channels	The total number of channels used for pub/sub.	✓
Pub/sub patterns	The number of pub/sub patterns.	✓
Total commands processed	The total number of commands processed by the Redis. The value is incremented every time Redis completes a command from connected clients. Since commands in Redis are processed sequentially, a high number of commands may cause latency. This metric is one of the things you can look at to diagnose latency issues.	

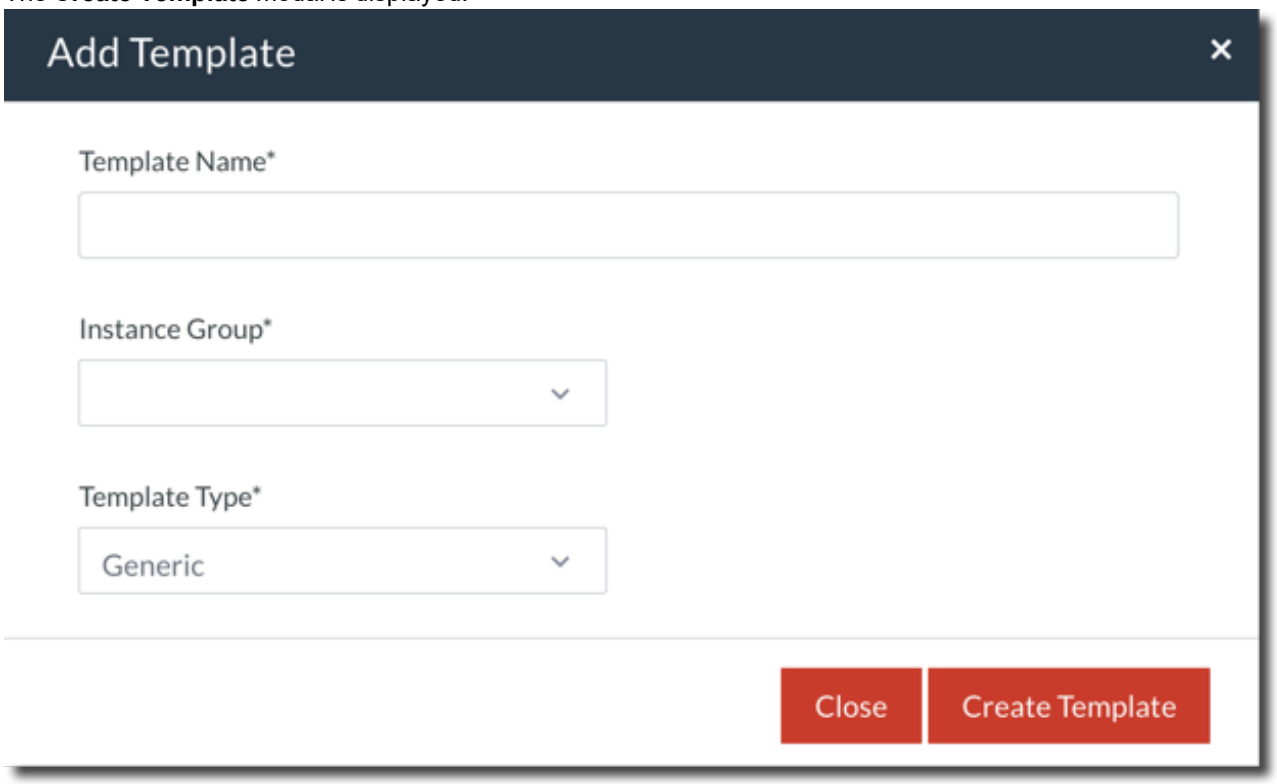
Templates

Templates are a fantastic tool for any of our customers. If you have several servers that you would like to monitor in the same fashion, a server template is a perfect way to accomplish this configuration quickly. Once you have created a template, you can change the configuration whenever you like.

Create a template from an existing server

From the **Instance Details** page of the server that you want to use as the base for the template, select **3-dot menu** > **Create Template**.

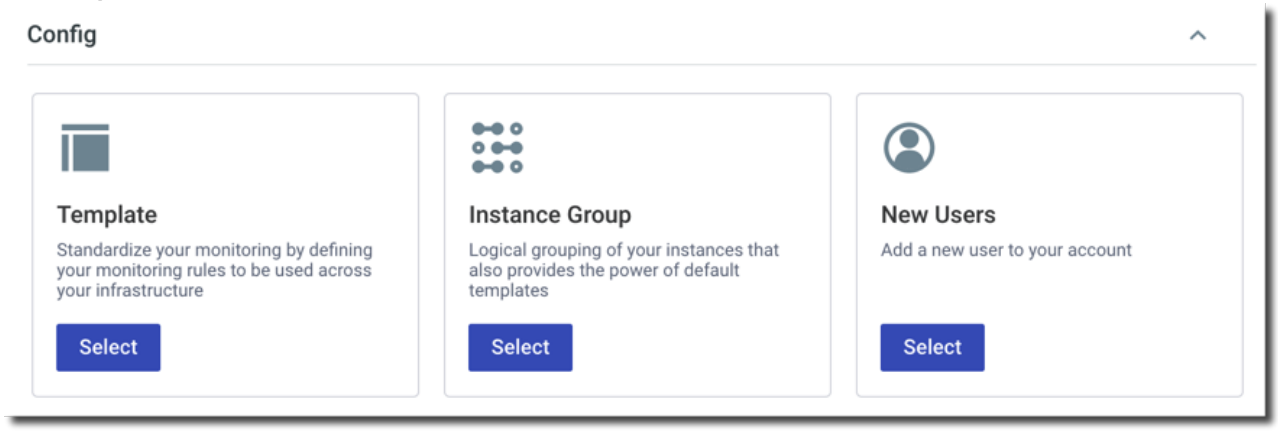
The **Create Template** modal is displayed.



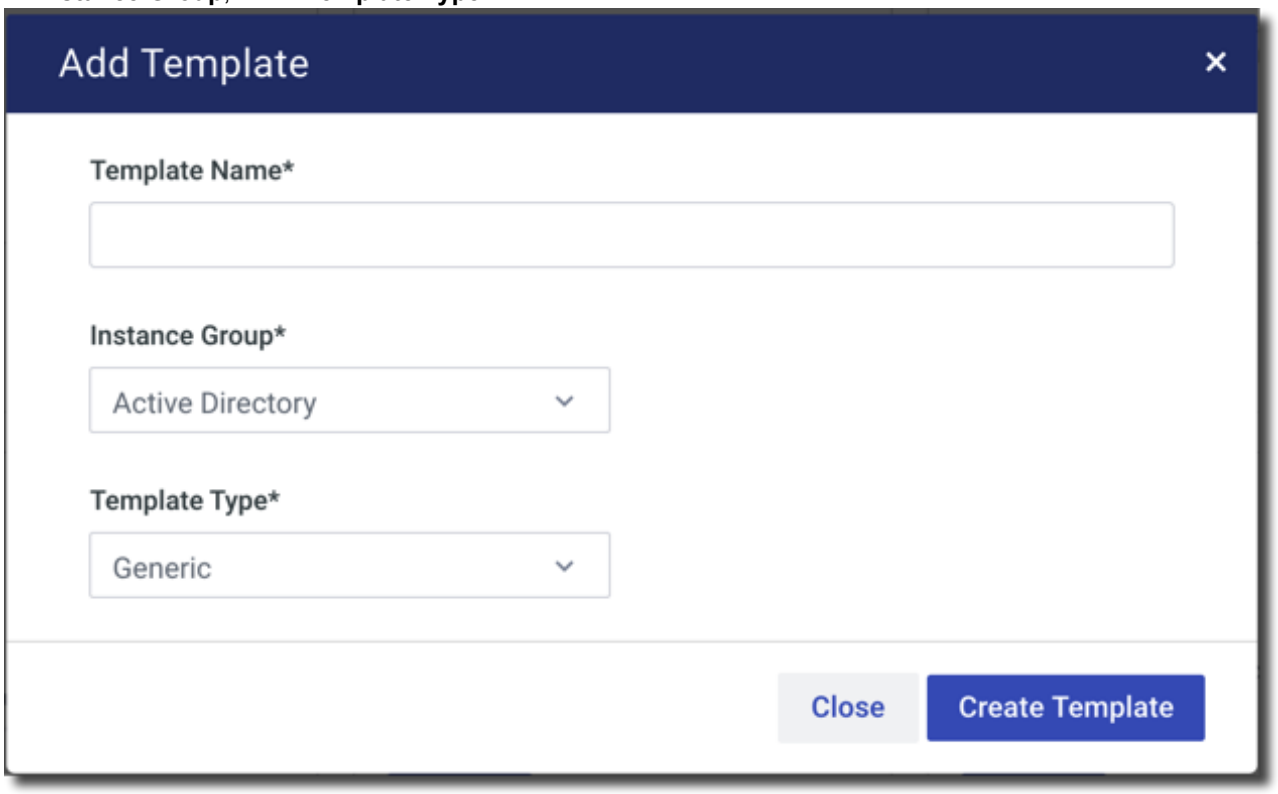
1.
2. Select a name for the template.
3. Select an **Instance Group**. This is where the new template will be located.
4. Select a **Template Type**.
   - **Network Device** – For network devices with SNMP and network device metrics.
   - **Generic** – General purpose template for all instances (except network devices).
   - **VMware** – For VMware virtual machines.
   - **Fabric** - For Fabric integration
   - **Application Monitoring** - For Application monitoring and DEM
   - **Third Party** - For third-party integrations
5. Click **Submit**. This will create and open the template.
6. To configure the template, see Configure a template.

Create a template from scratch

Click **Add** from the navigation menu. From the **Config** section of the **Infrastructure and Resource Catalog**, select the **Template** card.

1. 

Selecting **Template** will open the **Create Template** modal where you can name your template and choose an **Instance Group**, and a **Template Type**.

2. 

3. Select a name for the template.
4. Select an **Instance Group**. This is where the new template will be located.
5. Select a **Template Type**.
   - **Network Device** – For network devices with SNMP and network device metrics
   - **Generic** – General purpose template for all instances (except network devices)
   - **VMware** – For VMware virtual machines

Click **Submit**. This will create and open the template.

6. To configure the template, see Configure a template.

Default templates

FortiMonitor's default templates can facilitate provisioning of predefined metrics to the different types of infrastructure in your environment. This allows you to easily apply the same set of metrics to instances that you want to monitor in the same way.

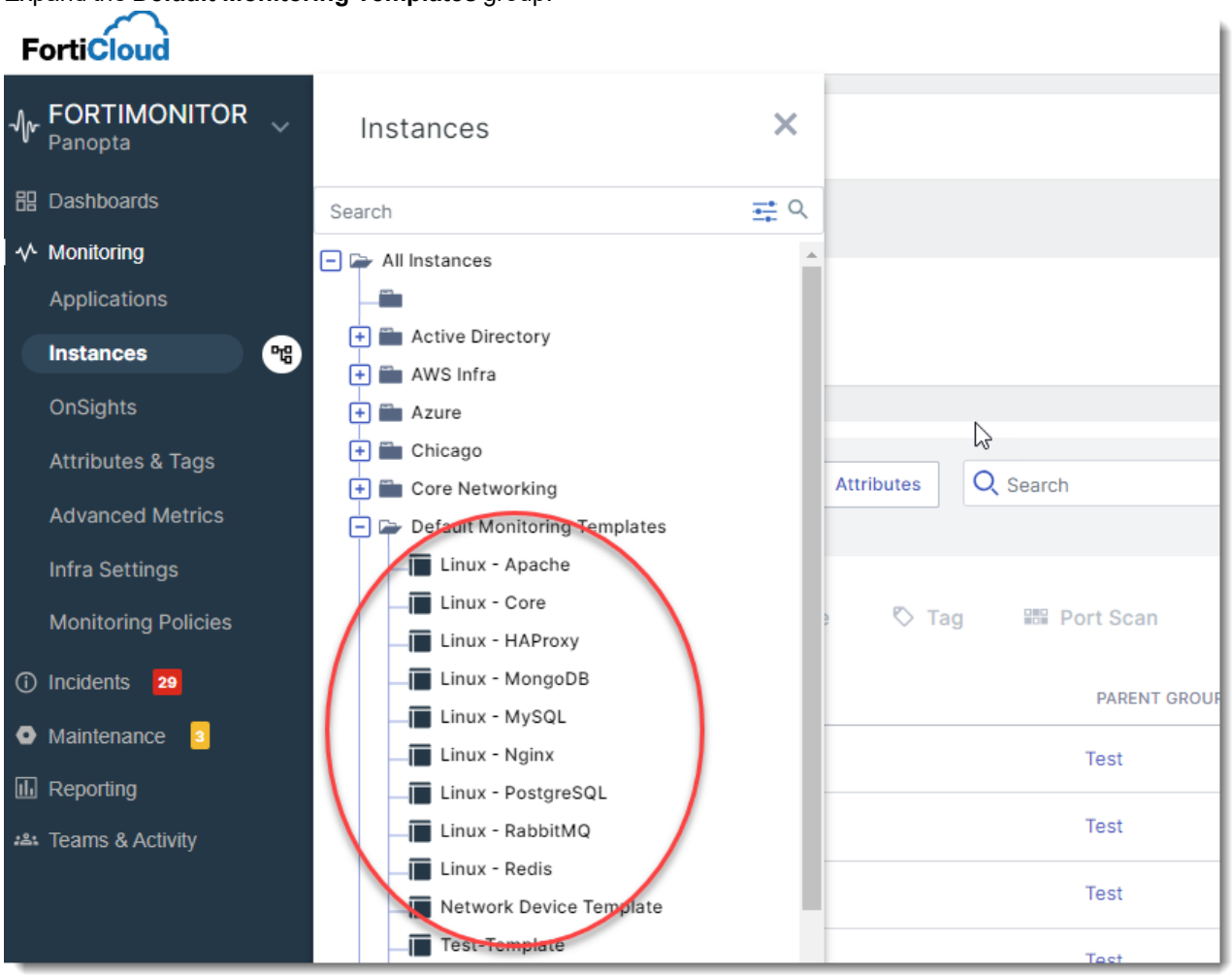
For example, when the default template **Linux - Apache** is applied to all of your Linux-based instances that host Apache webservers, the predefined metrics from the templates will be provisioned to these instances.

Although the default templates contain the most common metrics for a particular type, they do not include all of the supported metrics to prevent alerts from metrics you do not need.

Except for disk, memory, and CPU metrics, no other metric will have preconfigured alert thresholds. To configure alert thresholds, see Incident alert thresholds.

FortiMonitor's available collection of default templates is initially hidden. Individual default templates become visible once applicable instances are onboarded into the environment. In the example illustrated below, several Linux templates are visible. From the Control Panel, you can see a list of the default templates that have already been pulled into your environment by performing the following steps:

1. Click **Instances** to expand the Instances drawer.
2. Expand the **Default Monitoring Templates** group.



The following sections also detail which metrics are predefined for each default template.

Linux

- Linux - Apache
- Linux - Core
- Linux - MongoDB
- Linux - MySQL
- Linux - PostgreSQL
- Linux - Redis

Linux - Apache

Tags	Metric category	Metrics
Apache Linux	Apache	2xx Requests
		4xx Requests
		5xx Requests
		Active workers (%)
		Active workers (%) - All
		CPU used by workers
		CPU used by workers - All
		Idle workers (%)
		Idle workers (%) - All
		Requests per second
	Linux: Process	Process: Process Running Status - executable name and args - apache2

Linux - Core

Tags	Metric category	Metrics
Linux	Linux: Bandwidth	Bandwidth: Kilobytes IN per second
		Bandwidth: Kilobytes IN per second - All
		Bandwidth: Kilobytes OUT per second
		Bandwidth: Kilobytes OUT per second - All
	Linux: CPU	CPU % Used
	Linux: Disk	Disk: Percentage of disk used
		Disk: Percentage of disk used
		Disk: Percentage of disk used - All
	Linux: Memory	Memory: RAM % usage
	Linux: NTP	NTP: Difference between machine and NTP time

Linux - MongoDB

Tags	Metric category	Metrics
Linux MongoDB	MongoDB	Current connections
		Number of operations queued waiting for read lock
		Number of operations queued waiting for write lock
		Objects
		Open cursors
		Page Faults
		Reads/sec
		Size of documents
		Size of indexes
		Writes/sec

Linux - MySQL

Tags	Metric category	Metrics
Linux MySQL	MySQL	Questions - count of statements executed from the client
		Requests made to the buffer pool
		Requests unfulfilled by the buffer pool
		Slow queries per second
		Threads connected
		Threads running
		Total pages in the buffer pool

Linux - PostgreSQL

Tags	Metric category	Metrics

Linux postgresql	PostgreSQL	Active connections
		Blocks read from buffer cache
		Blocks read from disk
		Buffer cache hit rate (%)
		Connections waiting for lock
		DELETES
		Idle connections
		Idle in transaction connections
		INSERTS
		Minutes between database checkpoints

Linux - Redis

Tags	Metric category	Metrics
Linux redis	Redis	Blocked clients
		Changes since last save
		Connected clients
		Connected slaves
		Evicted keys
		Expired keys
		Keyspace hits
		Keyspace misses
		Total commands processed
		Used memory

Windows

- Windows - Active Directory
- Windows - Core
- Windows - Exchange- Core
- Windows - IIS
- Windows - SQL Server

Windows - Active Directory

Tags	Metric category	Metrics
windows activedirectory	Active Directory	DRA Inbound Bytes Total/sec
		DRA Inbound Objects/sec
		DRA Inbound Values Total/sec
		DRA Outbound Bytes Total/sec
		DRA Outbound Objects/sec
		DRA Outbound Values Total/sec
		DRA Pending Replication Synchronizations
		DS Client Binds/sec
		DS Directory Reads/sec
		DS Directory Searches/sec
	Windows: Service	Active Directory Certificate Service
		Active Directory Domain Services
		Active Directory Federation Services
		Active Directory Web Services

Windows - Core

Tags	Metric category	Metrics
windows	Windows: Bandwidth	Bandwidth: Bytes Received/sec
		Bandwidth: Bytes Sent/sec
	Windows: CPU	CPU % Used
	Windows: Disk - Logical	Disk - Logical: Disk Reads/sec
		Disk - Logical: Disk Writes/sec
		Disk - Logical: Percent used
	Windows: Memory	Memory: Percent used

Windows - Exchange- Core

Tags	Metric category	Metrics

windows exchange	.NET	# Bytes in all Heaps
		# of Excepts Thrown / sec
		% Time in GC
		Application Restarts
		Request Wait Time
		Requests Executing
		Requests In Application Queue
	Exchange Core	Database Mounted
		I/O Database Reads (Attached) Average Latency
		I/O Database Reads (Attached)/sec
		I/O Database Reads (Recovery) Average Latency
		I/O Database Writes (Attached) Average Latency
		I/O Database Writes (Attached)/sec
		I/O Database Writes (Recovery) Average Latency
		I/O Log Writes Average Latency
		I/O Log Writes/sec
		Log Record Stalls/sec
	Windows: Memory	Memory: Free System Page Table Entries
		Memory: Pages/sec
		Memory: Pool Nonpaged Bytes
		Memory: Pool Paged Bytes
	Windows: Service	Microsoft Exchange DAG Management
		Microsoft Exchange Information Store
		Microsoft Exchange Replication Service
		Microsoft Exchange RPC Client Access

Windows - IIS

Tags	Metric category	Metrics

windows IIS	IIS	Bytes received/sec
		Bytes sent/sec
		Current Application Pool State
		Current Connections
		Get Requests/sec
		Post Requests/sec

Windows - SQL Server

Tags	Metric category	Metrics
windows sqlserver	SQLServer	Batch Requests/Sec
		Buffer Cache Hit Ratio
		Checkpoint Pages/Sec
		Compilations/Sec
		Page Life Expectancy
		Processes Blocked
		Re-Compilations/Sec
		Transactions/Sec
		User Connections

Network devices

Tags	Metric category	Metrics
Linux redis	Admin Status	ifAdminStatus
	Operational status	ifOperStatus

Virtual machines

Tags	Metric category	Metrics

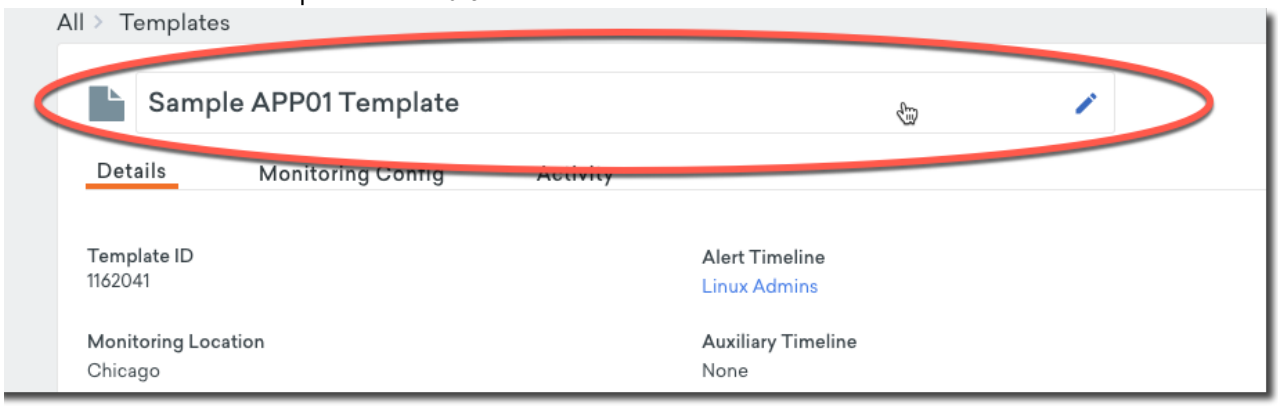
azure virtualmachines	CPU	CPU Credits Consumed
		CPU Credits Remaining
		Percentage CPU
	Disk	Disk Read Bytes
		Disk Read Operations/Sec
		Disk Write Bytes
		Disk Write Operations/Sec
	Network	Network In
		Network Out

Configure a template

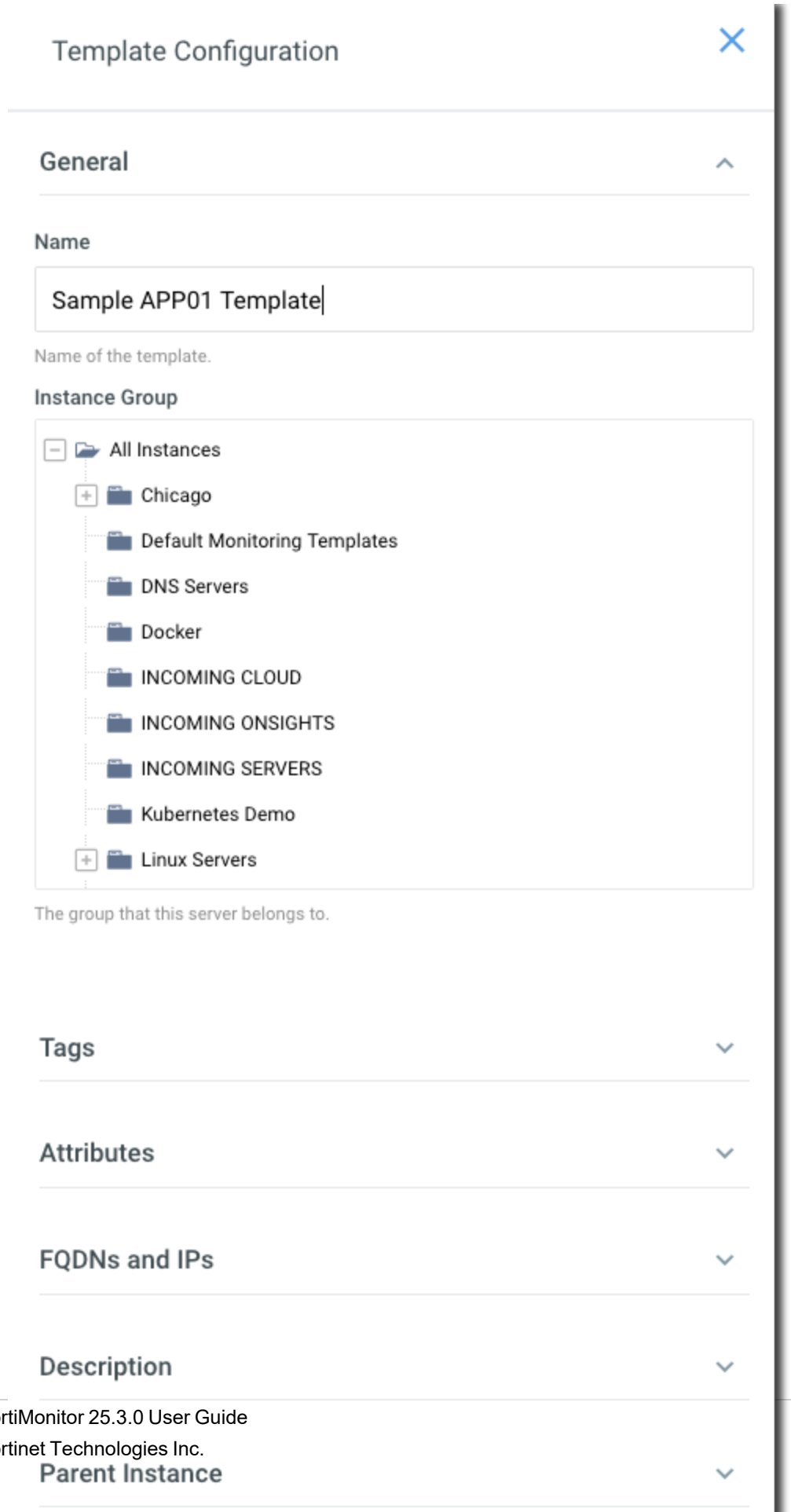
You can configure a new or existing template at any time by going to the template's details page.

Configure a new or existing template

To configure a template, perform the following steps:

1. Go to the page of the template that you want to edit.
   Select the name of the template or the **Edit** icon.
2. 

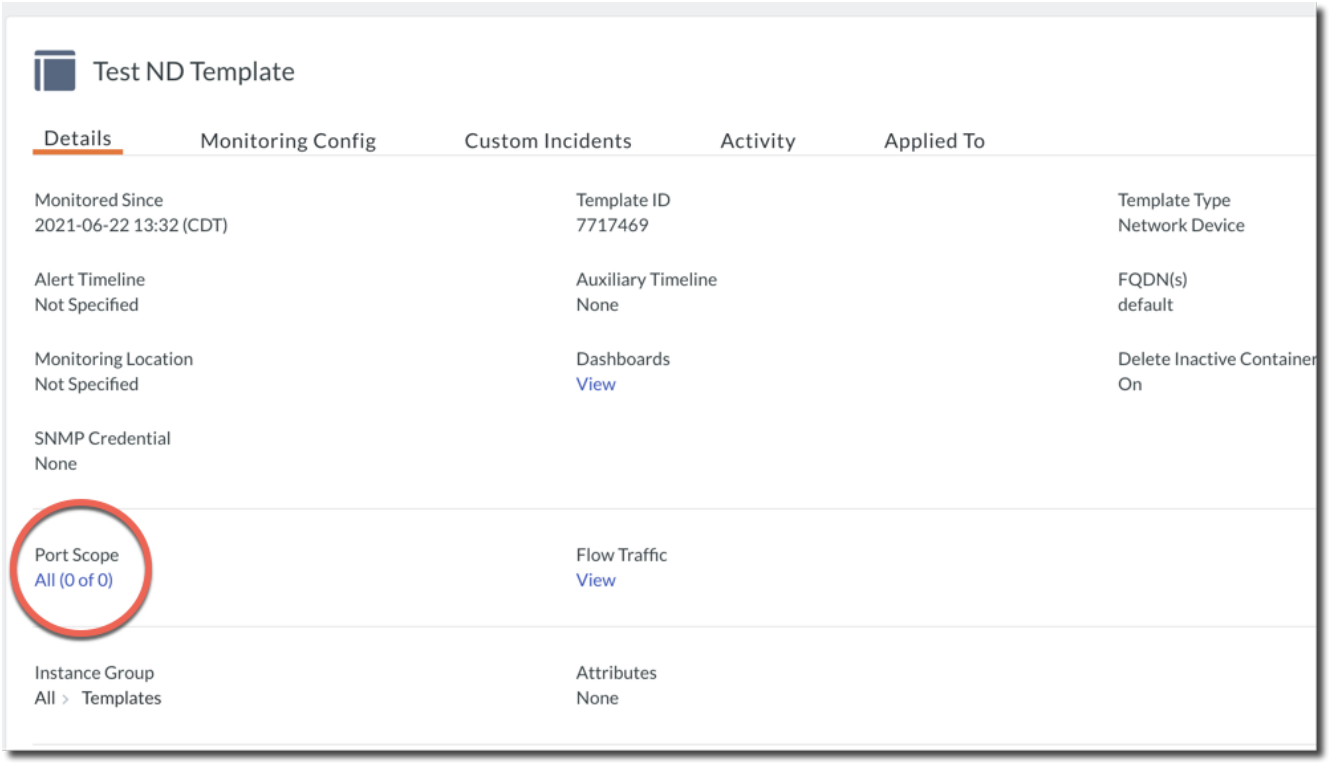
   The **Template Configuration** drawer will slide out.


Template Configuration
General
Name
Sample APP01 Template
Name of the template.
Instance Group
All Instances
Chicago
Default Monitoring Templates
DNS Servers
Docker
INCOMING CLOUD
INCOMING ONSIGHTS
INCOMING SERVERS
Kubernetes Demo
Linux Servers
The group that this server belongs to.
Tags
Attributes
FQDNs and IPs
Description
Parent Instance

3. Edit the following settings as needed:
   - **General** – Change the name of the template and select a server group for the template to reside in.
   - **Tags** – Enter tags that can help define the instance where the template will be applied. Agent heartbeat tags can also be defined here.
   - **Attributes** – Add key-value pairs that can provide structured instance metadata.
   - **FQDNs and IPs** – Add additional interface names. See Additional IP addresses or FQDNs.
   - **Description** – Enter a description for the template.
   - **Parent Instance** – (Optional) The device upstream from this server that it depends on. If the parent device experiences an outage, this server's outage notifications will be suppressed.
   - **SNMP Credentials** – Add an SNMP credential to the template.
   - **Prometheus Endpoints** – Integrate the instance where the template will be applied with Prometheus. To add a Prometheus endpoint:
     - Select **Add Endpoint**.
     - Select a **Host Name**.
     - Enter the Prometheus endpoint. For example **prometheus-rabbitmq-exporter-7fb45675b-nvqbk:9419/metrics**.
     - Select **Save Changes**.
       The **Host Name** will be computed when the template is applied and the URI will then be appended.
4. Select **Save Changes**.

Port filtering

For network device templates, you can filter the number of ports that you can monitor using Port Scope. To configure the ports to be monitored, click the link below **Port Scope**.

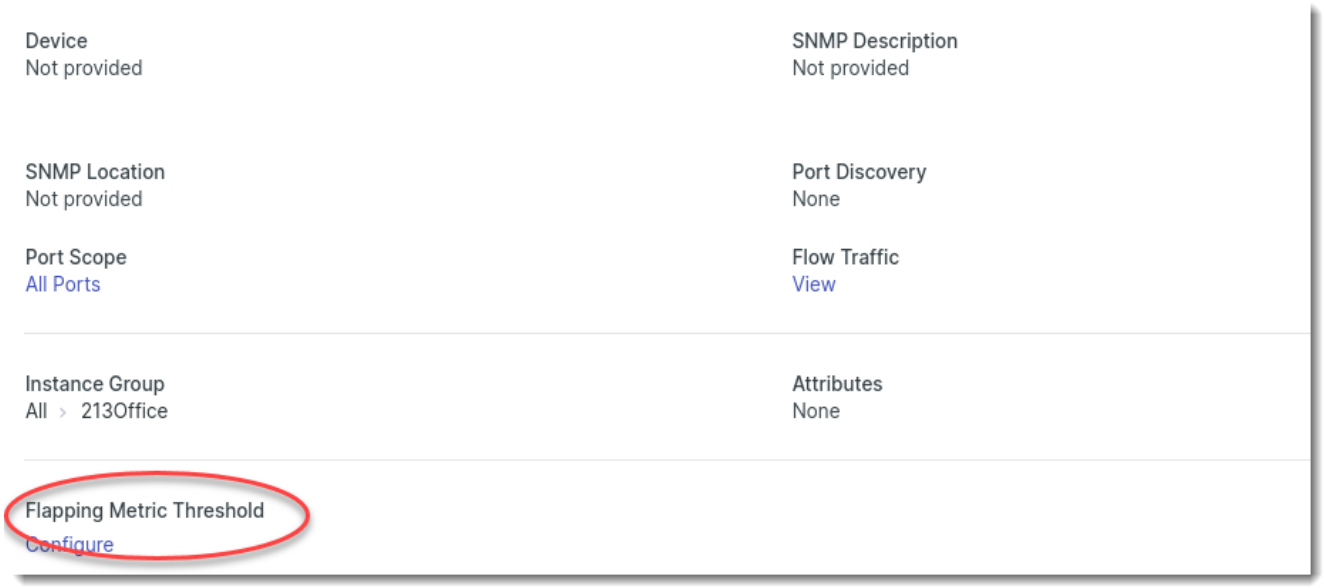


The following options are available:

- **Use Device's Settings** – Add the ports for monitoring as configured on the network device.
- **All Ports** – Add all of the ports on the device for monitoring.
- **By Name Filter** – Select the ports that will be monitored using regex or the port's name.

Flapping Metric incident

You can configure a Flapping Metric incident threshold using the template.



Continual matching

One of the great benefits of using templates in hand with the FortiMonitor server agent is the continual discovery of new resources. Every hour (or when manually triggered), the FortiMonitor server agent looks for new resources on the server - things such as drives/partitions, network interfaces, applications, etc. By default, as we find these new resources, we will run them through the metric rules you have defined in your template. If the new resources match any of the defined metric rules, we'll begin monitoring them according to your template definition.

To turn this feature off, uncheck the continual discovery option that appears in the Apply template confirmation modal when applying the template to a server instance.

Dynamic variables

You can also use dynamic variables in your templates if needed, read our article about it here.

Monitoring location

You can choose to select any monitoring location you would like for a template. This monitoring location will then override the monitoring location of any server you apply it do. If you do not want this to happen select **Not specified** for

your **Primary Monitoring Location**.

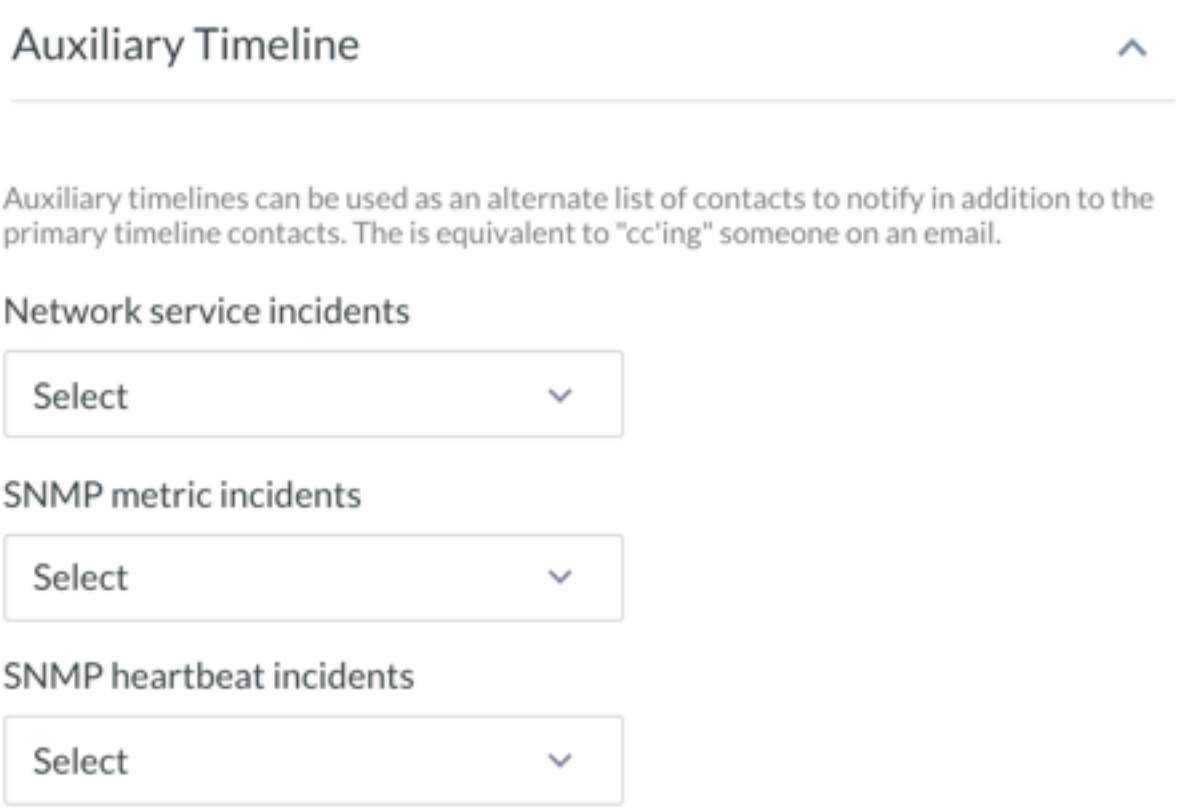
You can also choose to set your monitoring locations on any individual network service checks you may create for this template. This can be useful if, for example, you want a template that does ping checks from multiple locations around the world.

Alert timelines

You can assign the following Alert timelines to the template by selecting **Alert Timeline** from the **Details** tab of the template:

- **Primary Alert Timeline** – The primary alert timeline to use for server incidents.
- **Agent Heartbeat Alert Timeline** – Determines which alert timeline to send alerts to when no communication is being received from the FortiMonitor Agent.
- **Auxiliary Timeline** – Auxiliary timelines can be used as an alternate list of contacts to notify in addition to the primary timeline contacts. This is equivalent to copying someone on an email. Also, depending on the type of the instance, you can specify an alert timeline for each of the following incident types:
  - Network service incidents
  - SNMP metric incidents

    SNMP heartbeat incidents



- **SNMP Heartbeat Alert Timeline** – Determines which alert timeline to send alerts to when no communication is being received from SNMP polling.
- **Metric Heartbeat Alert Timeline** – Determines which Alert Timeline to send alerts to when no communication is received from one or more metrics configured on the instance.

See Assign Alert Timelines for more information.

Configure template metrics

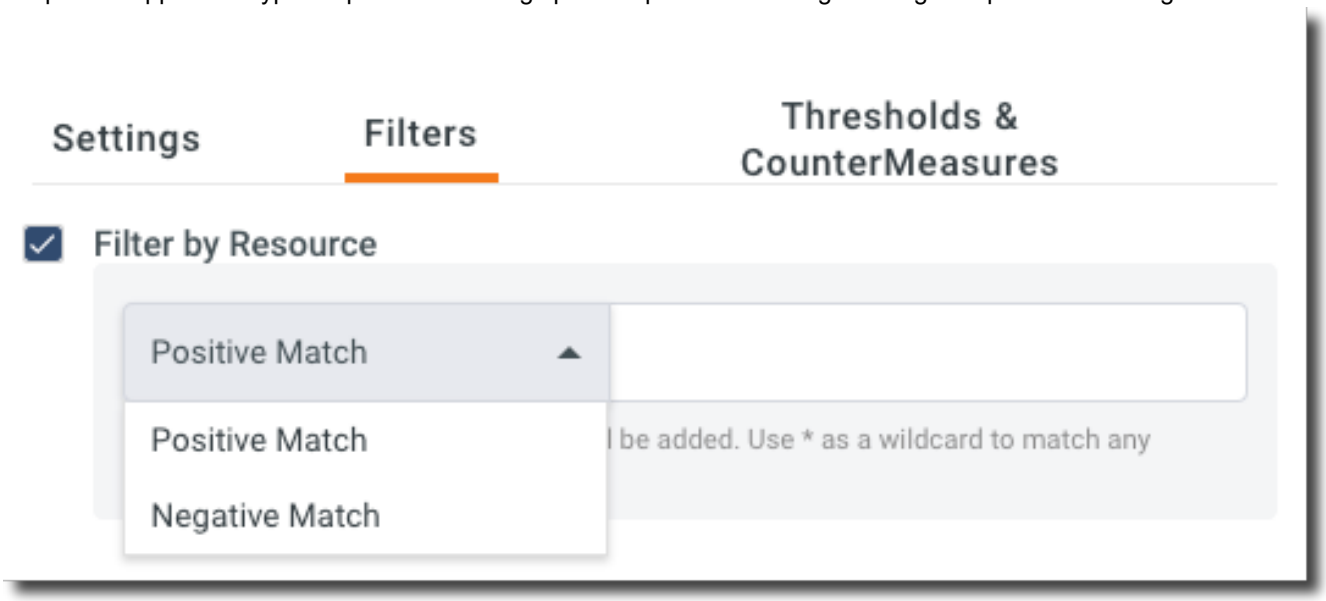
1. To configure template metrics, open an existing template.
2. To add checks, go to the **Monitoring config** tab and select **Add Monitoring**.
3. Configure the metrics and thresholds as required. Keep in mind, thresholds set will also be set when a template is applied to a server.

There will be no detected metrics, but you can select the checks you want to configure from the catalog. Remember that these should be basic checks, as they will be applied to multiple servers.

Pattern matching

Often, naming across your infrastructure resources does not match exactly - whether it's network interfaces, disk drives/partitions, or database instances; however, they are often close enough that a simple pattern match will allow you to target your desired resource.

To help you deal with this ambiguity and minimize the number of template metrics you have to manage, FortiMonitor templates support two types of pattern matching - positive pattern matching and negative pattern matching.



Configuration

Under the **Filters** tab of a metric's configuration drawer, you can choose your match selection preference from the following:

- **Select from list** - If you are creating a template off of a server, you will still retain the ability to pick discovered resource options from a select list.
- **Positive match** - allows you to supply alpha-numeric characters and wildcards * to match against resources. For example, if your network interfaces were eth01, eth02, and nic01, you could match just the first two with eth*. When you apply the template, only the first two interfaces would have monitoring added.

- **Negative match** - is as you'd expect, the opposite. By providing a pattern of eth* would lead to only the nic01 interface to have monitoring added to it.

Additional IP addresses or FQDNs

If you would like your template to be able to apply to servers with multiple IP addresses then you will want to use this field. Here you will invent your labels for those multiple IP addresses. For example, if you want this template to cover servers with both a public and a private IP address you could enter **PublicIP**, **PrivateIP** into this field. When applying a template with additional IP addresses you will be given the option to map the additional IP addresses of the template to the additional IP addresses of your server.

The Agent manifest file will also let you take advantage of this feature with **Interface Mapping**, to learn more about this you can see this article.

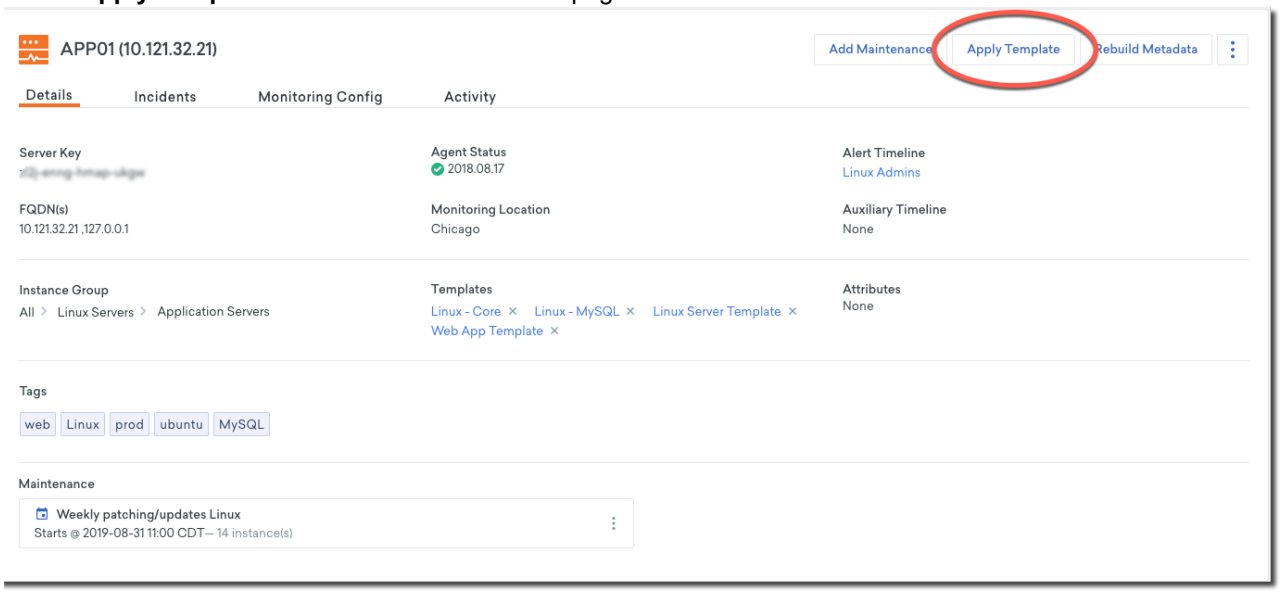
Apply server templates

There are a few ways you can apply a template to your server: during instance creation, manually on an existing instance, during Agent installation using the Agent manifest file, and as a default setting for instance groups.

Apply a template manually from the Instance Details page

You can manually apply a template to an instance by performing the following steps:

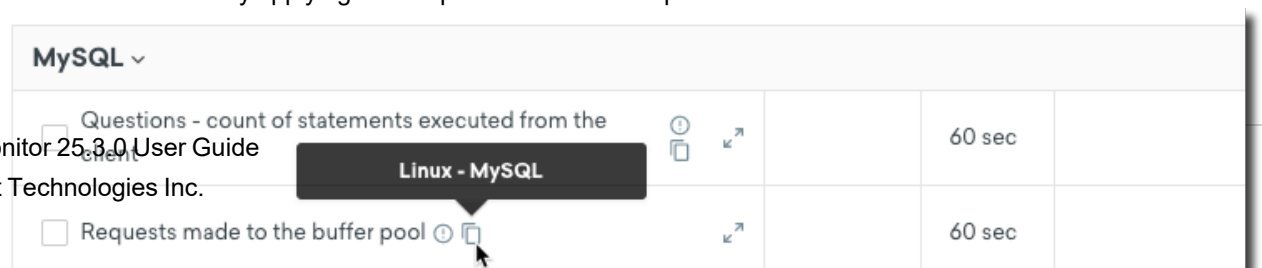
1. Select **Apply Template** from the **Instance Details** page.



A modal will open, allowing you to select which template you would like to apply to the server.

Click **Apply Template.**

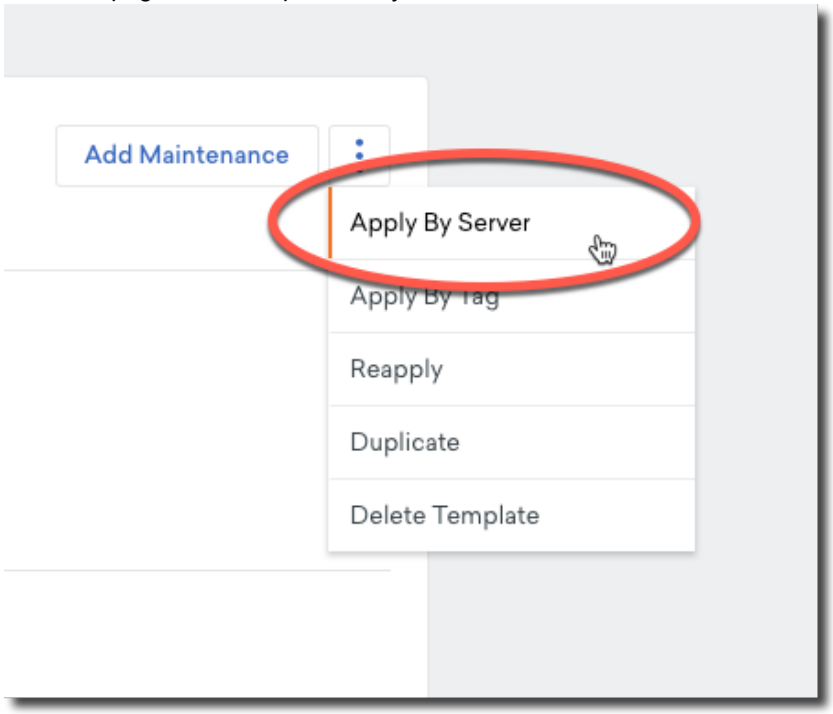
The metrics created by applying the template will have a template icon next to their name.



Apply a template manually from the template's page

You can apply templates to multiple servers by performing the following steps:

1. Go to the page of the template that you want to use and click the **3-dot menu** > **Apply by Server**.

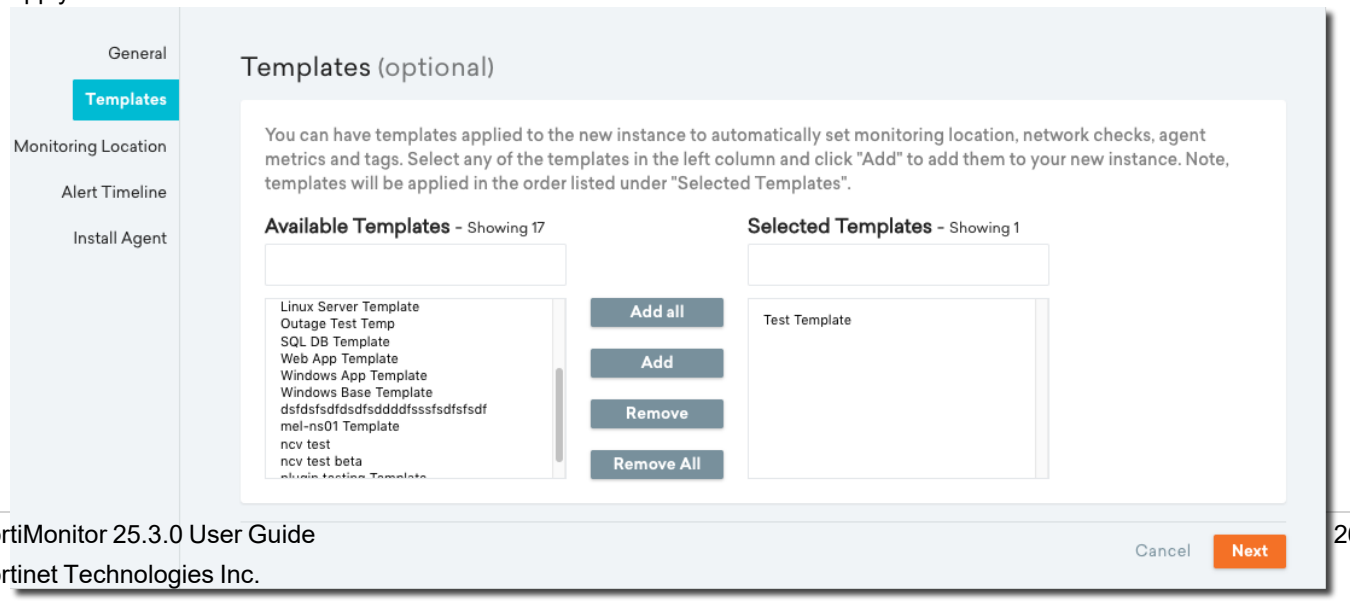


A modal will open allowing you to select the servers and server groups that you would like to apply the template to.

2. Select the servers or server groups that you want to apply the template to then click **Apply Template**.
   A confirmation modal will appear.
3. Select **Apply Template** on the modal to apply the template to the selected servers.

Apply a template during initial instance configuration

When adding an instance manually (not with the automated Agent method), one of the steps is to choose a template to apply to this new server.



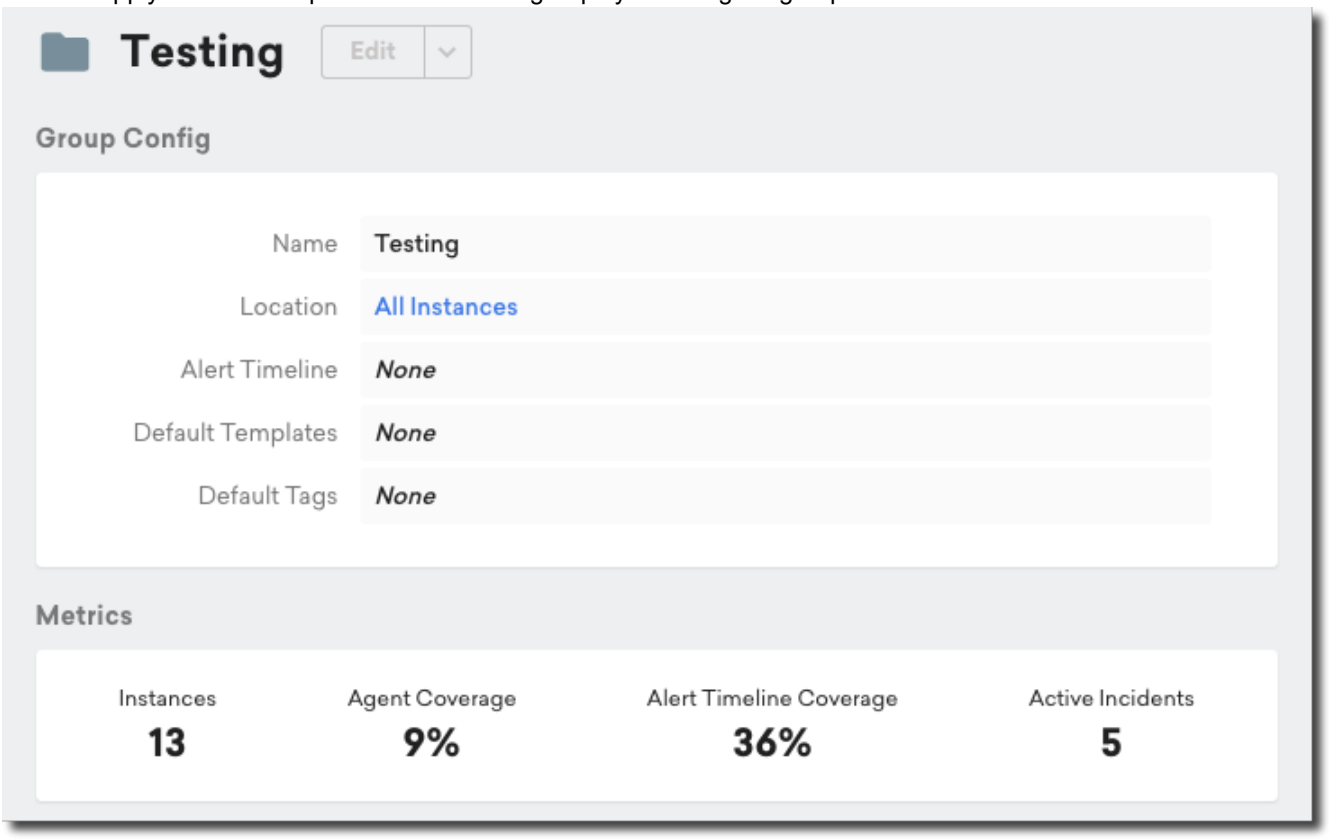
If you choose to use this method, the checks, thresholds, alert timelines, and anything on the template will be applied to the server upon creation.

Apply a template during Agent installation

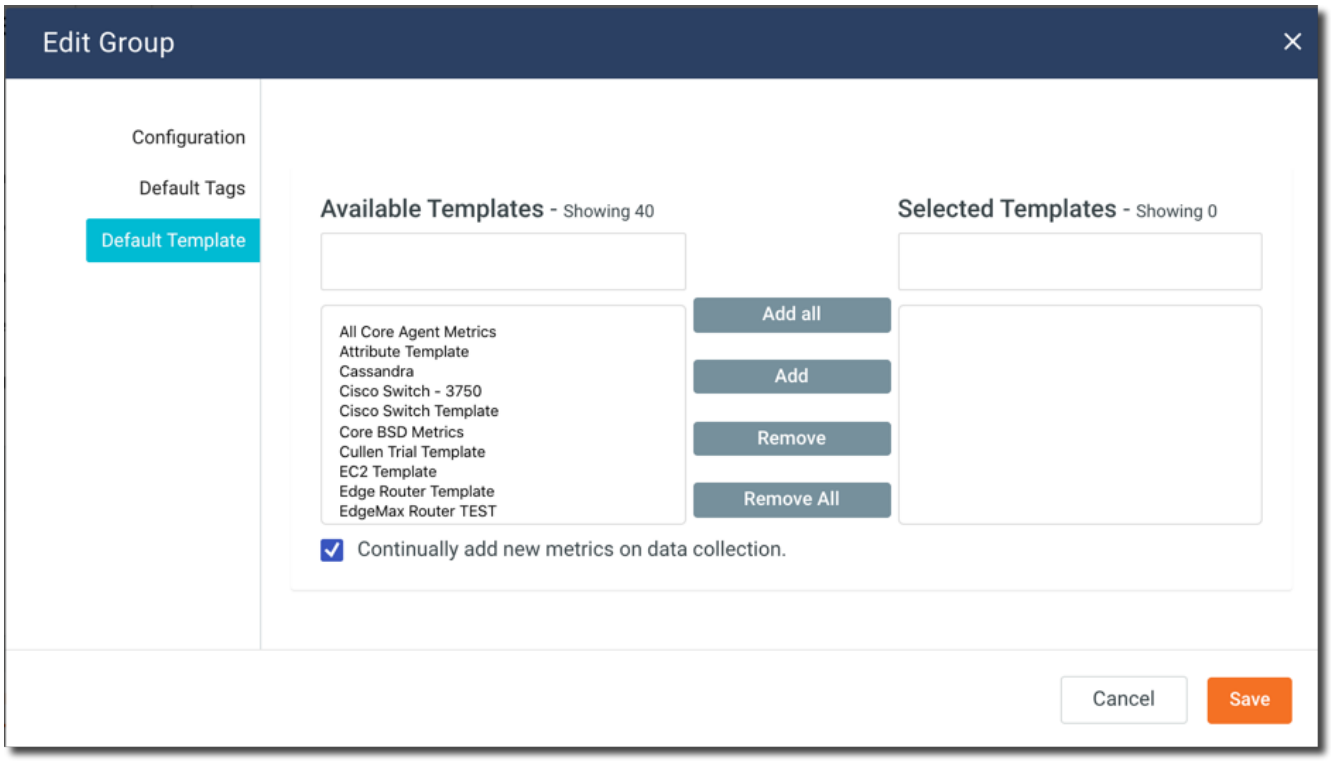
When using the Agent manifest file during Agent installation, you can specify a template to use in the manifest file. You can read more about the manifest file in this article.

Apply a template to an instance group

You can apply a default template to an instance group by selecting the group from the Instances tree in the left drawer.



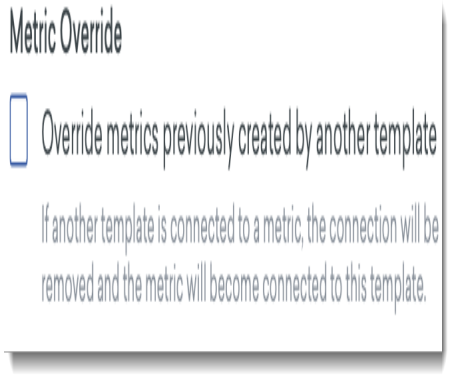
Now that you are on the group's page, you can edit the group to apply a default template. This way, any servers added to the group in the future will automatically have this template applied to them as well.



Enabling the option **Continually add new metrics on data collection** while editing the list of templates to apply also enables this option for all of the instances in the group.

Metric override when applying templates

The **Metric Override** option allows you to override metric settings defined from a previously applied template. Enabling the option removes the metric's association with its original template and configures the metric based on the new template.



To enable the option, perform the following steps:

1. From the **Instance Details** page of the template that you want to edit, select the **Monitoring Config** tab.
2. In the row of the metric that you would like to set the override on, click the **3-dot menu** on the right side. Select **Edit**.
3. Select the **Metric Override** checkbox

4. Select **Save.**
5. **Apply** the template to the desired instances. The metric will be added, overriding any metrics previously added from other templates.

Template pattern matching

Monitoring templates are an incredibly powerful feature within FortiMonitor. They enable you to scale your monitoring in a cohesive and consistent way. Taking templates a step further with pattern matching allows your monitoring strategy to scale in an automated fashion.

A brief example

Let's say you want to monitor every network interface that starts with "eth" across every server. In the past, this would be very time-consuming. With pattern matching, however, you can create a single interface-related check on a template - say Bandwidth In - and use "eth*" for wildcard pattern matching. Now, every time you apply this template to a server, it will add a metric for every interface that starts with "eth" on the server. Even better, you can have this matching be performed every time meta-data discovery is performed. If new resources are discovered that match your pattern, metrics will be created for them automatically.

Try it yourself

Let's see how this works, step by step.

1. Create a new template.
2. On the newly created template's details page, click the **Monitoring Config** tab.
3. Click **Add Monitoring**.
4. From the catalog, look for the **Linux** card then click **Add**.
5. With the Linux metrics drawer now open, click **Add** for **kb in for the mon**.
6. Click the **Filters** tab.
7. Enable the **Filter by Interface** option.
8. Select from the following options:
   - **Positive match** - allows you to supply alpha-numeric characters and wildcards * to match against resources. For example, if your network interfaces were eth01, eth02, and nic01, you could match just the first two with eth*. When you apply the template, only the first two interfaces would have monitoring added.
   - **Negative match** - is as you would expect, the opposite. By providing a pattern of eth* would lead to only the nic01 interface to have monitoring added to it.
9. Click **Save**.

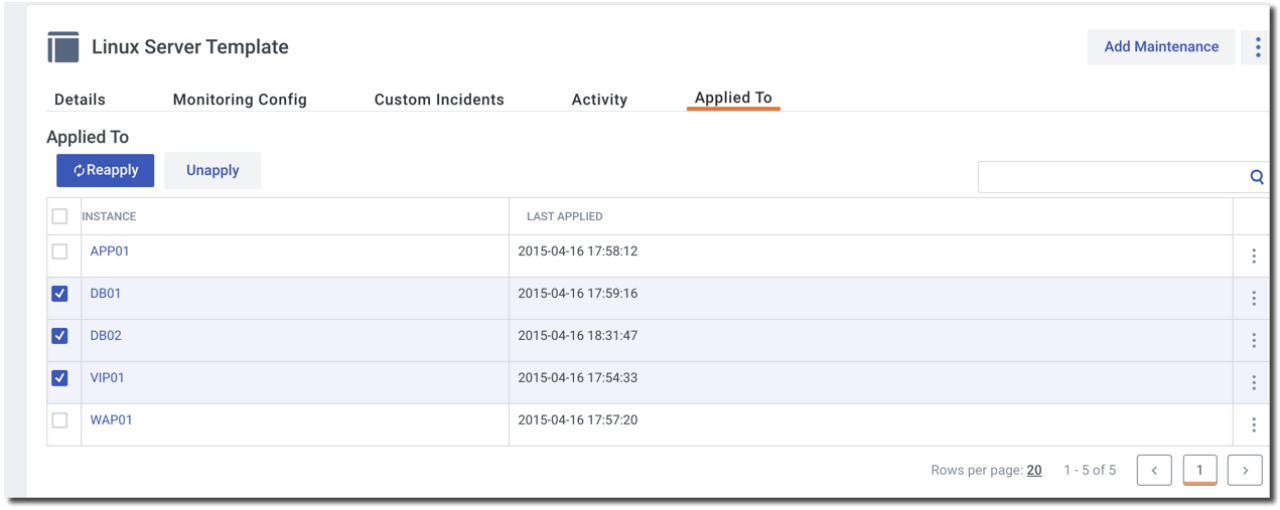
Now your *template* has a single metric on it. You'll notice in the monitoring table, the metric that you just added has the pattern matching rule next to its name as a friendly reminder.

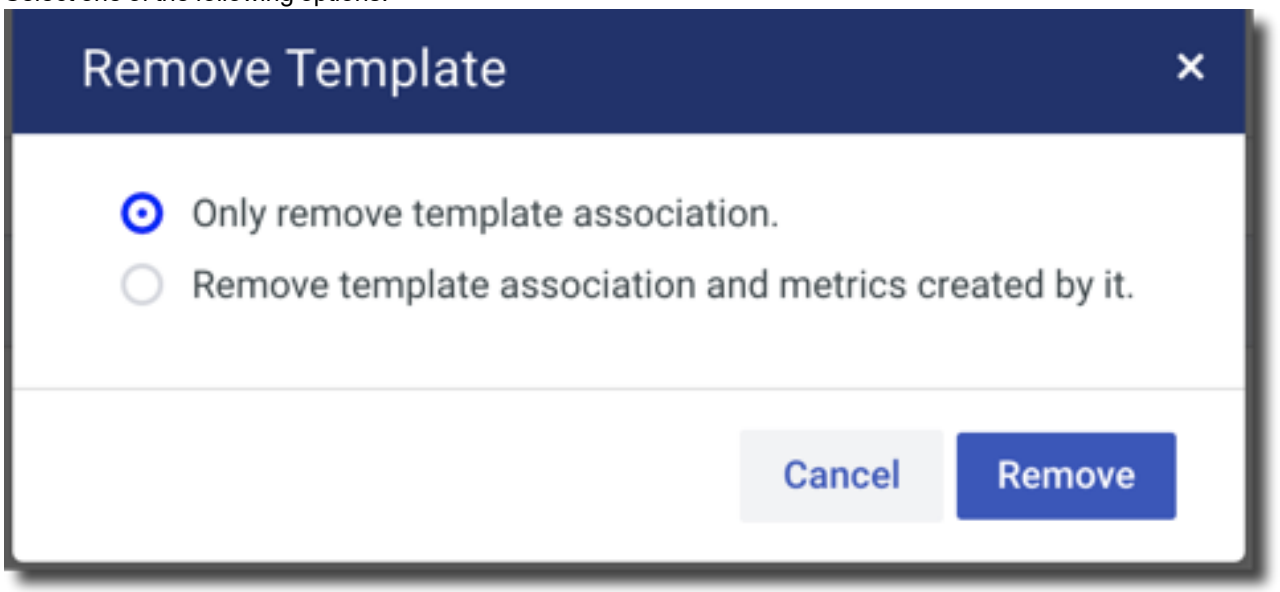
Now, let's apply this template to some servers. The steps are detailed in Apply server templates.

If you visit any of the servers you applied the template to, you'll see a metric that has been created for every network interface which the FortiMonitor agent had previously discovered on the given server.

Disassociate a template from multiple instances

You can disassociate a template from multiple instances by performing the following steps:

1. Go to the template's details page.
2. Click the **Applied to** tab.
   Select the instances that you want to disassociate from the template.
3. 

4. Click **Unapply**.
   Select one of the following options:
5. 

   Selecting the **Only remove template association** option will leave all existing metrics that were created from the template in place and will continue to collect data.
6. Click **Remove**. The instance/instances should disappear from the table after a few seconds.

Cloud monitoring

- AWS
- Azure
- Kubernetes

AWS

FortiMonitor offers native integration with AWS CloudWatch, enabling FortiMonitor to ingest your CloudWatch monitoring data. As well, FortiMonitor can perform automatic discovery and monitoring of instances within your AWS account. This is configurable by service type and region, and can also be fully customized using your AWS tags.

CloudWatch data should be used as an augmentation of, not a replacement for, the data obtained by the FortiMonitor server agent and external monitoring. The server agent can provide more detailed and accurate data across any OS distribution or application you may be running on your compute instance. As well, our external monitoring ensures you're getting the full picture of your current operating environment as well as a view into what your customers are experiencing.

Connecting FortiMonitor and CloudWatch

To grant FortiMonitor access to your CloudWatch data, you'll need to create an external account role within your AWS account that is tied to FortiMonitor's External AWS Account.

1. From the navigation menu, click **Add**. The **Infrastructure and Resource Catalog** will be displayed**.**
2. Select **Amazon Web Services**.
3. Enter an a name for the integration.
4. Follow the on-page instructions to create an IAM Policy and Role for the external FortiMonitor account.
5. Once you've obtained your ARN, select **Verify Connection**.
6. Once your ARN has been validated, you can configure your monitoring settings.
   - **Services:** Select the AWS services you'd like to monitor. It's better to only select the ones you're using, otherwise, it uses vital API calls.
   - **Filter Instances by Tag:** You can choose to only import instances that match the AWS tag filters you define.
   - **Regions:** Only select the regions you operate in, otherwise, it uses vital API calls
   - **Options - Import Tags:** Enabling tag import will pull in your AWS tags with your AWS instances. You have the option to import only the Value portion of the AWS tag or the entire AWS key-value pair as a single string tag.
   - **Options - Import AWS tags as FortiMonitor Attributes**: AWS tags will be imported as key-value pairs (attributes within FortiMonitor).
   - **Options - Routinely scan for new instances:** every 20 minutes, we'll look for new instances in your account and will begin monitoring them assuming they meet your filter criteria. EC2 instances using the FortiMonitor agent can be monitored immediately if you install the agent on boot.
   - **Options - Apply Monitoring Policies**: Apply a monitoring policy to the imported AWS instances.
   - **Options - Destination Group:** any time instances are imported, they'll be placed in this group in the control panel. This is great for setting default values which are inherited from their parent group as well as apply default templates.
   - **Options - Template:** apply a Template to every instance that's imported
7. Click **Complete Integration**. We'll start pulling in your instances that meet your filter criteria and begin monitoring them.

API Limits and Throttling

By default, each AWS account gets 1M CloudWatch API calls per month for free. When FortiMonitor makes CloudWatch calls to obtain metrics (every 10 minutes), it utilizes your API calls quota. Due to the highly decoupled design of the CloudWatch API, calls have to be made on a per-instance-per-metric basis - this means API calls add up fast. We encourage you to utilize the FortiMonitor agent on EC2 instances, not only for the cost savings but also for the increased functionality and granularity. You can read more about it here.

Once you exceed 1M CloudWatch calls for the month, AWS will charge your account $10 per 1M calls. You can read more about their pricing here.

In certain large-scale scenarios, AWS could begin throttling API calls. We will begin backing-off at that time. If you expect to utilize close to or the full 1M calls per month, we recommend reaching out to AWS to ask for a limit increase. If you'd like FortiMonitor to collect CloudWatch metrics more often than every 10 minutes, please contact our support team. As well, you can override this at the metric level by editing the metric. Check out Templates to do this in bulk.

Existing Monitoring

If you're running the agent (Linux version > 2017.40, Windows version > 18.34), EC2 metrics will be automatically added to your existing agent-based instances.

Example: if you have a Linux Virtual Machine instance you're already monitoring with the agent, and the agent version is > 2017.40, we won't create a second "EC2" instance with the CloudWatch connection - the new CloudWatch metrics will be added to your existing instance

EC2 Incident Confirmation

If you're monitoring an EC2 instance with external checks - such as HTTP, HTTPS, or Ping - and we identify an incident, we'll first confirm with AWS that the instance is still around. If it was gracefully removed, we will not alert. If the instance was not removed gracefully, we will alert as normal.

AWS integrations

The following AWS integrations are supported (free or charged per instance):

Free

- Amazon ASG
- Amazon EBS
- Amazon FSX
- AWS Lambda
- Amazon S3
- Amazon SNS
- Amazon SQS
- AWS VPN

Paid

- Amazon DynamoDB
- Amazon RDS
- Amazon RDS Cluster
- AWS ELB
- AWS ELBv2
- Amazon Redshift
- Amazon Elasticsearch Service
- Amazon ElastiCache
- Amazon EC2
- Amazon Route 53
- AWS Gateway Load Balancer
- AWS Transit Gateway
- AWS Direct Connect
- Amazon NAT Gateways

Amazon DynamoDB

The **Amazon DynamoDB** integration allows you to monitor your Amazon DynamoDB service through FortiMonitor.

To add the Amazon DynamoDB service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **DynamoDB**.

Metrics

The following table describes the metrics supported by FortiMonitor for DynamoDB.

See https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/metrics-dimensions.html for more information.

Metric	Unit	Description
Conditional Check Failed Requests	Count	The number of failed attempts to perform conditional operations.
Consumed Read Capacity Units	Count	The number of read capacity units consumed over a specified amount of time.
Consumed Write Capacity Units	Count	The number of write capacity units consumed over a specified amount of time.
Online Index Consumed Write Capacity	Count	The number of write capacity units consumed when adding a new global secondary index to a table.
Online Index Percentage Progress	Count	The progress in percent of a new global secondary index that is being added to a table.

Online Index Throttle Events	Count	The number of write throttle events that occur when adding a new global secondary index to a table.
Provisioned Read Capacity Units	Count	The number of provisioned read capacity units for a table or a global secondary index.
Provisioned Write Capacity Units	Count	The number of provisioned write capacity units for a table or a global secondary index.
Read Throttle Events	Count	Requests to DynamoDB that exceed the provisioned read capacity units for a table or a global secondary index.
Returned Bytes	Bytes	The number of bytes returned by **GetRecords** operations during the specified amount of time.
Returned Item Count	Count	The number of items returned by **Query** or **Scan** operations during the specified amount of time.
Returned Records Count	Count	The number of stream records returned by **GetRecords** operations during the specified amount of time.
Successful Request Latency	Milliseconds	Successful requests to DynamoDB or Amazon DynamoDB Streams during the specified amount of time.
System Errors	Count	Requests to DynamoDB or Amazon DynamoDB Streams that generate an HTTP 500 status code during the specified amount of time.
Time To Live Deleted Item Count	Count	The number of items deleted by Time to live (TTL) during the specified amount of time.
Throttled Requests	Count	Requests to DynamoDB that exceed the provisioned throughput limits on a resource.
User Errors	Count	Requests to DynamoDB or Amazon DynamoDB Streams that generate an HTTP 400 status code during the specified amount of time.
Write Throttle Events	Count	Requests to DynamoDB that exceed the provisioned write capacity units for a table or a global secondary index.

Amazon EBS

The **Amazon Elastic Block Store (EBS)** integration allows you to monitor your Amazon EBS service through FortiMonitor.

To add the Amazon EBS service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **EBS**.

Metrics

The following table describes the metrics supported by FortiMonitor for EBS.

See https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-volume-status.html for more information.

Metric	Unit	Description
Volume Read Bytes	Bytes	Provides information on the read operations in a specified amount of time.
Volume Write Bytes	Bytes	Provides information on the write operations in a specified amount of time.
Volume Read Ops	Count	The total number of read operations in a specified amount of time.
Volume Write Ops	Count	The total number of write operations in a specified amount of time.
Volume Total Read Time	Seconds	The total amount of time (in seconds) spent by all read operations that completed in a specified amount of time.
Volume Total Write Time	Seconds	The total amount of time (in seconds) spent by all write operations that completed in a specified period of time.
Volume Idle Time	Seconds	The total amount of time (in seconds) when no read or write operations were submitted.
Volume Queue Length	Count	The number of read and write operation requests waiting to be completed in a specified amount of time.
Volume Throughput Percentage	Percent	The percentage of I/O operations per second (IOPS) delivered of the total IOPS provisioned for an Amazon EBS volume. Applies to Provisioned IOPS SSD volumes only.
Volume Consumed Read Write Ops	Count	The total amount of read and write operations consumed in a specified period of time. Applies to Provisioned IOPS SSD volumes only.
Burst Balance	Percent	Provides information about the percentage of I/O credits (for gp2) or throughput credits (for st1 and sc1) remaining in the burst bucket. Applies to General Purpose SSD (gp2), Throughput Optimized HDD (st1), and Cold HDD (sc1) volumes only.

Amazon EC2

The **Amazon Elastic Compute Cloud (Amazon EC2)** integration allows you to monitor your Amazon EC2 service through FortiMonitor.

To add the Amazon EC2 service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **EC2**.

Metrics

The following table describes the metrics supported by FortiMonitor for EC2.

See https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/viewing_metrics_with_cloudwatch.html for more information.

Metric	Unit	Description
CPU Utilization	Percent	The percentage of allocated EC2 compute units that are currently in use on the instance.
Disk read ops	Count	Completed read operations from all instance store volumes available to the instance in a specified period of time.
Disk write ops	Count	Completed write operations to all instance store volumes available to the instance in a specified period of time.
Disk read bytes	Bytes	Bytes read from all instance store volumes available to the instance.
Disk write bytes	Bytes	Bytes written to all instance store volumes available to the instance.
Network bytes in	Bytes	The number of bytes received on all network interfaces by the instance.
Network bytes out	Bytes	The number of bytes sent out on all network interfaces by the instance.
Network packets in	Count	The number of packets received on all network interfaces by the instance.
Network packets out	Count	The number of packets sent out on all network interfaces by the instance.
CPU Credit Usage	Credits (vCPU-minutes)	The number of CPU credits spent by the instance for CPU utilization.
CPU Credit Balance	Credits (vCPU-minutes)	The number of earned CPU credits that an instance has accrued since it was launched or started.
CPU Surplus Credit Balance	Credits (vCPU-minutes)	The number of surplus credits that have been spent by an unlimited instance when its CPUCreditBalance value is zero.

CPU Surplus Credits Charged	Credits (vCPU-minutes)	The number of spent surplus credits that are not paid down by earned CPU credits, and which thus incur an additional charge.
Status Check Failed	Count	Reports whether the instance has passed both the instance status check and the system status check in the last minute.
Status Check Failed Instance	Count	Reports whether the instance has passed the instance status check in the last minute.
Status Check Failed System	Count	Reports whether the instance has passed the system status check in the last minute.

Amazon EC2 ASG

The **Amazon EC2 Auto Scaling Group (ASG)** integration allows you to monitor your Amazon ASG service through FortiMonitor.

To add the Amazon ASG service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **ASG**.

Metrics

The following table describes the metrics supported by FortiMonitor for ASG.

See https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-instance-monitoring.html#as-group-metrics for more information.

Metric	Unit	Description
Total Instances	Count	The total number of instances in the Auto Scaling group. This metric identifies the number of instances that are in service, pending, and terminating.
Terminating Instances	Count	The number of instances that are in the process of terminating. This metric does not include instances that are in service or pending.
Standby Instances	Count	The number of instances that are in a `Standby` state. Instances in this state are still running but are not actively in service.
Pending Instances	Count	The number of instances that are pending. A pending instance is not yet in service. This metric does not include instances that are in service or terminating.
Desired Capacity	Count	The number of instances that the Auto Scaling group attempts to maintain.

Maximum Group Size	Count	The maximum size of the Auto Scaling group.
In Service Instances	Count	The number of instances that are running as part of the Auto Scaling group. This metric does not include instances that are pending or terminating.
Minimum Group Size	Count	The minimum size of the Auto Scaling group.
The following metrics are available for instances:		
CPU Utilization	Percent	The percentage of allocated EC2 compute units that are currently in use on the instance.
Disk Read Operations	Bytes	Completed read operations from all instance store volumes available to the instance in a specified period of time.
Disk Write Operations	Bytes	Completed write operations to all instance store volumes available to the instance in a specified period of time.
Disk Reads	Bytes	Bytes read from all instance store volumes available to the instance.
Disk Writes	Bytes	Bytes written to all instance store volumes available to the instance.
Network In	Bytes	The number of bytes received on all network interfaces by the instance.
Network Out	Bytes	The number of bytes sent out on all network interfaces by the instance.
Status Check Failed (Any)	Count	Reports whether the instance has passed both the instance status check and the system status check in the last minute.
Status Check Failed (Instance)	Count	Reports whether the instance has passed the instance status check in the last minute.
Status Check Failed (System)	Count	Reports whether the instance has passed the system status check in the last minute.

Amazon ElastiCache

The **Amazon ElastiCache** integration allows you to monitor your Amazon ElastiCache service through FortiMonitor.

To add the Amazon ElastiCache service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **ElastiCache**.

Metrics

The following table describes the metrics supported by FortiMonitor for ElastiCache.

See https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/CacheMetrics.HostLevel.html for more information.

Metric	Unit	Description
CPU Utilization	Percent	The CPU utilization percentage.
Freeable Memory	Bytes	The amount of free memory available on the host.
Network Bytes In	Bytes	The number of bytes read by the host from the network.
Network Bytes Out	Bytes	The number of bytes written by the host to the network.
Swap Usage	Count	The number of swaps used on the host.

Amazon ElasticSearch

The **Amazon ElasticSearch Service (ES)** integration allows you to monitor your Amazon ElasticSearch service through FortiMonitor.

To add the Amazon ElasticSearch service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **ElasticSearch**.

Metrics

The following table describes the metrics supported by FortiMonitor for ElasticSearch.

See https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/es-managedomains-cloudwatchmetrics.html for more information.

Metric	Unit	Description
Cluster Status Green	Count	A value of 1 indicates that all index shards are allocated to nodes in the cluster.
Cluster Status Yellow	Count	A value of 1 indicates that the primary shards for all indices are allocated to nodes in the cluster, but replica shards for at least one index are not.
Cluster Status Red	Count	A value of 1 indicates that the primary and replica shards for at least one index are not allocated to nodes in the cluster.
Nodes	Count	The number of nodes in the Amazon ES cluster.
Searchable Documents	Count	The total number of searchable documents across all data nodes in the cluster.
Deleted Documents	Count	The total number of documents marked for deletion across all data nodes in the cluster.

CPU Utilization	Percentage	The percentage of CPU usage for data nodes in the cluster.
Free Storage Space	Count	The free space for data nodes in the cluster.
Cluster Used Space	Count	The total used space for the cluster.
Cluster Index Writes Blocked	Count	Indicates whether your cluster is accepting or blocking incoming write requests. A value of 0 means that the cluster is accepting requests. A value of 1 means that it is blocking requests.
JVM Memory Pressure	Count	The maximum percentage of the Java heap used for all data nodes in the cluster.
Automated Snapshot Failure	Count	The number of failed automated snapshots for the cluster.
CPU Credit Balance	Count	The remaining CPU credits available for data nodes in the cluster.
Kibana Healthy Nodes	Count	Kibana health check.
KMS Key Error	Count	A value of 1 indicates that the KMS customer master key used to encrypt data at rest has been disabled.
KMS Key Inaccesible	Count	A value of 1 indicates that the KMS customer master key used to encrypt data at rest has been deleted or revoked its grants to Amazon ES.
Invalid Host Header Requests	Count	The number of HTTP requests made to the Elasticsearch cluster that included an invalid (or missing) host header.
Elasticsearch Requests	Count	The number of requests made to the Elasticsearch cluster.
Master CPU Utilization	Percentage	The maximum percentage of CPU resources used by the dedicated master nodes.
Master JVM Memory Pressure	Percentage	The maximum percentage of the Java heap used for all dedicated master nodes in the cluster.
Master CPU CreditBalance	Count	The remaining CPU credits available for dedicated master nodes in the cluster.
Master Reachable From Node	Count	A health check for **MasterNotDiscovered** exceptions.
Read Latency	Seconds	The latency for read operations on EBS volumes.
Write Latency	Seconds	The latency for write operations on EBS volumes.
Read Throughput	Bytes/second	The throughput for read operations on EBS volumes.
Write Throughput	Bytes/second	The throughput for write operations on EBS volumes.
Disk Queue Depth	Count	The number of pending input and output (I/O) requests for an EBS volume.

Read IOPS	IO/second	The number of input and output (I/O) operations per second for read operations on EBS volumes.
Write IOPS	IO/second	The number of input and output (I/O) operations per second for write operations on EBS volumes.

AWS ELB

The **AWS Elastic Load Balancing (ELB)** integration allows you to monitor your AWS ELB service through FortiMonitor.

To add the AWS ELB service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **ELB**.

Metrics

The following table describes the metrics supported by FortiMonitor for ELB.

See https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-cloudwatch-metrics.html for more information.

Metric	Unit	Description
Backend Connection Errors	Count	The number of connections that were not successfully established between the load balancer and the registered instances.
Healthy Host Count	Count	The number of healthy instances registered with your load balancer.
HTTPCode ELB 4XX	Count	The number of HTTP 4XX client error codes generated by the load balancer.
HTTPCode ELB 5XX	Count	The number of HTTP 5XX server error codes generated by the load balancer.
Latency	Seconds	The total time elapsed, in seconds, from the time the load balancer sent the request to a registered instance until the instance started to send the response headers.
Request Count	Count	The number of requests completed or connections made during the specified interval. Options for Average, Maximum, Minimum and Sum are available. Note per AWS, all except Sum return a value of 1.
Spillover Count	Count	The total number of requests that were rejected because the surge queue is full.

Surge Queue Length	Count	The total number of requests (HTTP listener) or connections (TCP listener) that are pending routing to a healthy instance.
UnHealthy Host Count	Count	The number of unhealthy instances registered with your load balancer.
Estimated ALB Active Connection Count	Count	The estimated number of concurrent TCP connections active from clients to the load balancer and from the load balancer to targets.
Estimated ALB Consumed LCUs	Count	The estimated number of load balancer capacity units (LCU) used by an Application Load Balancer. You pay for the number of LCUs that you use per hour.
Estimated ALB New Connection Count	Count	The estimated number of new TCP connections established from clients to the load balancer and from the load balancer to targets.
Estimated Processed Bytes	Count	The estimated number of bytes processed by an Application Load Balancer.

AWS ELBv2

The **AWS Elastic Load Balancing v2 (ELBv2)** integration allows you to monitor your AWS ELBv2 service through FortiMonitor.

To add the AWS ELBv2 service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **ELBv2**.

Metrics

The following table describes the metrics supported by FortiMonitor for ELBv2.

See https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-cloudwatch-metrics.html for more information.

Metric	Unit	Description
Active Connection Count	Count	The total number of simultaneous TCP connections active from clients to the load balancer and from the load balancer to targets.
Client TLS Negotiation Error Count	Count	The number of TLS connections initiated by the client that did not establish a session with the load balancer
Active Flow Count	Count	The total number of concurrent flows (or connections) from clients to targets.

Consumed LCUs	Count	The number of load balancer capacity units (LCU) used by your load balancer.
Healthy Host Count	Count	The number of targets that are considered healthy.
New Flow Count	Count	The total number of new flows (or connections) established from clients to targets in the time period.
TCP Client Reset Count	Count	The total number of reset (RST) packets sent from a client to a target.
TCP ELB Reset Count	Count	The total number of reset (RST) packets generated by the load balancer.
TCP Target Reset Count	Count	The total number of reset (RST) packets sent from a target to a client.
HTTP Code ELB 4XX Count	Count	The number of HTTP 4xx error codes that originate from the load balancer.
HTTP Code ELB 5XX Count	Count	The number of HTTP 5xx error codes that originate from the load balancer.
HTTP Code Target 2XX Count	Count	The number of HTTP 2xx response codes generated by the targets.
HTTP Code Target 3XX Count	Count	The number of HTTP 3xx response codes generated by the targets.
HTTP Code Target 4XX Count	Count	The number of HTTP 4xx response codes generated by the targets.
HTTP Code Target 5XX Count	Count	The number of HTTP 5xx response codes generated by the targets.
IPv6 Processed Bytes	Bytes	The total number of bytes processed by the load balancer over IPv6.
IPv6 Request Count	Count	The number of IPv6 requests received by the load balancer.
New Connection Count	Count	The total number of new TCP connections established from clients to the load balancer and from the load balancer to targets.
Processed Bytes	Count	The total number of bytes processed by the load balancer over IPv4 and IPv6.
Rejected Connection Count	Count	The number of connections that were rejected because the load balancer had reached its maximum number of connections.
Request Count	Count	The number of requests processed over IPv4 and IPv6. Options for Average, Maximum, Minimum and Sum are available. Note per AWS, all except Sum return a value of 1.

Request Count Per Target	Count	The average number of requests received by each target in a target group.
Rule Evaluations	Count	The number of rules processed by the load balancer given a request rate averaged over an hour.
Target Connection Error Count	Count	The number of connections that were not successfully established between the load balancer and target.
Target Response Time	Seconds	The time elapsed, in seconds, after the request leaves the load balancer until a response from the target is received.
Target TLS Negotiation Error Count	Count	The number of TLS connections initiated by the load balancer that did not establish a session with the target.
UnHealthy Host Count	Count	The number of targets that are considered unhealthy.

AWS Lambda

The **AWS Lambda** integration allows you to monitor your AWS Lambda service through FortiMonitor.

To add the AWS Lambda service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **Lambda**.

Metrics

The following table describes the metrics supported by FortiMonitor for Lambda.

See https://docs.aws.amazon.com/lambda/latest/dg/monitoring-functions-metrics.html for more information.

Metric	Unit	Description
Invocations	Count	The number of times a function is invoked in response to an event or invocation API call.
Errors	Count	The number of invocations that failed due to errors in the function (response code 4XX).
Dead Letter Error	Count	Incremented when Lambda is unable to write the failed event payload to your configured Dead Letter Queues.
Duration	Milliseconds	The elapsed wall clock time from when the function code starts executing as a result of an invocation to when it stops executing.
Throttles	Count	The number of Lambda function invocation attempts that were throttled due to invocation rates exceeding the customer's concurrent limits (error code 429).

Iterator Age	Milliseconds	The age of the last record for each batch of records processed.

Amazon RDS

The **Amazon Relational Database Service (RDS)** integration allows you to monitor Amazon RDS through FortiMonitor.

To add the Amazon RDS service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **RDS**.

Metrics

The following table describes the metrics supported by FortiMonitor for RDS.

See https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/MonitoringOverview.html for more information.

Metric	Unit	Description
CPU Utilization	Percent	The percentage of CPU utilization.
Database Connections	Count	The number of database connections in use.
Disk Queue Depth	Count	The number of outstanding read/write requests waiting to access the disk.
Freebale Memory	Bytes	The amount of available random access memory.
Free Storage Space	Bytes	The amount of available storage space.
Replica Lag	Seconds	The amount of time a Read Replica DB instance lags behind the source DB instance.
Swap Usage	Bytes	The amount of swap space used on the DB instance.
Read IOPS	Count/Second	The average number of disk read I/O operations per second.
Write IOPS	Count/Second	The average number of disk write I/O operations per second.
Read Latency	Seconds	The average amount of time taken per disk I/O operation.
Write Latency	Seconds	The average amount of time taken per disk I/O operation.
Read Throughput	Bytes/Second	The average number of bytes read from disk per second.

Write Throughput	Bytes/Second	The average number of bytes written to disk per second.
Network Receive Throughput	Bytes/Second	The incoming (Receive) network traffic on the DB instance, including both customer database traffic and Amazon RDS traffic used for monitoring and replication.
Network Transmit Throughput	Bytes/Second	The outgoing (Transmit) network traffic on the DB instance, including both customer database traffic and Amazon RDS traffic used for monitoring and replication.

Amazon RDS Cluster

The **Amazon Relational Database Service (RDS) Cluster** integration allows you to monitor Amazon RDS Cluster through FortiMonitor.

To add the Amazon RDS Cluster service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **RDS Cluster**.

Metrics

The following table describes the metrics supported by FortiMonitor for RDS Cluster.

See https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Monitoring.html for more information.

Metric	Unit	Description
Queries	Count	The average number of queries per second.
Deadlocks	Count	The average number of deadlocks in the database per second.
Volume Bytes Used	Bytes	The amount of storage used by your database instance, in bytes.
CPU Utilization	Percent	The percentage of CPU used by a database instance.
CPU Credit Usage	Count	The number of CPU credits consumed during the specified period.
CPU Credit Balance	Count	The number of CPU credits that an instance has accumulated.
UpdateLatency	Milliseconds	The amount of latency in milliseconds for update queries.

DatabaseConnections	Count	The number of connections to a database instance.
Network Throughput	Bytes	The amount of network throughput both received from and transmitted to clients by each instance in database cluster, in bytes per second.
Update Throughput	Count	The average number of update queries per second.
Delete Throughput	Count	The average number of delete queries per second.
Blocked Transactions	Count	The average number of transactions in the database that are blocked per second.
Login Failures	Count	The average number of failed login attempts per second.
Commit Latency	Milliseconds	The amount of latency for commit operations, in milliseconds.
Commit Throughput	Count	The average number of commit operations per second.
Engine Uptime	Count	The amount of time that the instance has been running, in seconds.
InsertLatency	Milliseconds	The amount of latency for insert queries, in milliseconds.
Insert Throughput	Count	The average number of insert queries per second.
Active Transactions	Count	The average number of current transactions executing on a database instance per second.
Network Transmit Throughput	Bytes	The amount of network throughput sent to clients by each instance in the database cluster, in bytes per second.
Select Latency	Milliseconds	The amount of latency for select queries, in milliseconds.
Network Receive Throughput	Bytes	The amount of network throughput received from clients by each instance in the database cluster, in bytes per second.
Delete Latency	Milliseconds	The amount of latency for delete queries, in milliseconds.
Select Throughput	Count	The average number of select queries per second.
DDL Throughput	Count	The average number of data definition language (DDL) requests per second.

DML Throughput	Count	The average number of inserts, updates, and deletes per second.
DML Latency	Milliseconds	The amount of latency for inserts, updates, and deletes, in milliseconds.
DDL Latency	Milliseconds	The amount of latency for DDL requests, in milliseconds
Serverless Database Capacity	GiB	The capacity allocated to the database cluster.
Freeable Memory	Bytes	The amount of available random access memory, in bytes.
FreeLocal Storage	Bytes	The amount of storage available for temporary tables and logs, in bytes.
Snapshot Storage Used	GiB	The total amount of backup storage consumed by all snapshots for a database cluster outside its backup retention window.
Volume Read IOPs	Count	The number of billed read I/O operations from a cluster volume reported at 5-minute intervals.
Volume Write IOPs	Count	The number of write disk I/O operations to the cluster volume reported at 5-minute intervals.
Aurora Binlog Replica Lag	Seconds	The amount of time a replica database cluster running on Aurora with MySQL compatibility lags behind the source database cluster.
Buffer Cache Hit Ratio	Percent	The percentage of requests that are served by the buffer cache.
Result Set Cache Hit Ratio	Percent	The percentage of requests that are served by the Resultset cache.
Bin Log Disk Usage	Bytes	The amount of disk space occupied by binary logs on the master, in bytes.
TotalBackupStorageBilled	GiB	The total amount of backup storage in GiB for which you are billed for a given Aurora database cluster.
BackupRetentionPeriodStorageUsed	GiB	The total amount of backup storage in GiB used to support the point-in-time restore feature within the Aurora database cluster's backup retention window.

Amazon Redshift

The **Amazon Redshift** integration allows you to monitor your Amazon Redshift service through FortiMonitor.

To add the Amazon Redshift service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **Redshift**.

Metrics

The following table describes the metrics supported by FortiMonitor for Redshift.

See https://docs.aws.amazon.com/redshift/latest/mgmt/metrics-listing.html for more information.

Metric	Unit	Description
CPU Utilization	Percent	The percentage of CPU utilization.
Database Connections	Count	The number of database connections to a cluster.
Health Status	1/0 (HEALTHY/UNHEALTHY in the Amazon Redshift console)	Indicates the health of the cluster.
Maintenance Mode	1/0 (ON/OFF in the Amazon Redshift console)	Indicates whether the cluster is in maintenance mode.
Network Receive Throughput	Bytes/second	The rate at which the node or cluster receives data.
Network Transmit Throughput	Bytes/second	The rate at which the node or cluster writes data.
Percentage Disk Space Used	Percent	The percent of disk space used.
Read IOPS	Count/second	The average number of disk read operations per second.
Read Latency	Seconds	The average amount of time taken for disk read I/O operations.
Read Throughput	Bytes	The average number of bytes read from disk per second.
Write IOPS	Count/seconds	The average number of write operations per second.
Write Latency	Seconds	The average amount of time taken for disk write I/O operations.
Write Throughput	Bytes	The average number of bytes written to disk per second.

Amazon S3

The **Amazon Simple Storage Service (S3)** integration allows you to monitor your Amazon S3 service through FortiMonitor.

To monitor Amazon S3 instances, make sure that Detailed Monitoring is enabled.

To add the Amazon S3 service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **S3**.

Metrics

The following table describes the metrics supported by FortiMonitor for S3.

See https://docs.aws.amazon.com/AmazonS3/latest/dev/cloudwatch-monitoring.html for more information.

Metric	Unit	Description
Bucket Size Bytes	Bytes	The amount of data in bytes stored in a bucket in the STANDARD storage class, INTELLIGENT_TIERING storage class, Standard - Infrequent Access (STANDARD_IA) storage class, OneZone - Infrequent Access (ONEZONE_IA), Reduced Redundancy Storage (RRS) class, or Glacier (GLACIER) storage class.
Number Of Objects	Count	The total number of objects stored in a bucket for all storage classes except for the GLACIER storage class.
All Requests	Count	The total number of HTTP requests made to an Amazon S3 bucket.
Get Requests	Count	The number of HTTP GET requests made for objects in an Amazon S3 bucket.
Put Requests	Count	The number of HTTP PUT requests made for objects in an Amazon S3 bucket.
Delete Requests	Count	The number of HTTP DELETE requests made for objects in an Amazon S3 bucket.
Head Requests	Count	The number of HTTP HEAD requests made to an Amazon S3 bucket.
Post Requests	Count	The number of HTTP POST requests made to an Amazon S3 bucket.
List Requests	Count	The number of HTTP requests that list the contents of a bucket.
Bytes Downloaded	Bytes	The number of bytes downloaded for requests made to an Amazon S3 bucket, where the response includes a body.
Bytes Uploaded	Bytes	The number of bytes uploaded that contain a request body, made to an Amazon S3 bucket.
4xx Errors	Count	The number of HTTP 4xx client error status code requests made to an Amazon S3 bucket.
5xx Errors	Count	The number of HTTP 5xx server error status code requests made to an Amazon S3 bucket

First Byte Latency	Milliseconds	The per-request time from the complete request being received by an Amazon S3 bucket to when the response starts to be returned.
Total Request Latency	Milliseconds	The elapsed per-request time from the first byte received to the last byte sent to an Amazon S3 bucket.

Amazon SNS

The **Amazon Simple Notification Service (Amazon SNS)** integration allows you to monitor your Amazon SNS service through FortiMonitor.

To add the Amazon SNS service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **SNS**.

Metrics

The following table describes the metrics supported by FortiMonitor for SNS.

See https://docs.aws.amazon.com/sns/latest/dg/sns-monitoring-using-cloudwatch.html for more information.

Metric	Unit	Description
Number Of Messages Published	Count	The number of messages published to your Amazon SNS topics.
Publish Size	Bytes	The size of messages published.
Number Of Notifications Delivered	Count	The number of messages successfully delivered from your Amazon SNS topics to subscribing endpoints.
Number Of Notifications Failed	Count	The number of messages that Amazon SNS failed to deliver.
SMS Success Rate	Count	The rate of successful SMS message deliveries.

Amazon FSx

The **Amazon FSx** integration allows you to monitor your Amazon FSx service through FortiMonitor.

To add the Amazon FSx service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **FSX**.

Metrics

The following table describes the metrics supported by FortiMonitor for FSx.

See https://docs.aws.amazon.com/fsx/latest/WindowsGuide/monitoring-cloudwatch.html for more information.

Metric	Unit	Description
Free Storage Capacity	Bytes	The amount of available storage capacity.
Data Write Operations	Bytes	The number of write operations.
Data Read Operations	Bytes	The number of read operations.
Metadata Operations	Bytes	The number of metadata operations.
Data Write Bytes	Bytes	The number of bytes for file system write operations.
Data Read Bytes	Bytes	The number of bytes for file system read operations.

Amazon SQS

The **Amazon Simple Query Service (Amazon SQS)** integration allows you to monitor your Amazon SQS service through FortiMonitor.

To add the Amazon SQS service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **SQS**.

Metrics

The following table describes the metrics supported by FortiMonitor for SQS.

See https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-available-cloudwatch-metrics.html for more information.

Metric	Unit	Description
Approximate Age Of Oldest Message	Seconds	The approximate age of the oldest non-deleted message in the queue.
Approximate Number Of Messages Delayed	Count	The number of messages in the queue that are delayed and not available for reading immediately.
Approximate Number Of Messages Not Visible	Count	The number of messages that are in flight.
Approximate Number Of Messages Visible	Count	ApproximateNumberOfMessagesVisible
Number Of Empty Receives	Count	The number of ReceiveMessage API calls that did not return a message.

Number Of Messages Deleted	Count	The number of messages deleted from the queue.
Number Of Messages Received	Count	The number of messages returned by calls to the ReceiveMessage action.
Number Of Messages Sent	Count	The number of messages added to a queue.
Sent Message Size	Bytes	The size of messages added to a queue.

AWS VPN

The **AWS VPN** integration allows you to monitor VPN tunnels through FortiMonitor.

To add the AWS VPN service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and CloudWatch.
2. Under **Services**, select **VPN**.

Metrics

The following table describes the supported AWS VPN metrics.

See Monitoring VPN tunnels using Amazon CloudWatch for more information.

Metric	Unit	Description
Tunnel State	State	The state of the VPN tunnels. A value of 0 indicates that the VPN tunnel is DOWN and a value of 1 indicates that it is UP.
Tunnel Data In	Bytes	The number of bytes received via the VPN tunnel.
Tunnel Data Out	Bytes	The number of bytes sent via the VPN tunnel.

Azure

FortiMonitor offers native integration with Azure Monitor, enabling FortiMonitor to ingest your Azure monitoring data. FortiMonitor can also perform automatic discovery and monitoring of instances within your Azure account. This is configurable by service type and region, and can also be fully customized using your Azure tags.

Azure data should be used as an augmentation of, not a replacement for, the data obtained by the FortiMonitor server Agent and external monitoring. The server Agent can provide more detailed and accurate data across any OS distribution or application you may be running on your compute instance. As well, our external monitoring ensures you're getting the full picture of your current operating environment as well as a view into what your customers are experiencing.

Connecting FortiMonitor and Azure

Azure takes a few minutes to provision or spin up the new role/ credential set. Ensure that Azure has completed this process before trying to create the FortiMonitor integration. Creating the FortiMonitor integration before Azure is ready may cause credential errors.

To grant FortiMonitor access your Azure Monitor data, you'll need to create a service account within your Azure account that grants FortiMonitor read-only access.

1. From the navigation menu, click **Add**. The **Infrastructure and Resource Catalog** will be displayed**.**
2. Select **Microsoft Azure**.
3. Follow the on-page instructions to create an Azure service account.
4. Once you've obtained your **App ID** and **Password**, select **Verify Connection**.
5. Once your **App ID** and **Password** have been validated, you can configure your monitoring settings.
   - **Services**: Select the Azure services you'd like to monitor. It's better to only select the ones you're using, otherwise, it uses vital API calls.
   - **Filter Instances by Tag**: You can choose to only import instances that match the Azure tag filters you define.
   - **Regions**: Only select the regions you operate in, otherwise, it uses vital API calls
   - **Options - Import Tags**: Enabling tag import will pull in your Azure tags with your Azure instances. You have the option to import only the Value portion of the Azure tag or the entire Azure key-value pair as a single string tag.
   - **Options - Import Azure tags as FortiMonitor Attributes**: Azure tags will be imported as key-value pairs (attributes within FortiMonitor).
   - **Options - Routinely scan for new instances**: every 20 minutes, we'll look for new instances in your account and will begin monitoring them assuming they meet your filter criteria. Instances using the FortiMonitor Agent can be monitored immediately if you install the Agent on boot.
   - **Options - Apply Monitoring Policies**: Apply a monitoring policy to the imported Azure instances.
   - **Options - Destination Group**: any time instances are imported, they'll be placed in this group in the control panel. This is great for setting default values that are inherited from their parent group as well as apply default templates.
   - **Options - Template**: apply a Template to every instance that's imported
6. Click **Complete Integration**. We'll start pulling in your instances that meet your filter criteria and begin monitoring them.

API Limits and Throttling

By default, each Azure account gets 1M Azure Monitor API calls per month for free. When FortiMonitor makes Azure calls to obtain metrics (every 10 minutes), it utilizes your API calls quota. We encourage you to utilize the FortiMonitor Agent on Virtual Machine instances, not only for the cost savings but also for the increased functionality and granularity. You can read more about it here.

Once you exceed 1M Azure Monitor calls for the month, Microsoft will charge your account $10 per 1M calls. You can read more about their pricing here.

In certain large-scale scenarios, Azure could begin throttling API calls. We will begin backing-off at that time. If you expect to utilize close to or the full 1M calls per month, we recommend reaching out to Azure to ask for a limit increase. If you'd like FortiMonitor to collect Azure metrics more often than every 10 minutes, please contact our support team. As well, you can override this at the metric level by editing the metric. Check out Templates to do this in bulk.

Existing Monitoring

If you're running the Agent (Linux version > 2017.40, Windows version > 18.34), Virtual Machine metrics will be automatically added to your existing Agent-based instances.

Example: if you have a Linux Virtual Machine instance you're already monitoring with the Agent, and the Agent version is > 2017.40, we won't create a second "Virtual Machine" instance with the Azure Monitor connection - the new Azure Monitor metrics will be added to your existing instance

Virtual Machine Incident Confirmation

If you're monitoring a Virtual Machine instance with external checks - such as HTTP, HTTPS, or Ping - and we identify an incident, we'll first confirm with Azure that the instance is still around. If it was gracefully removed, we will not alert. If the instance was not removed gracefully, we will alert as normal.

Azure integrations

The following Azure integrations are supported (free or charged per instance):

Free

- App Service
- Functions Apps
- Storage Account
- Storage: File
- Storage: Queue
- Storage: Table
- Storage: Blob
- ServiceBus

Paid

- Virtual Machines
- SQL Databases
- Datawarehouse
- MySQL Databases
- PostgreSQL Databases
- Redis Cache

Azure App Service

The **Azure App Service** integration allows you to monitor your Azure App service through FortiMonitor.

To add the service to your FortiMonitor account:

1. Follow the instructions provided in Connecting FortiMonitor and Azure

Metrics

The following table describes the metrics supported by FortiMonitor for EC2.

See https://docs.microsoft.com/en-us/azure/app-service/ for more information.

Metric	Unit	Description
Bytes Received	Bytes	The amount of incoming bandwidth consumed by the app.
Bytes Sent	Bytes	The amount of outgoing bandwidth consumed by the app.
Http Server Errors	Count	The number of HTTP server errors.
Memory Working Set	Bytes	The current amount of memory used by the app.
Average Memory Working Set	Bytes	The average amount of memory used by the app.
CPU Time	Seconds	The amount of CPU consumed by the app. For more information about this metric, see https://docs.microsoft.com/en-us/azure/app-service/web-sites-monitor#cpu-time-vs-cpu-percentage .
Requests	Count	The total number of requests.
Http 101	Count	The number of requests resulting in HTTP 101 status code.
Http 2xx	Count	The number of requests resulting in an HTTP status code ≥ 200 but < 300.
Http 3xx	Count	The number of requests resulting in an HTTP status code ≥ 300 but < 400.
Http 401	Count	The number of requests resulting in HTTP 401 status code.
Http 403	Count	The number of requests resulting in HTTP 403 status code.
Http 404	Count	The number of requests resulting in HTTP 404 status code.
Http 406	Count	The number of requests resulting in HTTP 406 status code.
Http 4xx	4xxs	The number of requests resulting in an HTTP status code ≥ 400 but < 500.
Http 5xx	5xxs	The number of requests resulting in an HTTP status code ≥ 500 but < 600.
Average Response Time	sec	The average time it takes for the app to respond to requests.

Connections	Count	The number of bound sockets existing in the sandbox.
Handle Count	Count	The total number of handles currently open by the app process.
Thread Count	Count	The number of threads currently active in the app process.

Google Cloud Platform

FortiMonitor offers native integration with Google Cloud Platform (GCP), enabling FortiMonitor to ingest your GCP monitoring data. Also, FortiMonitor can perform automatic discovery and monitoring of instances within your GCP account. This is configurable by service type and region, and can also be fully customized using your GCP tags.

GCP data should be used as an augmentation of, not a replacement for, the data obtained by the FortiMonitor server agent and external monitoring. The server agent can provide more detailed and accurate data across any OS distribution or application you may be running on your compute instance. Also, our external monitoring ensures you are getting the full picture of your current operating environment as well as a view into what your customers are experiencing.

Connecting FortiMonitor and GCP

To grant FortiMonitor access to your GCP data, you will need to create an external account role within your GCP account that is tied to FortiMonitor's External GCP Account.

1. From the navigation menu, click **Add**. The **Infrastructure and Resource Catalog** will be displayed**.**
2. Select **Google Cloud Platform**.
3. Enter an a name for the integration.
4. Follow the instructions in the Control Panel.
5. Once you've obtained your Service Account key, select **Verify Connection**.
6. Once your key has been validated, you can configure your monitoring settings.
   - **Services:** Select the GCP services you'd like to monitor. It's better to only select the ones you're using, otherwise, it uses vital API calls.
   - **Filter Instances by Tag:** You can choose to only import instances that match the GCP tag filters you define.
   - **Regions:** Only select the regions you operate in, otherwise, it uses vital API calls
   - **Options - Import Tags:** Enabling tag import will pull in your GCP tags with your GCP instances. You have the option to import only the Value portion of the GCP tag or the entire GCP key-value pair as a single string tag.
   - **Options - Import GCP tags as FortiMonitor Attributes**: GCP tags will be imported as key-value pairs (attributes within FortiMonitor).
   - **Options - Routinely scan for new instances:** every 20 minutes, we'll look for new instances in your account and will begin monitoring them assuming they meet your filter criteria. Instances using the FortiMonitor agent can be monitored immediately if you install the agent on boot.
   - **Options - Apply Monitoring Policies**: Apply a monitoring policy to the imported GCP instances.
   - **Options - Destination Group:** any time instances are imported, they'll be placed in this group in the control panel. This is great for setting default values which are inherited from their parent group as well as apply default

templates.

- **Options - Template:** apply a Template to every instance that's imported

7. Click **Complete Integration**. We'll start pulling in your instances that meet your filter criteria and begin monitoring them.

GCP integrations

The following GCP integrations are supported:

- GCP Compute Engine
- GCP Router
- GCP Cloud Load Balancing

Kubernetes

Prerequisites

- A Kubernetes cluster configured with kubectl
- Helm installed locally

Resource requirements

This chart comes with three default deployment sizes: **small**, **medium**, and **large**
You can specify any upon installation using for example `--set size=large`
You can also override any individual request using `--set onsightRequests.<resource>=<value>` where **<resource>** is **cpu**, **memory**, or **ephemeral**

Size	Suggested # of Pods	Requested OnSight CPUs	Requested OnSight Memory	Requested OnSight Ephemeral Storage
small	< 500	1.0	1Gi	10Gi
medium	500-2000	2.0	2Gi	20Gi
large	>2000	3.0	3Gi	50Gi

Deploying FortiMonitor

See above for determining **size**. The default is **medium**.

If your cluster already has **metrics-server** installed, disable it as described below.

1. Add this Helm repo using `helm repo add panopta https://panopta.github.io/kubernetes/repo`
2. Install FortiMonitor using `helm install --set customer_key=YOUR-CUSTOMER-KEY <name-of-release> panopta/panopta`

In a few minutes, your cluster should show up in the FortiMonitor control panel.

A Note on metrics-server

The chart will install metrics-server by default. If you already have metrics-server installed in your cluster, you can skip it with `--set metricsServer.install=false`

Advanced configuration

If you wish to further customize your FortiMonitor deployment, you can pass additional options to the install command by adding one-to-many`--set <key>=<value>` to the install command.
Available configuration options are listed in the following table.
You can also specify such options in a YAML-formatted **values.yaml** file which you can then pass along to the install command with `-f values.yaml`

Configuration Options

Key Name	Default	Description
customer_key	None (Required)	Your FortiMonitor customer key
size	medium	Size of the cluster you are deploying to
clusterName	Kubernetes Cluster	The name of this cluster as it will show up in the control panel
metricsServer.install	true	Whether to install metrics-server as part of the deployment. Set to **false** if it's already installed.
topNNamespaces	0	The number of namespaces to pull in, ordered by the number of pods. 0 to include all.
onsightRequests.cpu	3.0	Requested CPU for the FortiMonitor OnSight
onsightRequests.memory	3Gi	Requested Memory for the FortiMonitor OnSight
onsightRequests.ephemeral	None	Requested Ephemeral Storage for the FortiMonitor OnSight
agent_config	None	Any additional blocks of configuration to deploy onto the nodes' agents

Upgrading FortiMonitor

1. Fetch the new chart using `helm repo update`
2. Upgrade your deployment using `helm upgrade <deployment name> panopta/panopta`

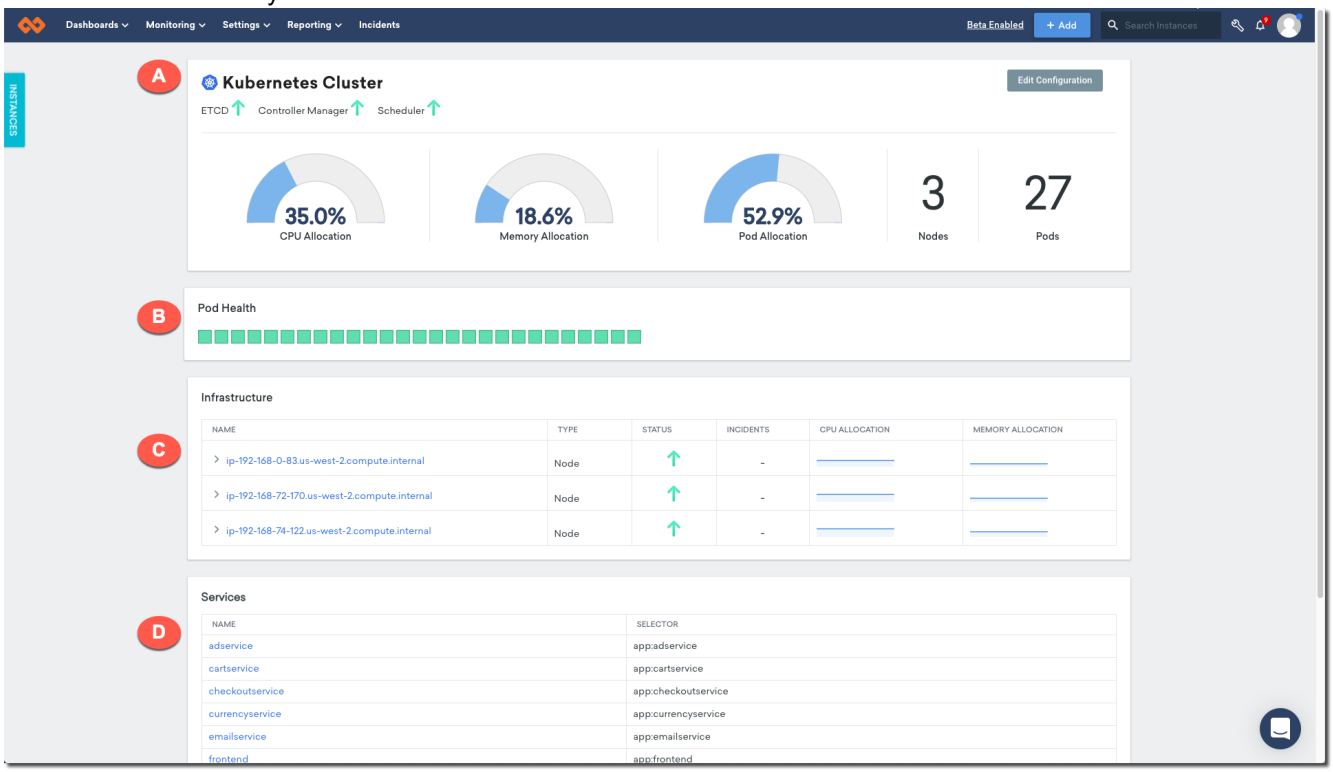
Uninstalling FortiMonitor

Run `helm uninstall <release_name>`

You can find the name of the release with `helm ls`

Kubernetes Details page

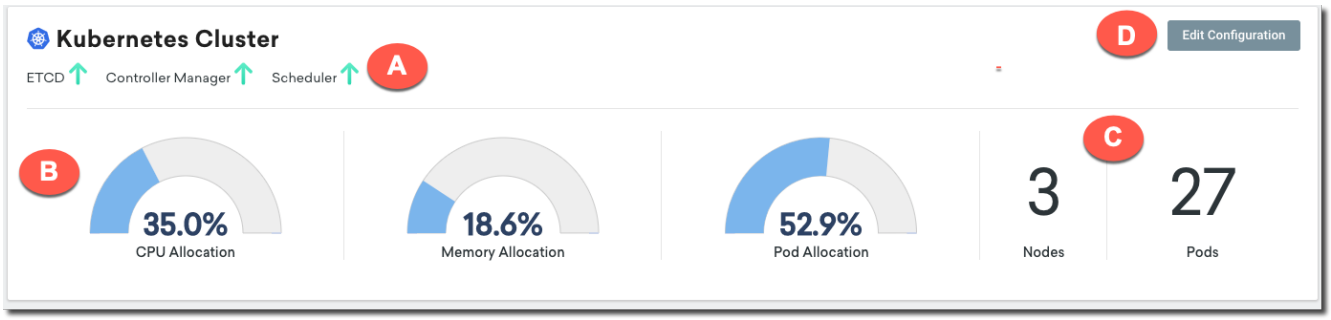
The **Kubernetes Details** page is located under the **Incoming OnSights** group in the **Instance Tree**. Select the Kubernetes cluster that you want to view.



Each of the sections on this page are described in detail in the following sections:

Kubernetes cluster

The Kubernetes cluster section of the details page shows the status of the components in the Kubernetes cluster. Gauges to measure resource allocation and the number of nodes and pods are also displayed. The section is described in the following table.



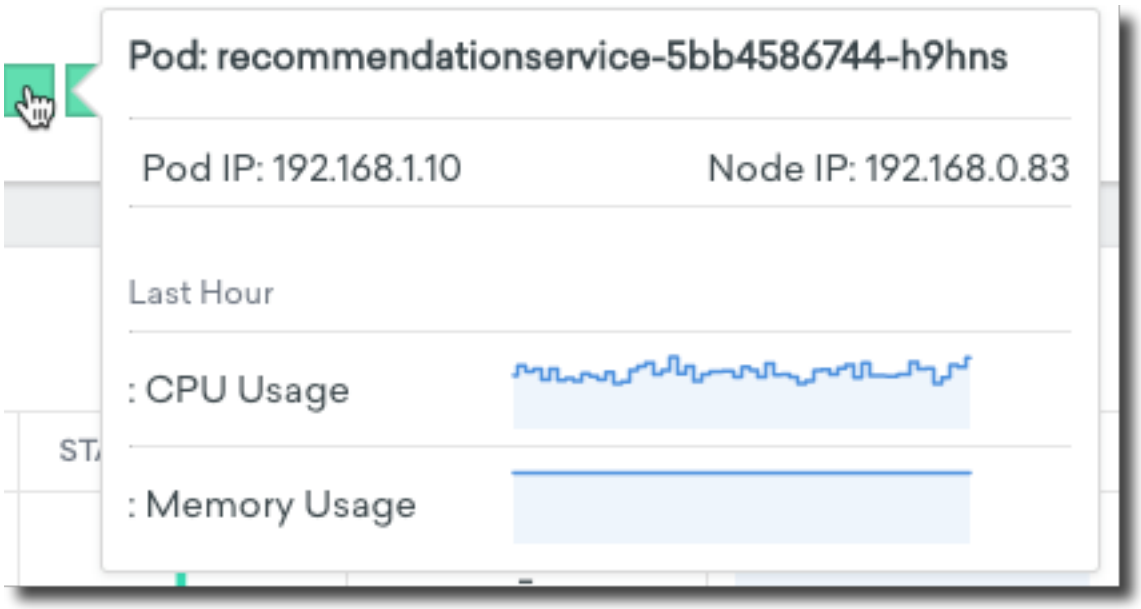
A	**Component status**
	Shows the status (up or down) of the following Kubernetes components: • ETCD (etcd) • Controller Manager (kube-controller-manager) • Scheduler (kube-scheduler)
B	**Gauges**
	The aggregated value of CPU, memory, and pods allocated to the cluster.
C	**Nodes and Pods**
	The total number of nodes and pods in the cluster.
D	**Edit Configuration**
	Select **Edit Configuration** to open the **Cluster Configuration** drawer. The drawer contains the following tabs: • **Settings** – You can rename the cluster from this tab. • **Delete Cluster** – This removes the Kubernetes cluster from FortiMonitor. • **Thresholds** – You can set thresholds and assign Alert Timelines to each of the services in the cluster.

Pod health

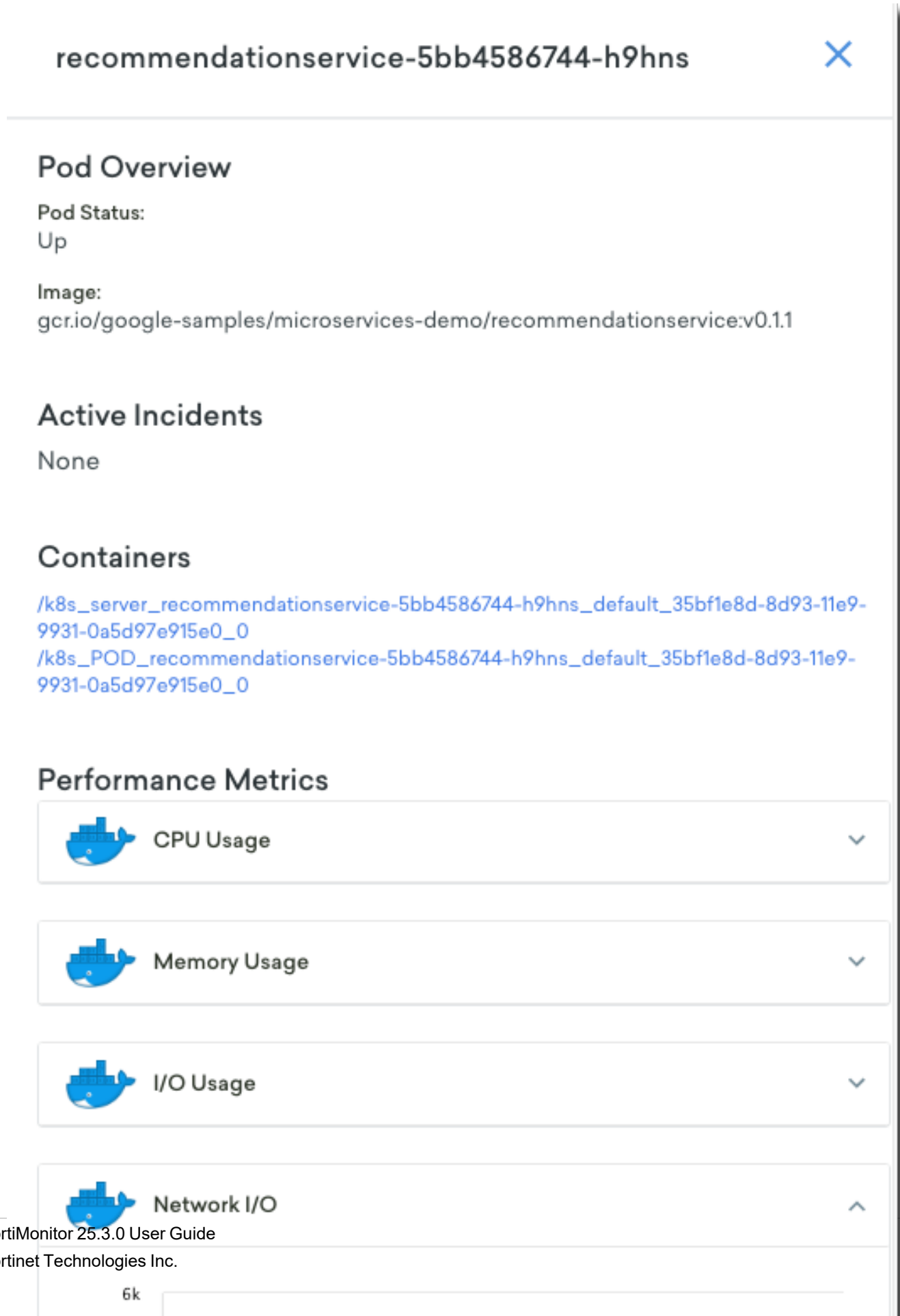
The Pod health section provides a high-level view of the health of the Kubernetes pods in your infrastructure. Pods are represented by **green** squares if there are no incident detected. Otherwise, the square becomes **red**.



You can hover your mouse to a pod to show the pod's details.



When you select a pod, a drawer detailing the pod's active incidents, containers, and performance metrics slides out.


recommendationservice-5bb4586744-h9hns
Pod Overview
Pod Status:
Up
Image:
gcr.io/google-samples/microservices-demo/recommendationservice:v0.1.1
Active Incidents
None
Containers
/k8s_server_recommendationservice-5bb4586744-h9hns_default_35bf1e8d-8d93-11e9-9931-0a5d97e915e0_0
/k8s_POD_recommendationservice-5bb4586744-h9hns_default_35bf1e8d-8d93-11e9-9931-0a5d97e915e0_0
Performance Metrics
CPU Usage
Memory Usage
I/O Usage
Network I/O
6k

Kubernetes infrastructure

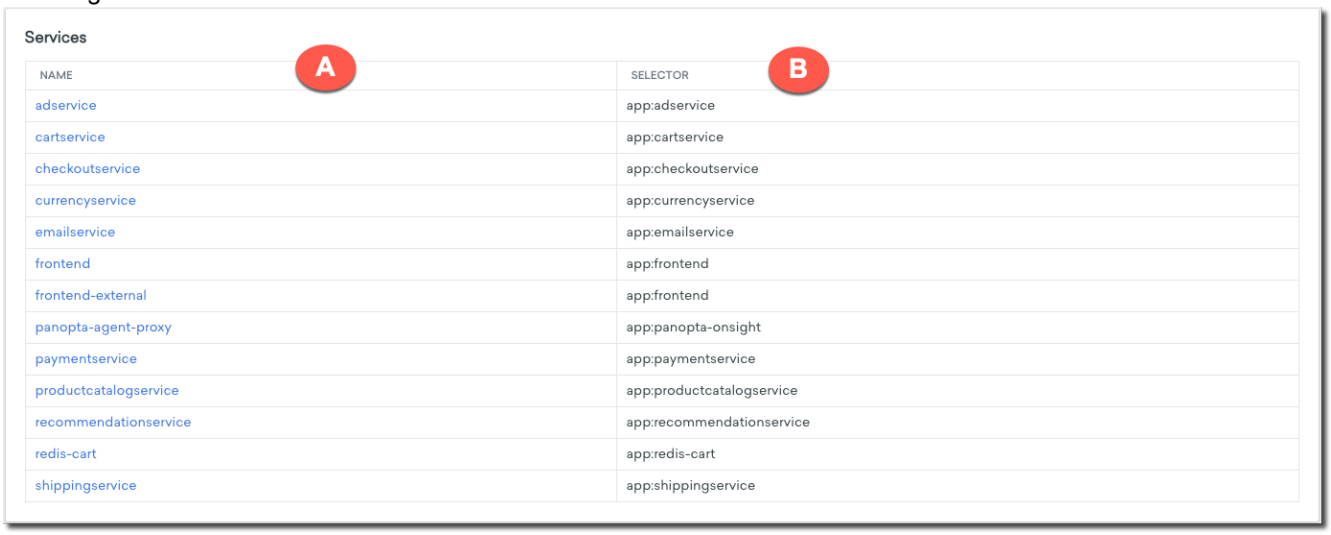
The Kubernetes infrastructure section lists all the nodes, pods, and containers in your infrastructure. The section is described in the following table.



A	**Name**
	When you click on the name, a Node or Pod Overview drawer slides out. You can also expand the list to show all of the pods under a node by selecting the node's expand arrow. Select the expand arrow on a pod to show all of the containers under that pod.
B	**Type**
	The type of the Kubernetes component (Node, Pod, Container).
C	**Status**
	A **green** up arrow is displayed if there are no incidents detected in any of the pods or containers under the node. If an incident is detected, a **red** down arrow is displayed.
D	**Incidents**
	The aggregated number of active incidents in this node, pod, or container.
E	**CPU and memory allocation**
	The line graph representation of the aggregated value of CPU and memory resources allocated to the node, pod, or container.

Kubernetes services

The Kubernetes services page lists all of the services running in the Kubernetes cluster. The section is detailed in the following table.



A	**Services**
	A service is a group of pods running on the cluster.
B	**Selector**
	The label selector used to define the service.

Application monitoring

The **Application Monitoring** solution can ensure that applications and services are available, functional, and are meeting expectations from a customer's perspective in real time. FortiMonitor allows you to capture application performance data, synthetic monitoring, and network checks to understand a customer's digital experience and support your organization's key performance indicators.

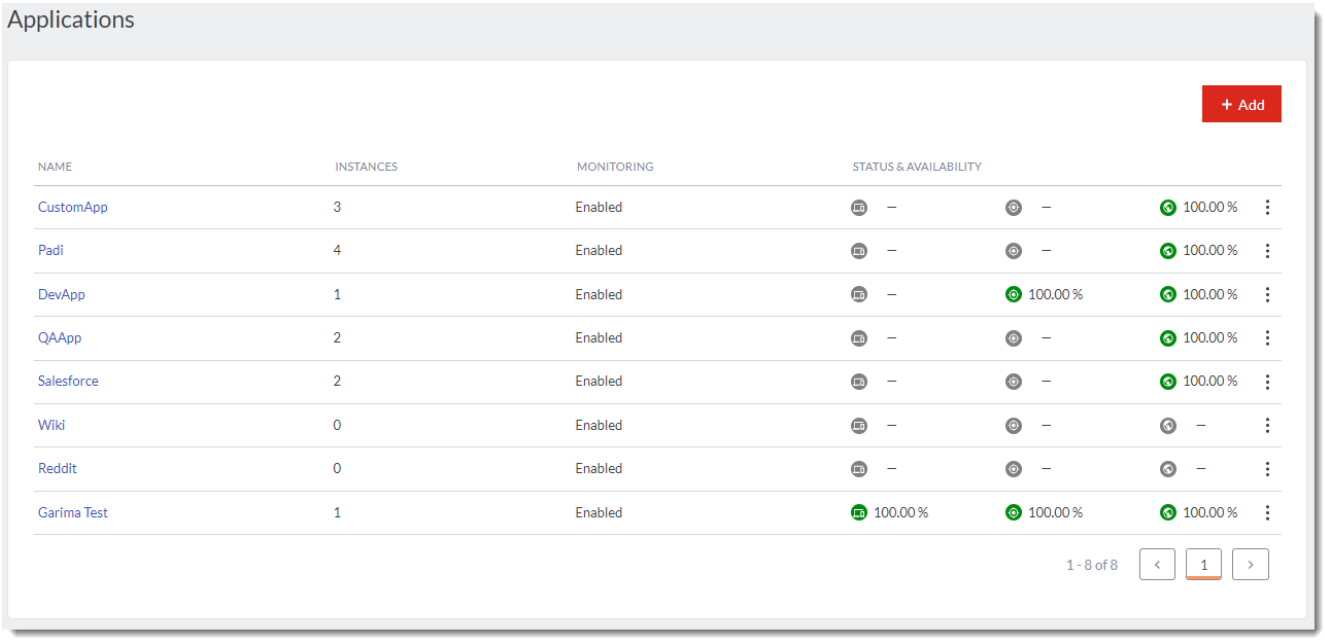
Application monitoring also allows you to collect, view, and analyze several metrics to ensure that your customer's experience is seamless and as intended.

An **application** can be defined as a collection of target instances which can be monitored from a number of monitoring sources, allowing you to measure performance from different vantage points:

- Public probe - Global network of monitoring nodes designed to track the availability of your servers 24 hours a day from outside your network.
- OnSight - The OnSight vCollector sits behind your firewall and functions similarly to our public monitoring nodes, which runs uptime checks and synthetic monitoring against your publicly available servers and web endpoints. The OnSight can be used to verify performance from core office or data center environments.
- Endpoint Agents (DEM Agent) - Collects key OS health and application performance metrics from remote locations like employee computers, retail locations, or public probes for external verification.
- SD-WAN device - Provides accurate user-level performance statistics to gain insight into the SD-WAN traffic performance.

The **Application Monitoring** page shows a focused view of the application and it instances. This page displays the following information:

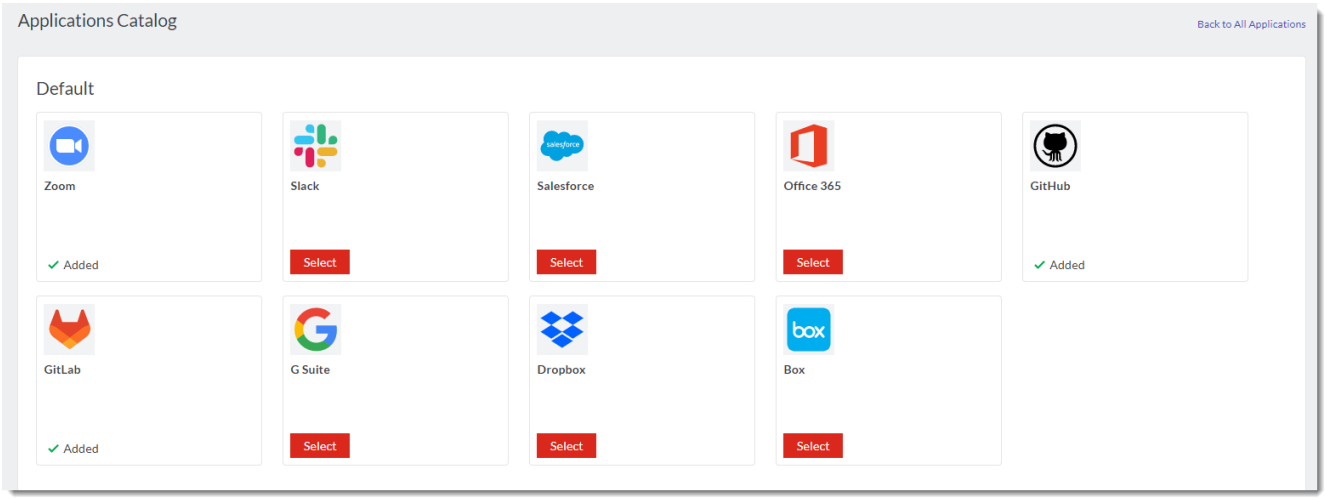
- Applications you have created
- The number of instances associated with each application
- Monitoring status (Enabled/ Disabled)
- Status and availability



The page is described in detail in Get started with Application monitoring.

Custom and default applications

When adding applications to FortiMonitor, you have the option to create custom applications, or use default applications. Default applications are pre-configured checks for common applications such as Salesforce or Zoom. With default applications, instances and checks are configured out-of-the box to satisfy the monitoring requirements of an application.



As of writing, the following default applications are available:

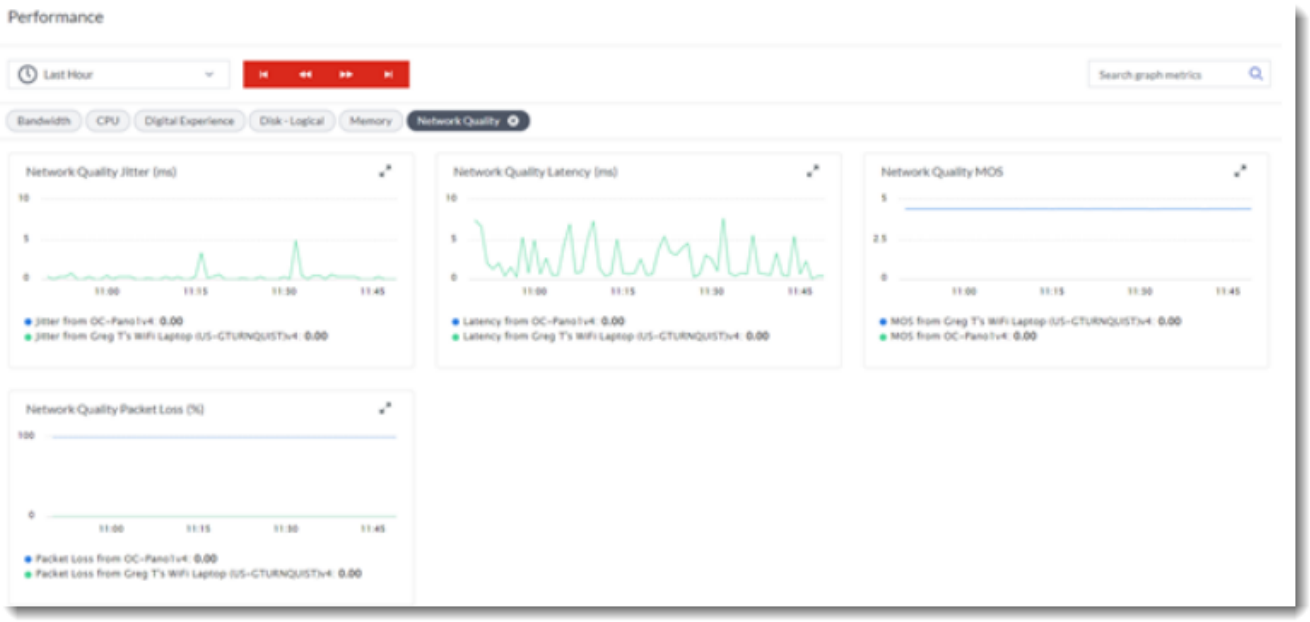
- Zoom
- Slack
- Salesforce
- Office 365
- GitHub
- GitLab
- G Suite
- Dropbox
- Box

Application monitoring usage examples

The following sections provide examples and use cases on how **Application Monitoring** can help with your customer's digital experience:

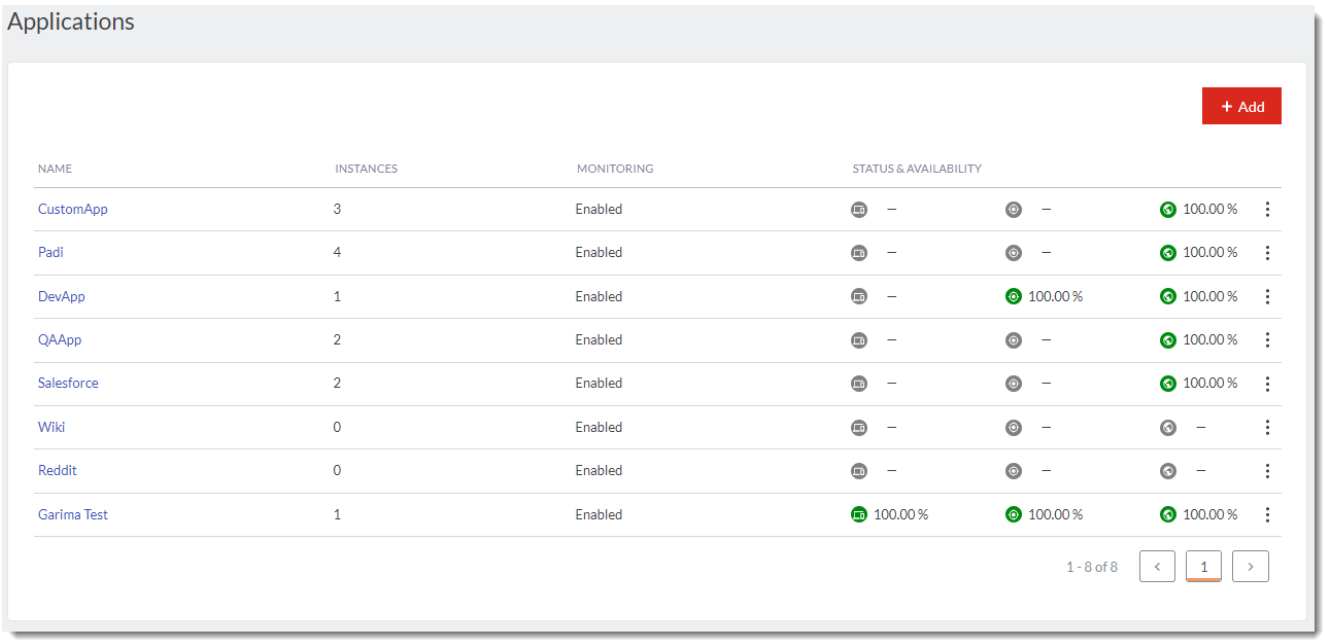
Overlay/underlay network quality

Application Monitoring can be used to collect detailed information about the network quality of active and backup links. This ability to measure the quality of backup links allows you to fix problems before there is an attempt to switch over to a backup link that is having network quality issues.

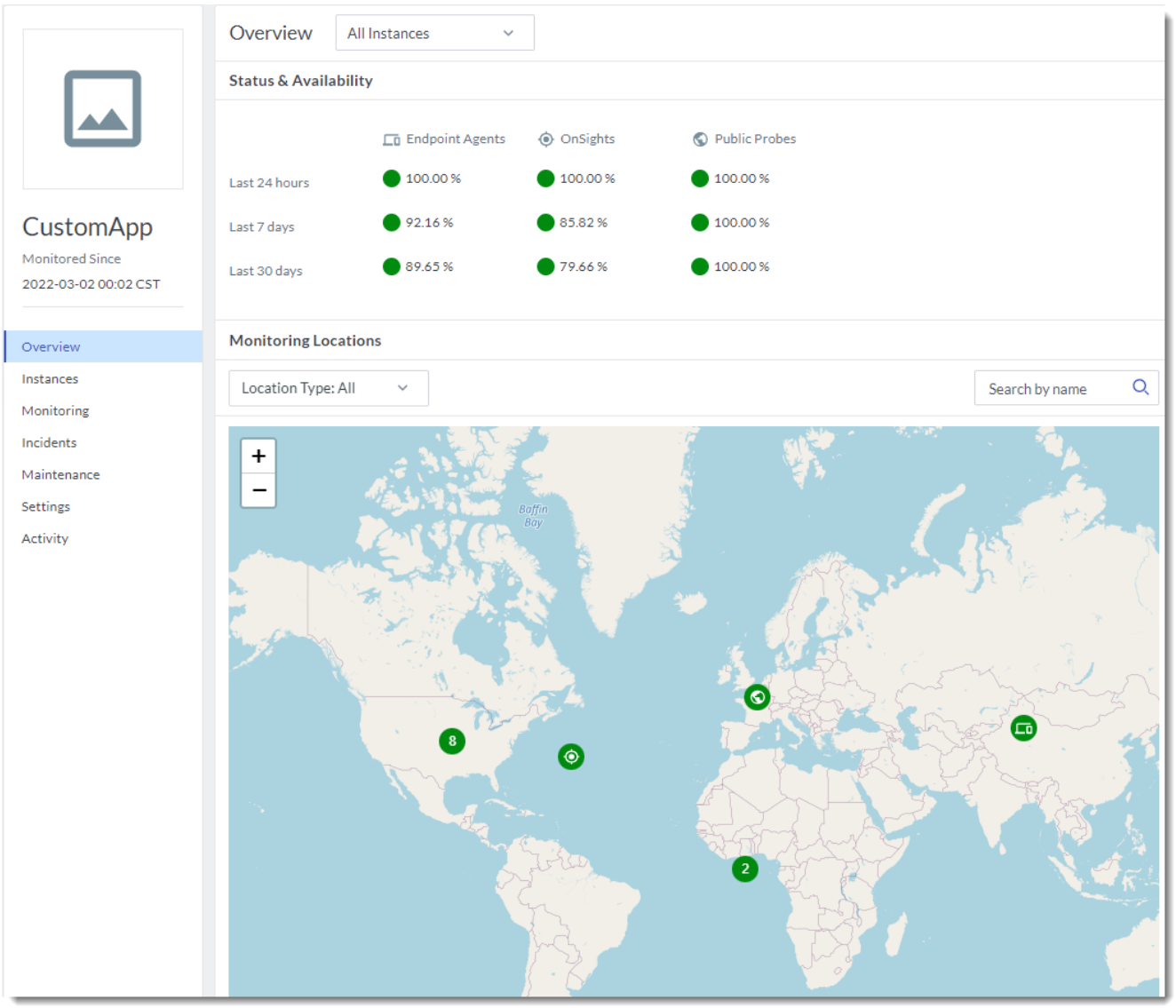


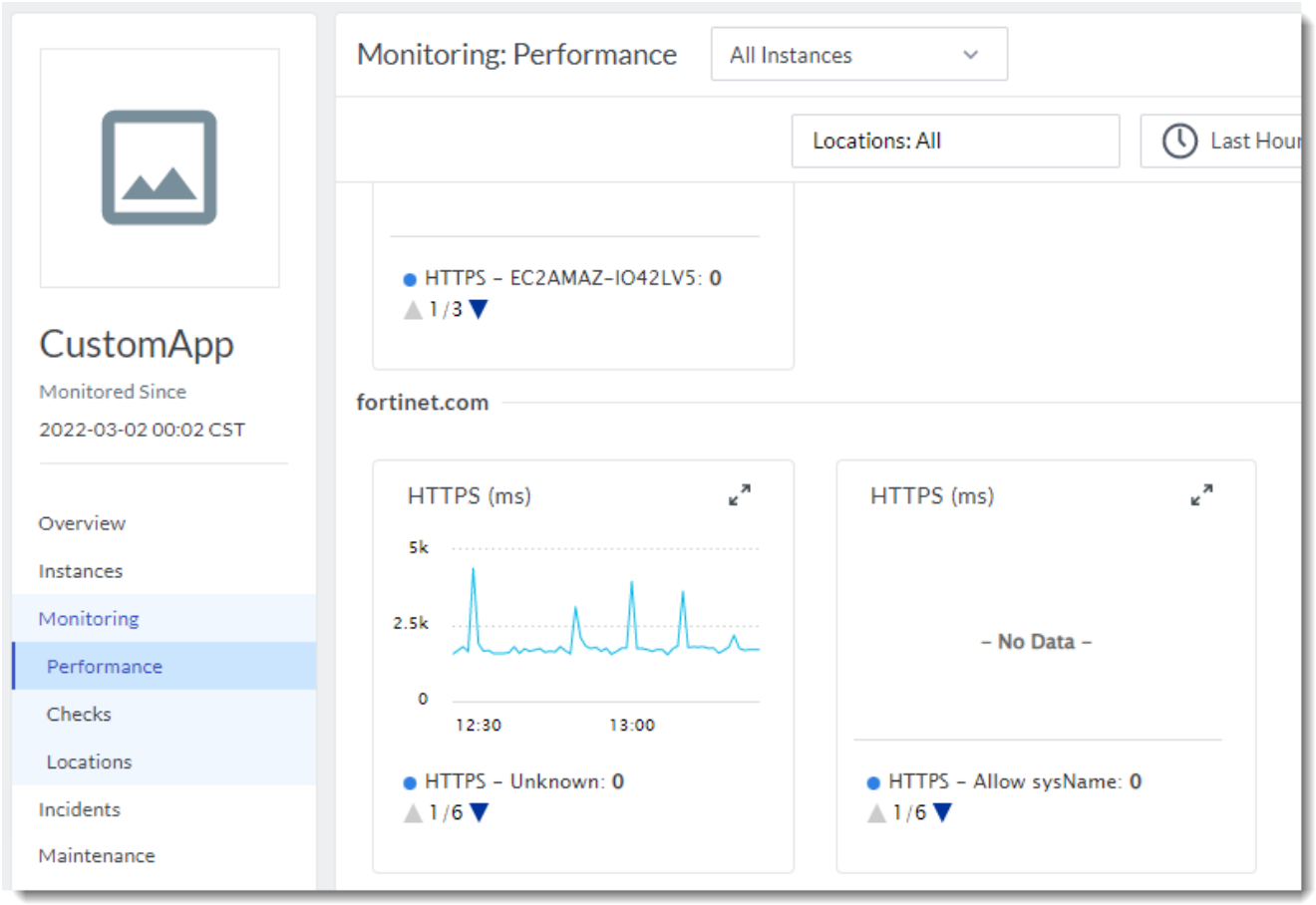
Measure application performance from multiple different vantage points

Application Monitoring can be used to report on the reachability and performance of important applications like Zoom and Teams from multiple different perspectives – endpoint, data center, and public test locations.



For any given application, you can quickly drill down and see what the performance and availability looks like from multiple different perspectives, including actionable information that allows you to quickly isolate the issue.


CustomApp
Monitored Since
2022-03-02 00:02 CST
Overview
Instances
Monitoring
Incidents
Maintenance
Settings
Activity
Overview
All Instances
Status & Availability
Endpoint Agents
OnSights
Public Probes
Last 24 hours
100.00 %
100.00 %
100.00 %
Last 7 days
92.16 %
85.82 %
100.00 %
Last 30 days
89.65 %
79.66 %
100.00 %
Monitoring Locations
Location Type: All
Search by name
Baffin Bay
8
2

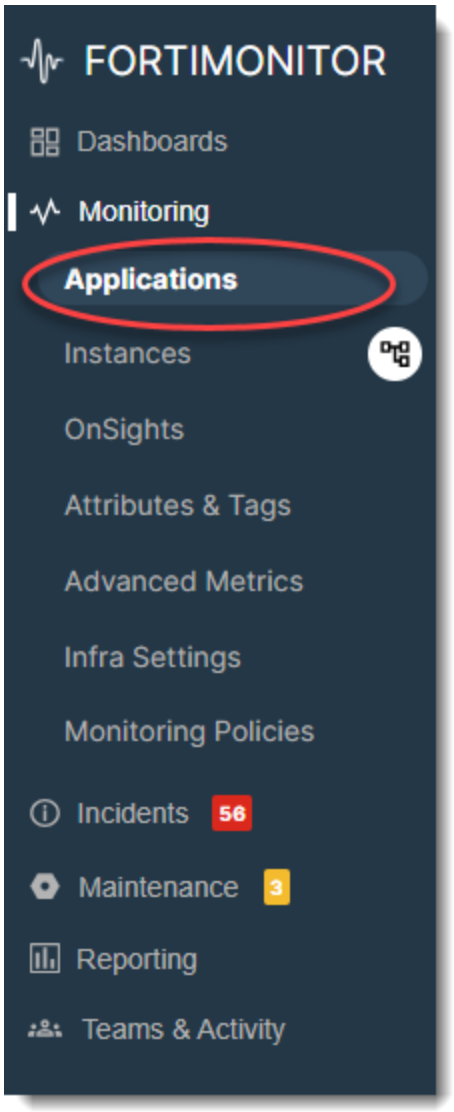


Get started with Application Monitoring

Application Monitoring page

The **Application Monitoring** page shows a focused view of an application and the number instances associated with it, providing a quick view on the application's status and availability.

To go to the **Application Monitoring** page, click **Monitoring** > **Applications** from the navigation menu.



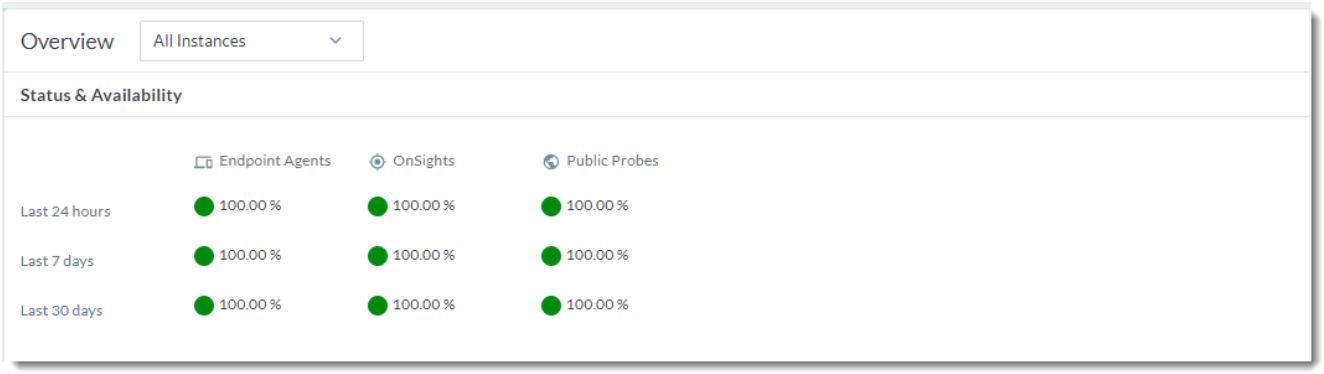
The **Application Monitoring** page is displayed.

To add an application, see Add an Application.

To manage the application and view its details, click the application's name. The application's page is displayed. The different sections of the page are detailed in the following sections.

Status and Availability

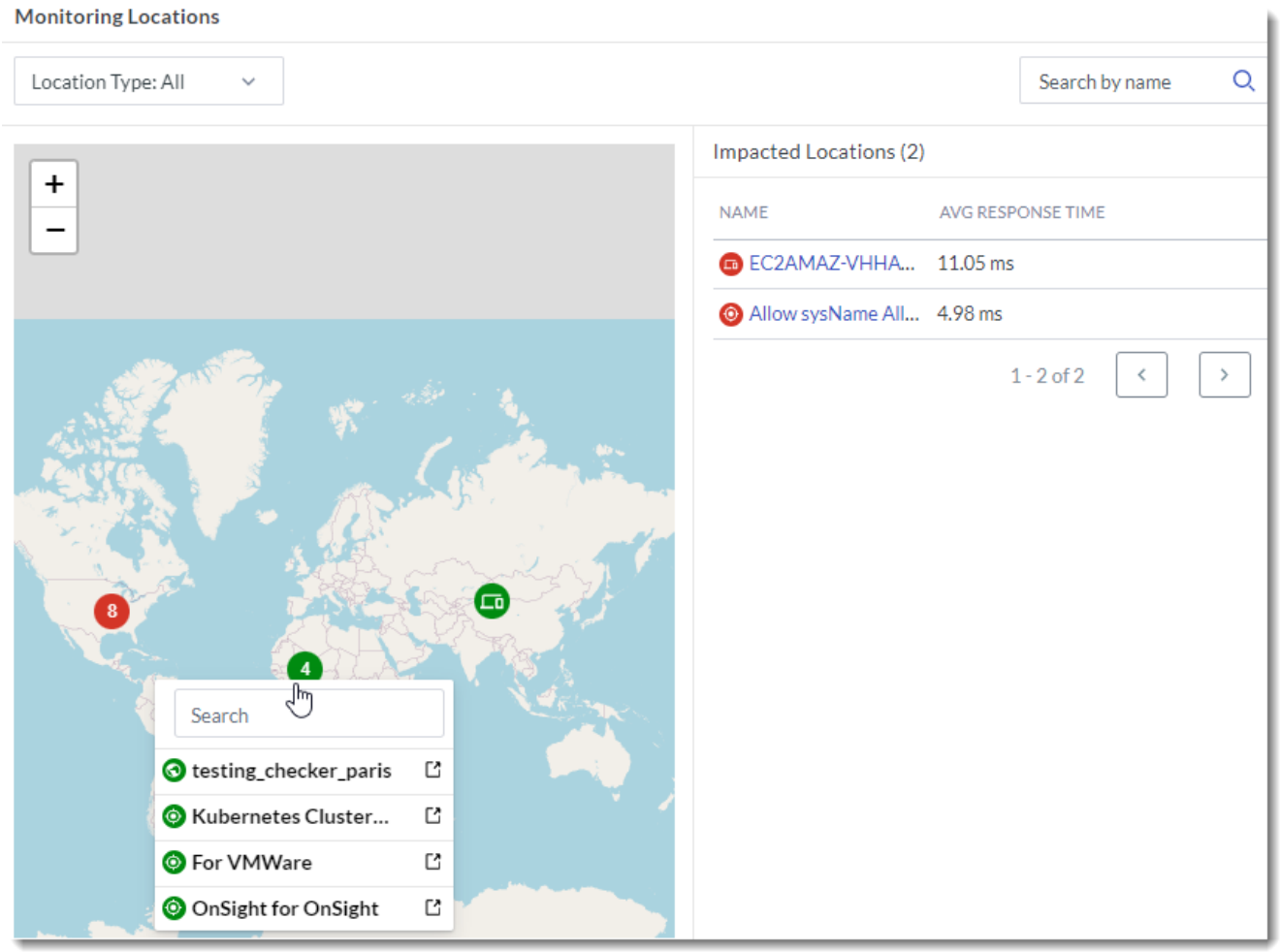
The status and health of the instances are shown in this section. The color of the marker represents the health or availability of the instance. Checks are run from every location (endpoint agents, OnSights, and public probes) at the same time.



Monitoring locations

Each geographic location is represented by a color-coded icon that shows the number of monitoring locations in that particular region. The color of the marker is based on the overall health of the monitoring locations in the region. A red icon signifies that there is one or more active incident monitored by a location.

On hover, the icon opens a panel that displays a list of monitoring locations in the region and their individual health. Click a location to open its Instance Details page.



Impacted Locations

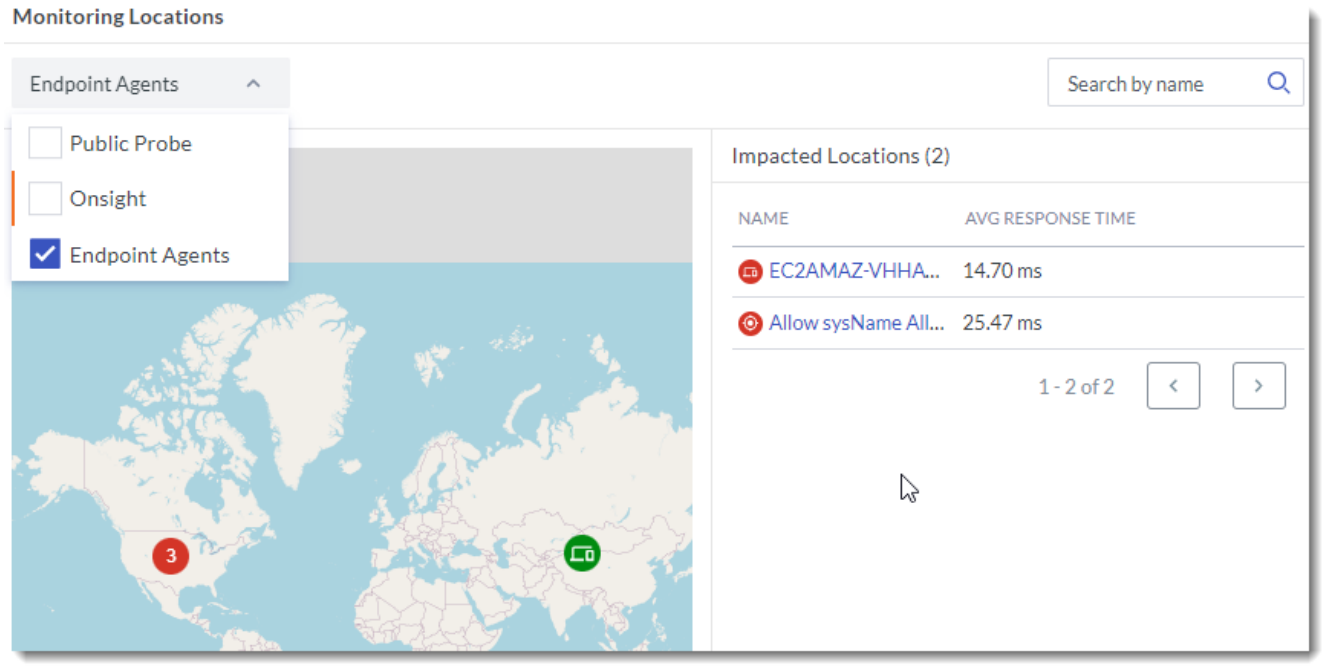
This section shows monitoring locations that have active incidents and their individual average response times. Click a monitoring location to learn more about the active incident.

Location Types

Application Monitoring supports three location types:

- Public probe - Global network of monitoring nodes designed to track the availability of your servers 24 hours a day from outside your network.
- OnSight - The OnSight vCollector sits behind your firewall and functions similarly to our public monitoring nodes, which runs uptime checks and synthetic monitoring against your publicly available servers and web endpoints. The OnSight can be used to verify performance from core office or data center environments.
- Endpoint Agents (DEM Agent) - Collects key OS health and application performance metrics from remote locations like employee computers, retail locations, or public probes for external verification.

The map can also be filtered by **Location Type** .

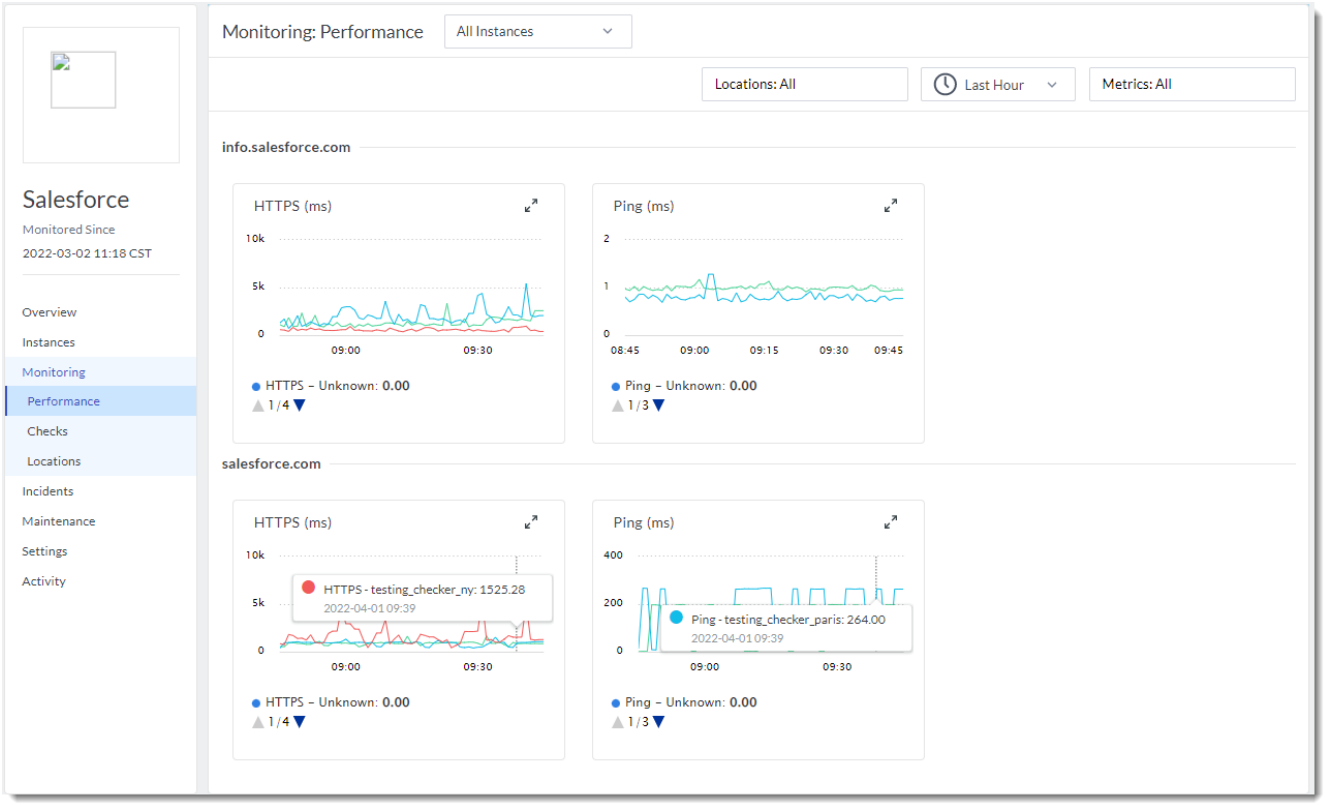


Performance monitoring

Application Monitoring supports server health, performance, and DEM (Digital Experience Monitoring) metrics to demonstrate the user experience from different monitoring locations.

In addition to standard server health metrics, the following metrics are also supported:

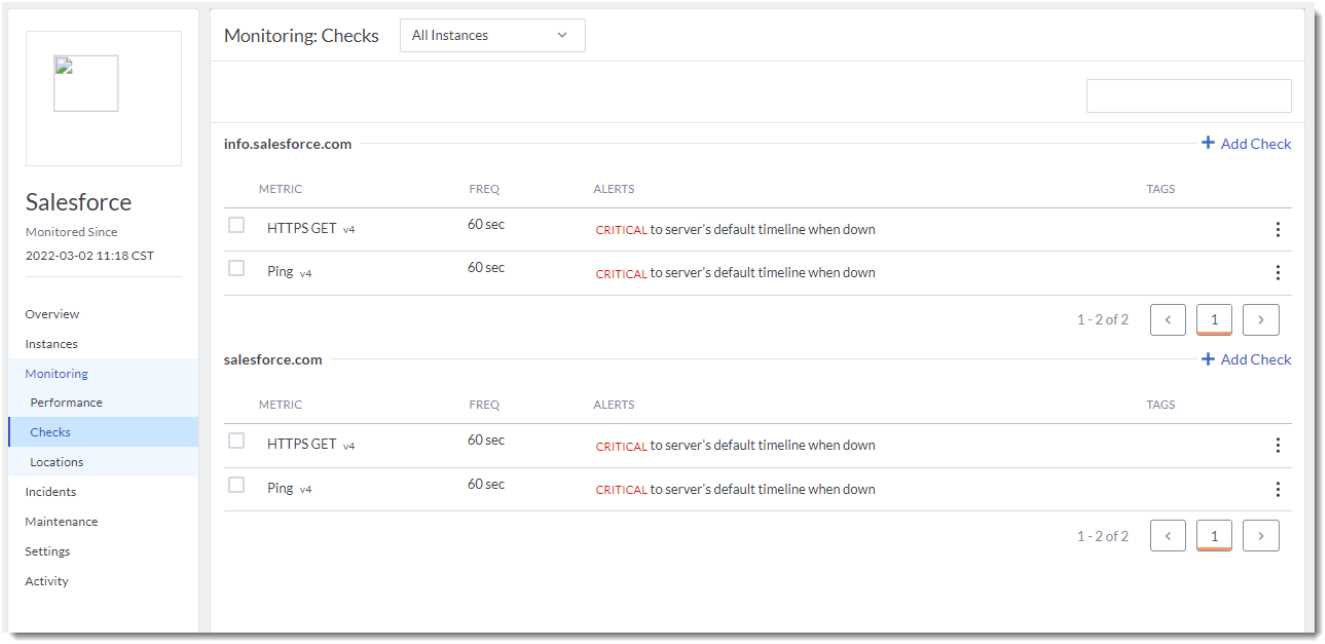
- Network quality checks
  - Jitter
  - Latency
  - MOS
  - Packet loss
- Synthetic checks
  - HTTP/HTTPS transactions (including SSL/TLS)
  - Traceroutes
- Download speed
- WiFi Rx (Reception) Rate
- WiFi Tx (Transmission) Rate
- WiFi Signal Strength
- Windows Endpoint Agent health metrics



The **Performance** section provides a visual representation of each metric, represented by a line graph that can be filtered by location, metric, and a specific time range.

Checks

The **Checks** tab displays all of the metrics configured for this application. You can add checks and configure metrics from this tab. Checks will be run from all applicable locations that have been added and selected.

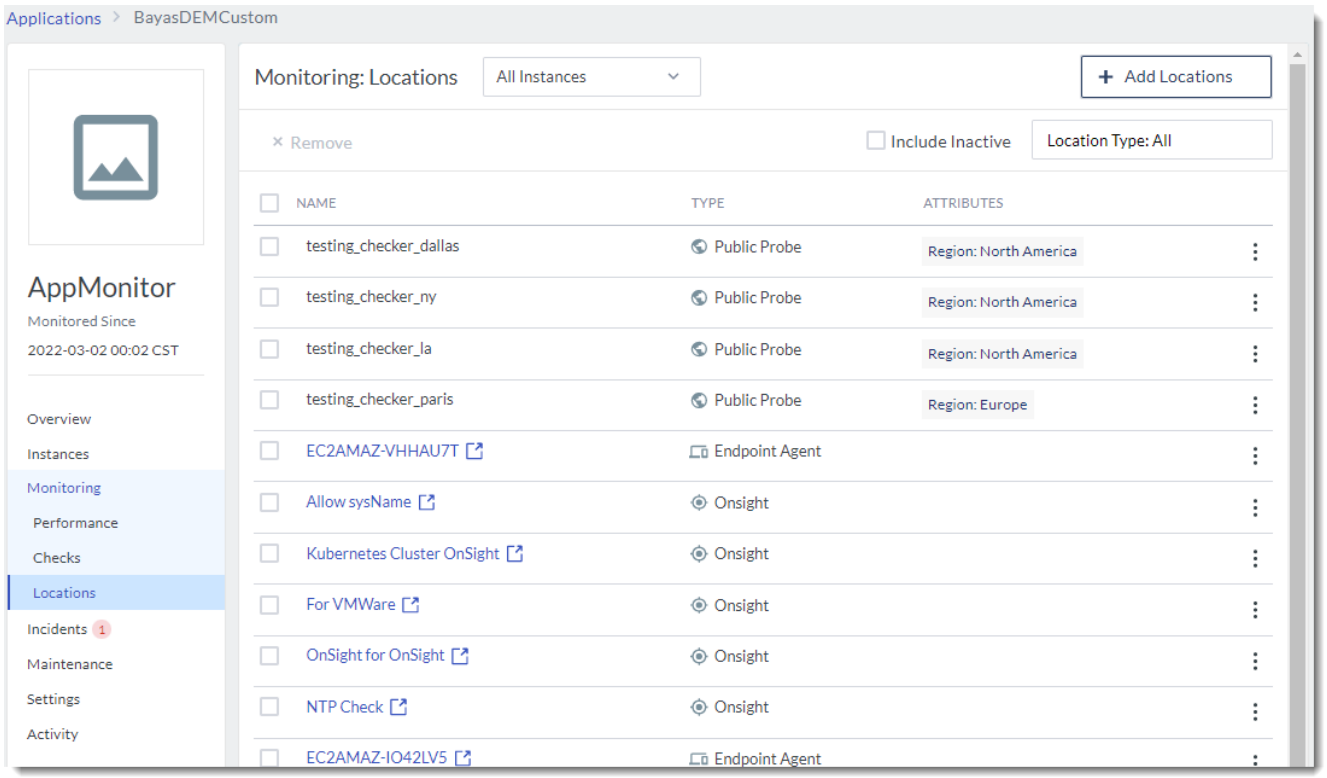


To add a check, see Create checks.

Locations

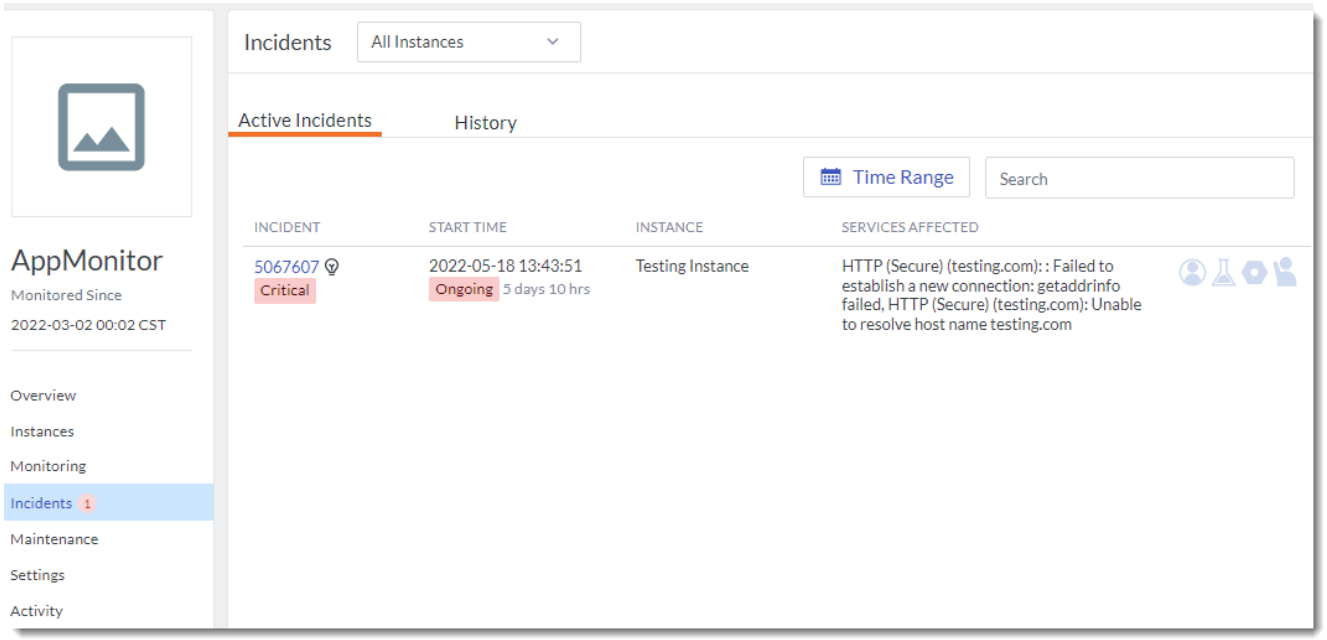
The **Locations** tab shows a list of all the **Locations** associated with the application. The list can be filtered by **Location Type** and include or exclude inactive locations.

To add a location click **Add Locations**. See Add monitoring locations for more details.

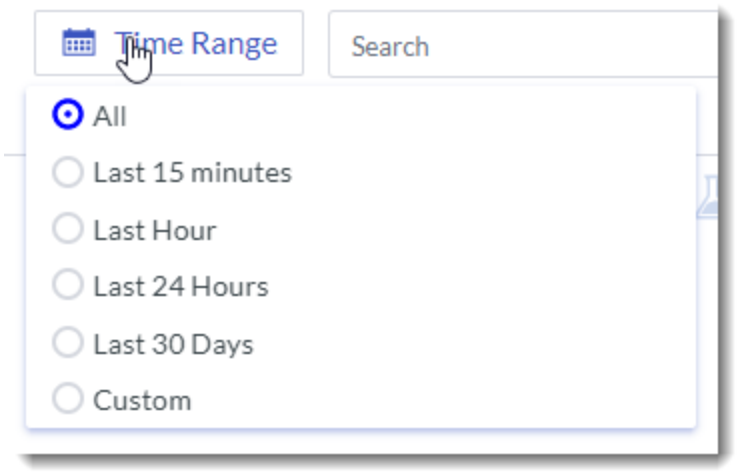


Incidents

The **Incidents** tab displays a list of active list of incidents affecting application instances.

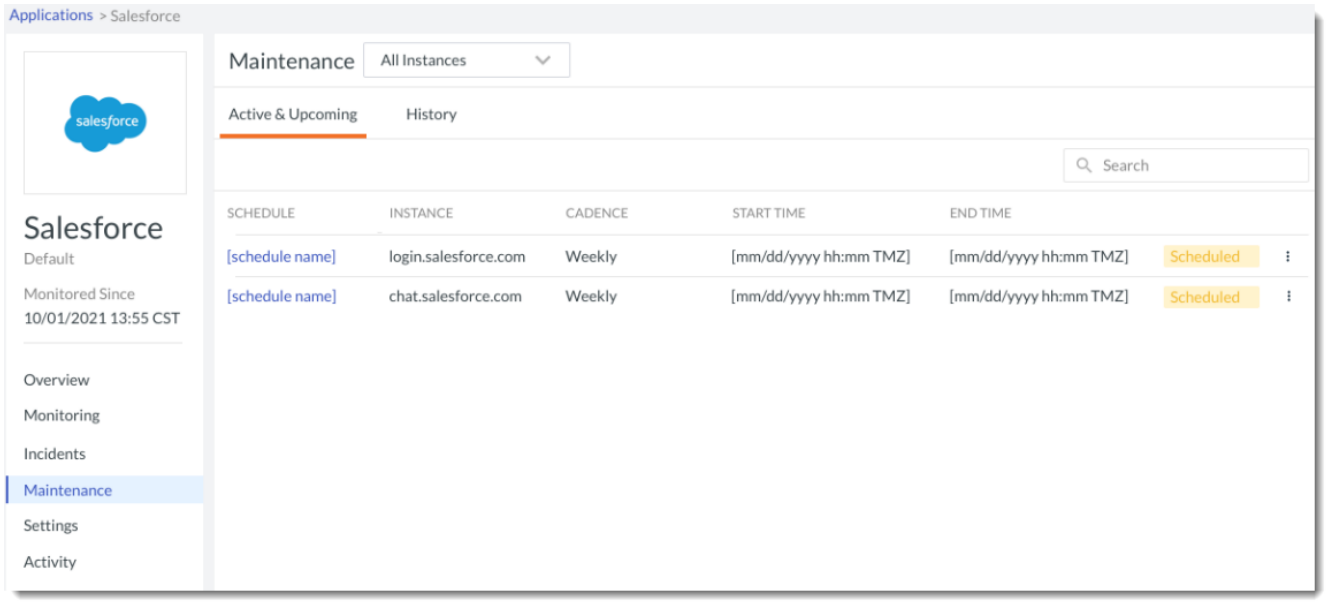


You can also specify a time range.



Maintenance

The **Maintenance** tab displays a list of active and upcoming maintenance affecting the application's instances.



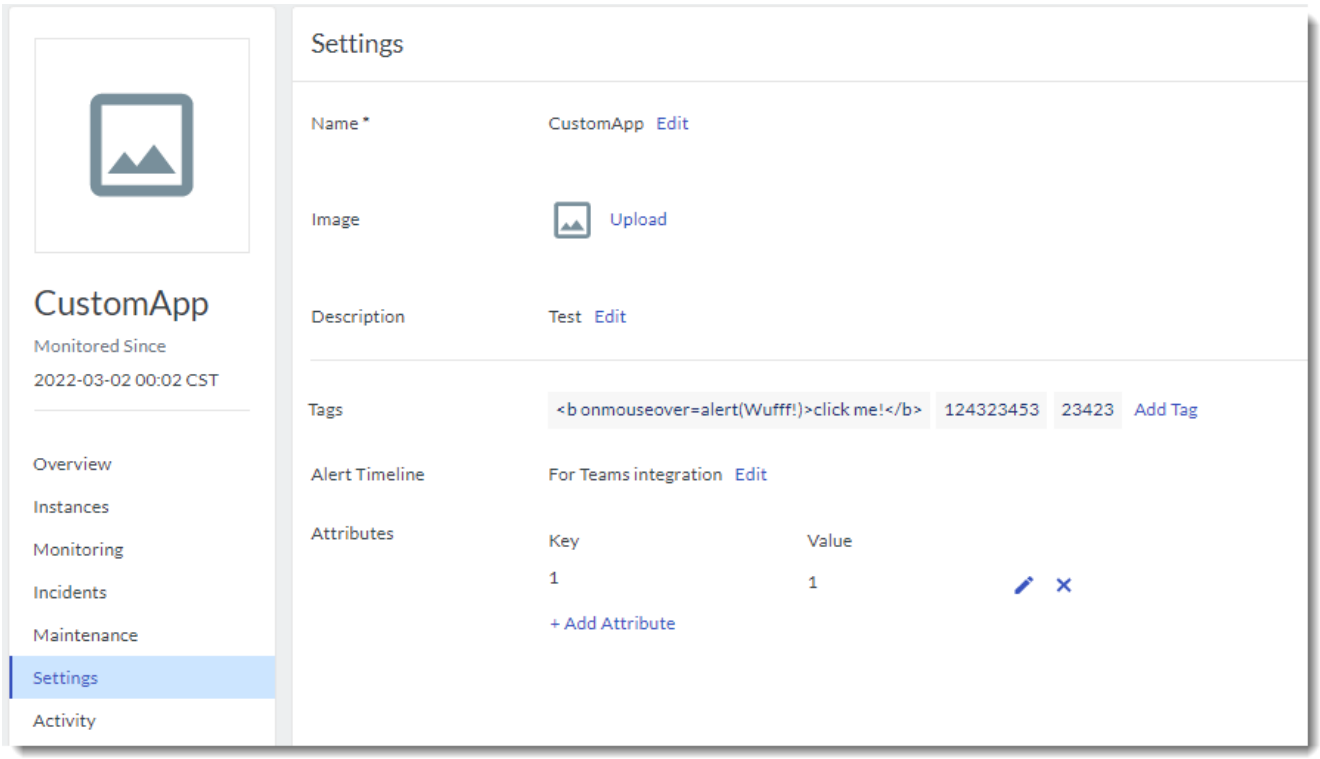
Settings

The **Settings** tab allows you to edit the following fields:

- Name
- Image
- Description

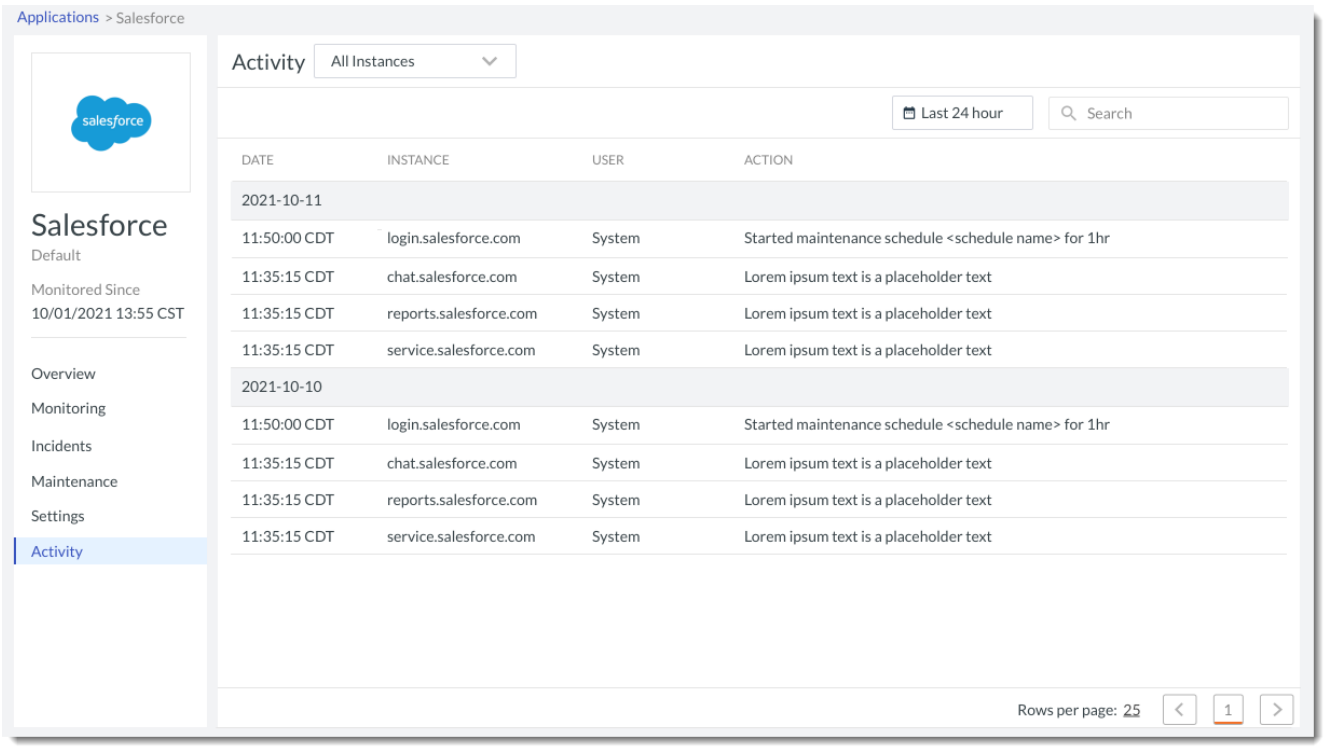
You can also:

- Apply tags to the application
- Assign alert timelines to the application
- Add attributes to the application



Activity

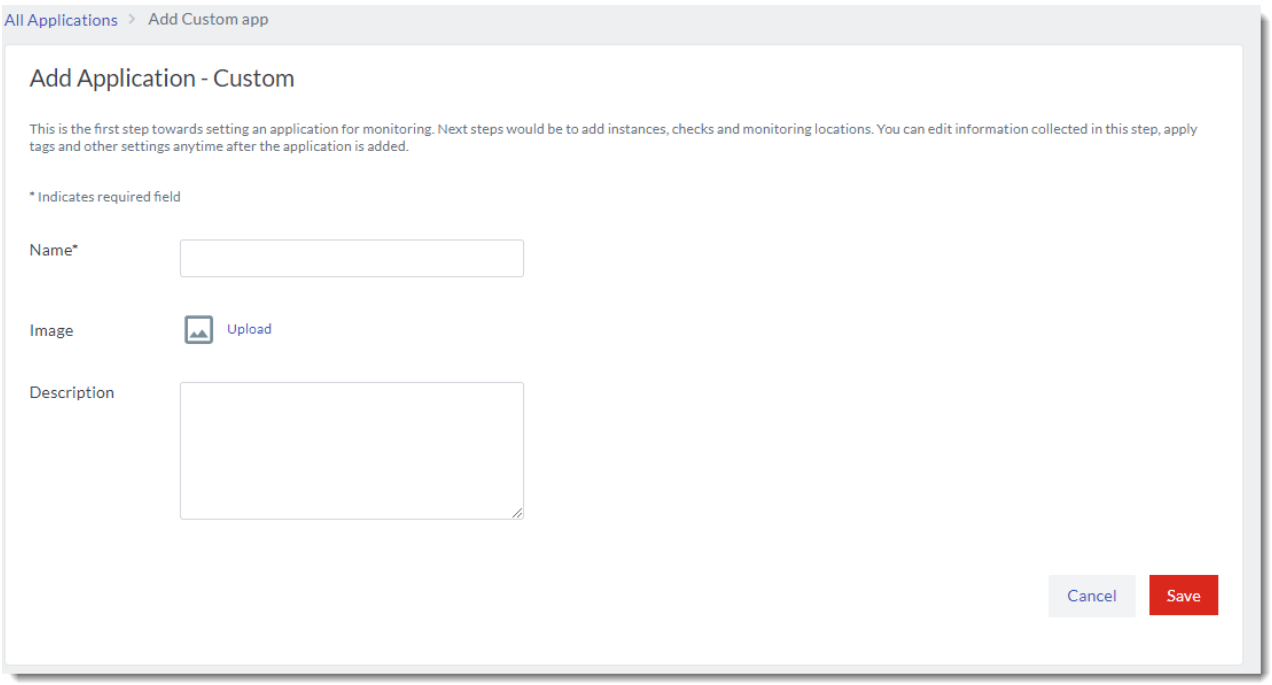
The **Activity** tab provides a log of all actions performed on the application. The initiator of the action, the date, the action performed, and target instance are detailed in the **Activity Log** table.



Add an Application

To add an Application, perform the following steps:

1. From the navigation menu, click **Monitoring** > **Applications**.
2. Select any of the **Default Applications** or **Add custom Application**. If you select any of the Default Applications, you may directly go to Monitoring Locations.
3. Enter a name and description for the application. You can also upload an image to be used as a logo for the application.



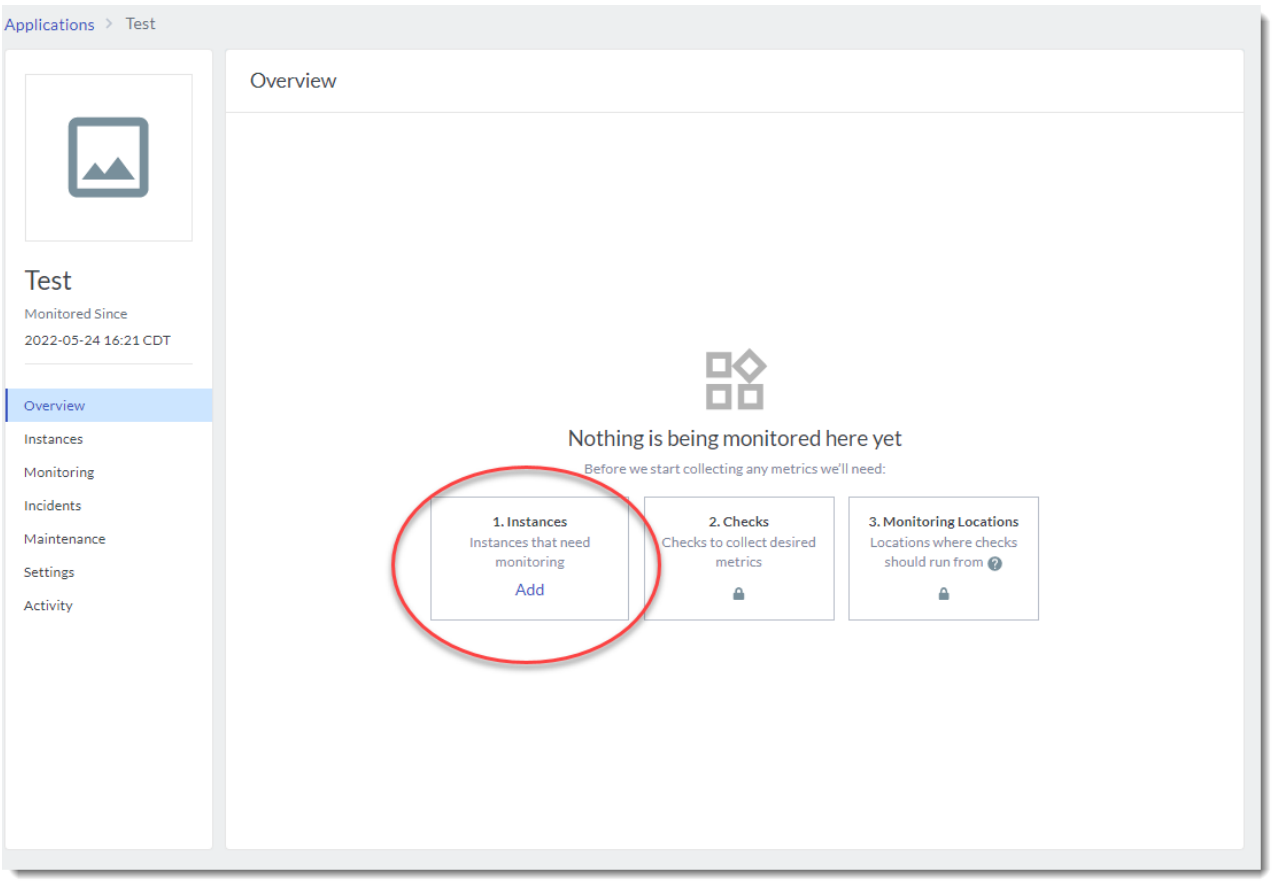
4. Click **Save**. The application is added to the **Application Monitoring** page.

You can now add instances to the application.

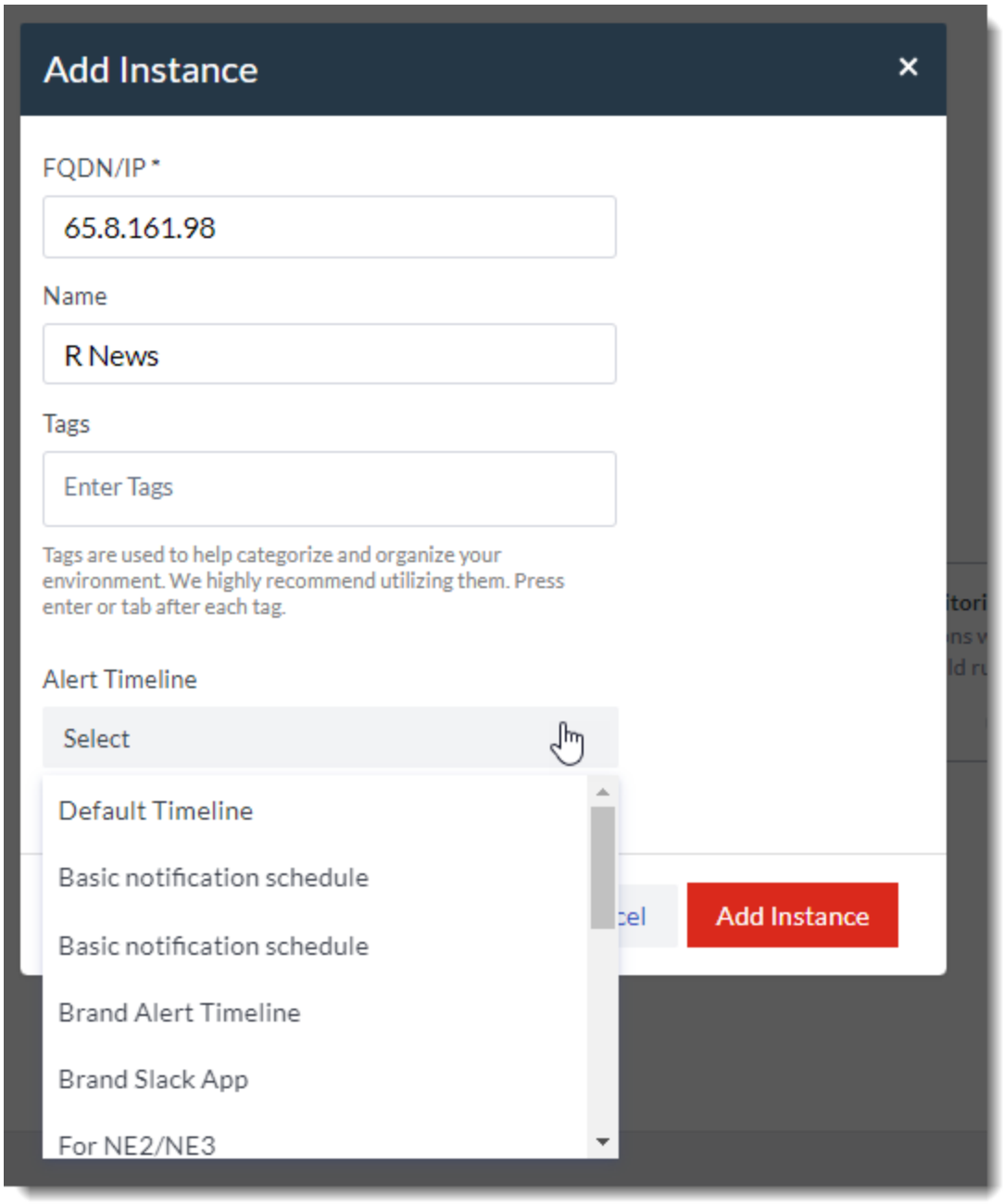
Add instances

To add an instances, perform the following steps:

1. From the **Overview** tab of the application during initial set up, click **Add** under **Instances**. Alternatively, you can also go to the **Instances** tab and click **Add Instance**.



The **Add Instance** modal is displayed.



2. Enter the required information. The fields are defined in the following table.

Field	Description
FQDN/IP	**Required** Enter the FQDN or IP address of the instance.
Name	**Optional** Enter a name for the instance.
Tags	**Optional**

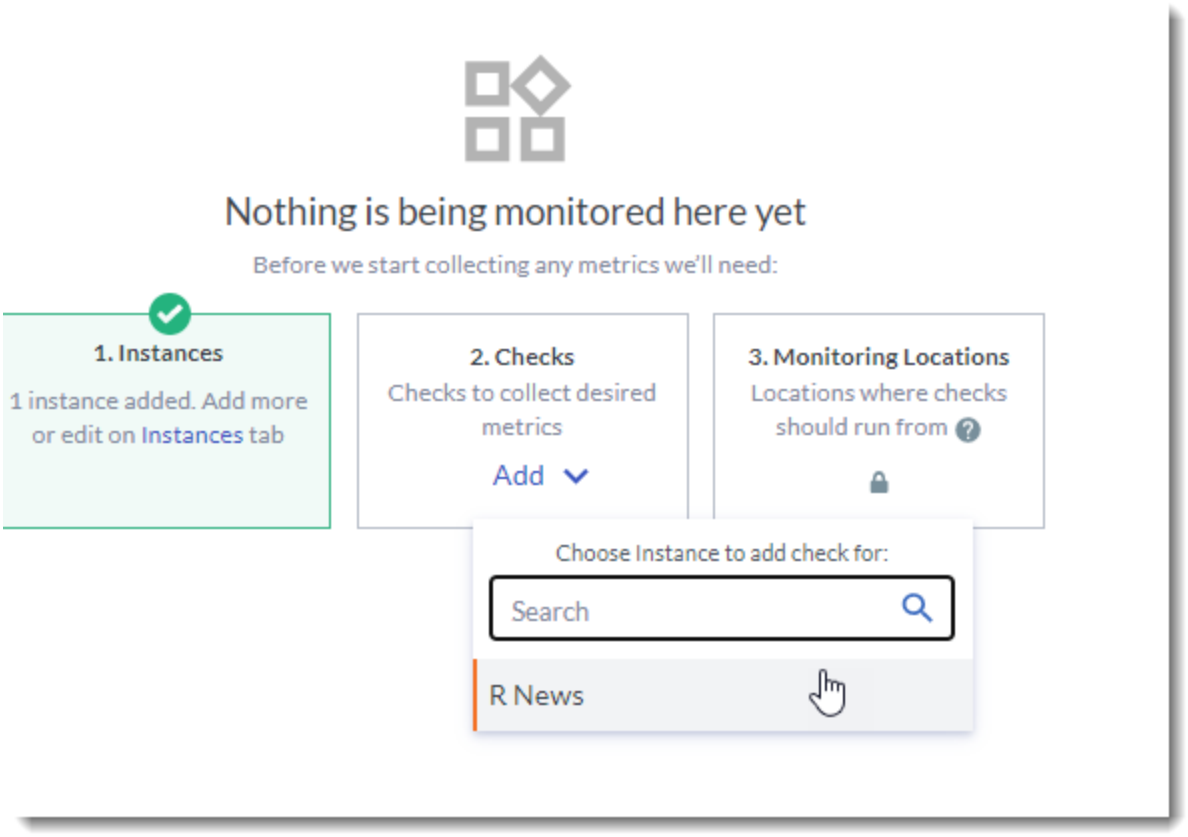
Field	Description
	Apply Tags to the instance. To add more than one tag, press the Enter or Tab key after each tag.
Alert Timeline	**Optional** The Alert Timeline to use for the instance.

3. Click **Add Instance** to create the instance. To add more instances, go to the **Instances** tab and click **Add Instance**.

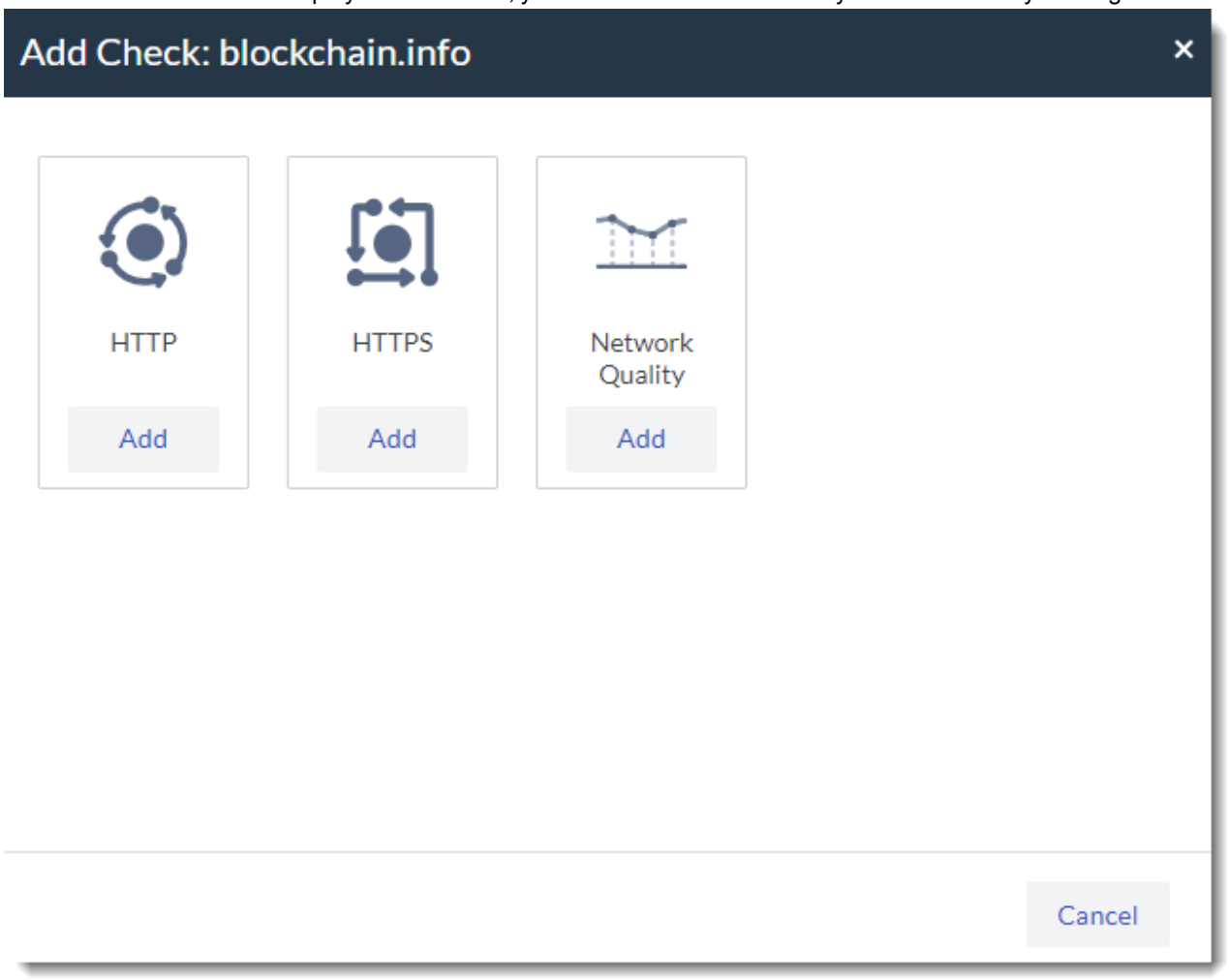
Create checks

To create a check, perform the following steps:

1. From the **Overview** tab of the application during initial set up, click **Add** under **Checks**. Alternatively, you can also go to the **Checks** tab and click **Add Check**.



2. The **Add Check** modal is displayed. From here, you can select the check that you want to add by clicking **Add**.



3. Configure the metric. For specific instruction on how to add and configure each metric, see Application plugins.

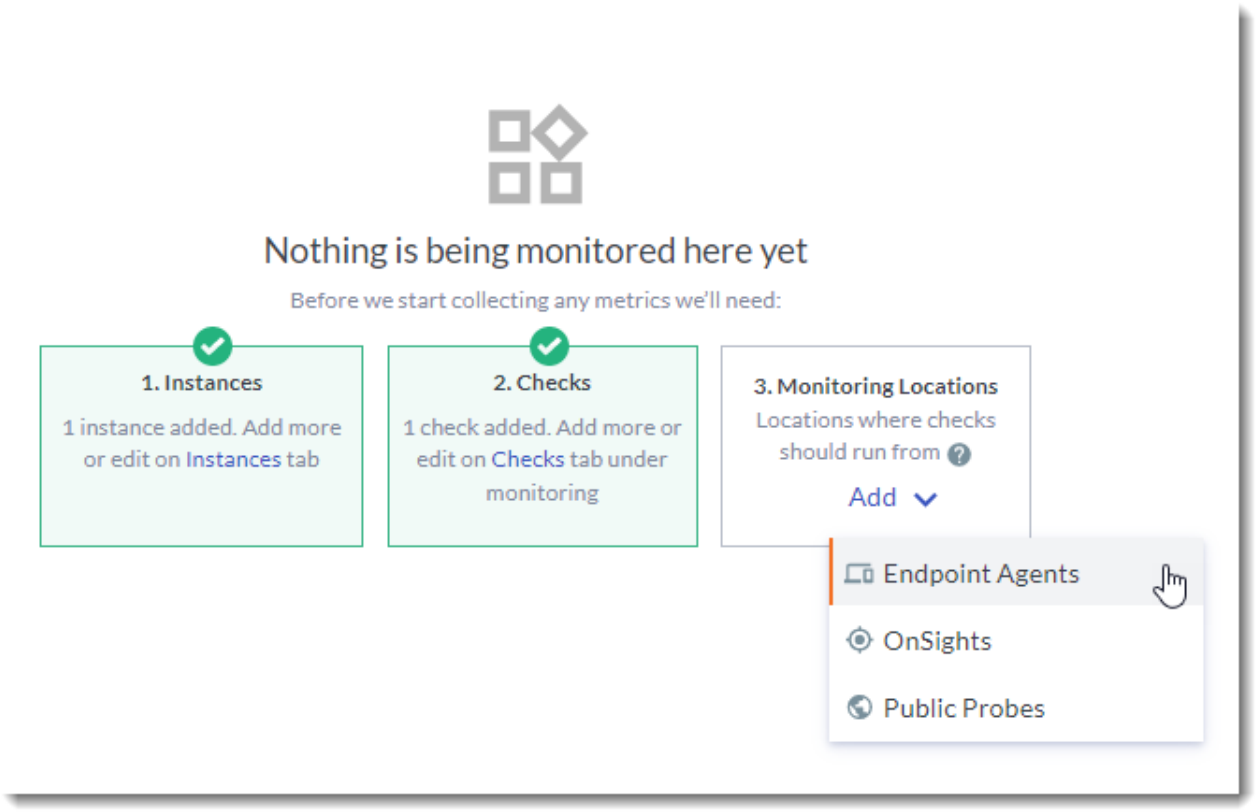
Add monitoring locations

Application Monitoring supports three location types:

- Public probe - Global network of monitoring nodes designed to track the availability of your servers 24 hours a day from outside your network.
- OnSight - The OnSight vCollector sits behind your firewall and functions similarly to our public monitoring nodes, which runs uptime checks and synthetic monitoring against your publicly available servers and web endpoints. The OnSight can be used to verify performance from core office or data center environments.
- Endpoint Agents (DEM Agent) - Collects key OS health and application performance metrics from remote locations like employee computers, retail locations, or public probes for external verification.

To add a monitoring location, perform the following steps:

1. From the **Overview** tab of the application during initial set up, click **Add** under **Monitoring Location**. Alternatively, you can also go to the **Locations** tab and click **Add Location**.



2. Select a location type from the following options:
   a. Endpoint Agent
   b. OnSights
   c. Public probes
3. Choosing any of the above location types launches a modal where you can search for and add an existing location to the application.
4. 4. Click **Add** to assign the location to the application. To add more locations, go to the **Locations** tab and click **Add Location**.

Path monitoring

The path monitoring feature allows you to look into the reachability of target instances from one or more sources. The source can be OnSights, DEM Agents (MacOS and Windows), or public checkers. Path monitoring also features a timescale selector that allows you to see the results of a path recorded at any point in time.

Prerequisites

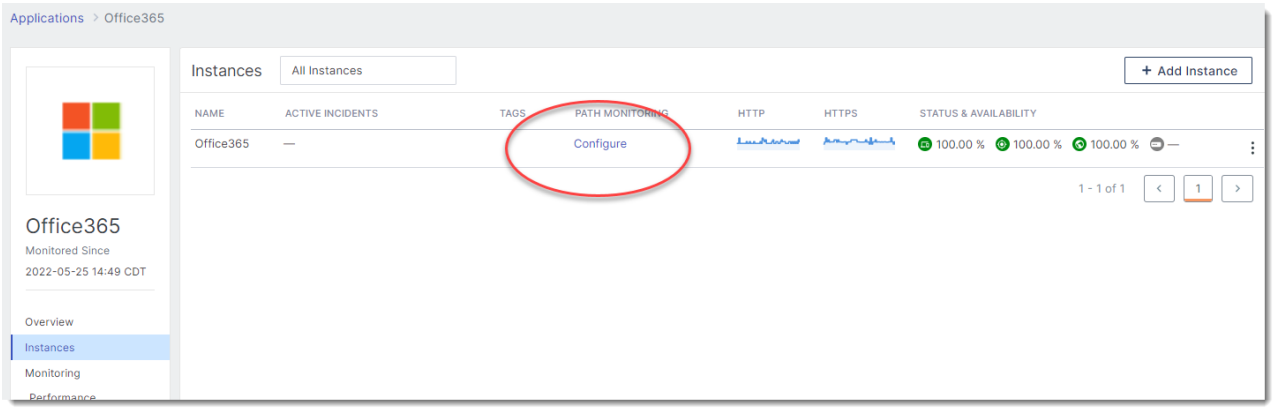
- FortiMonitor Windows Agent 23.3.7
- FortiMonitor MacOS Agent 2023.03.11247
- OnSight vCollector 23.3.7

Note: Path monitoring is currently only available for DEM applications.

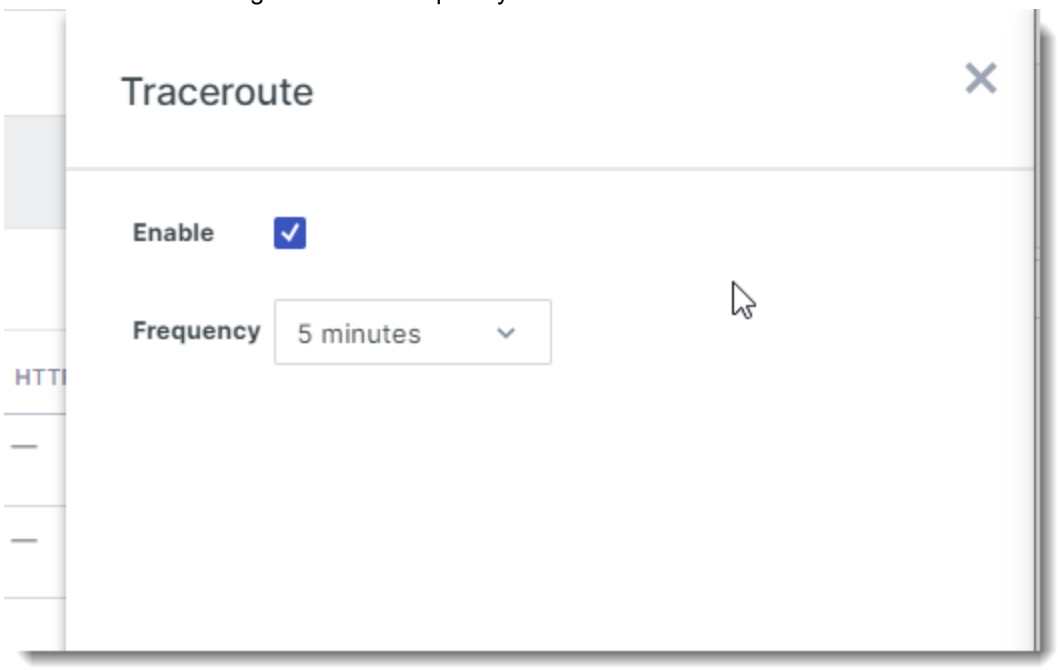
Configure Path Monitoring

To configure path monitoring, perform the following steps:

1. From the navigation menu Select **Monitoring** > **Applications**.
2. Open an **Application**.
3. Click **Instances**.
4. Choose the instance that you want to configure then click **Configure**. The configuration drawer will slide out.



5. Enable Path Monitoring and set the frequency.

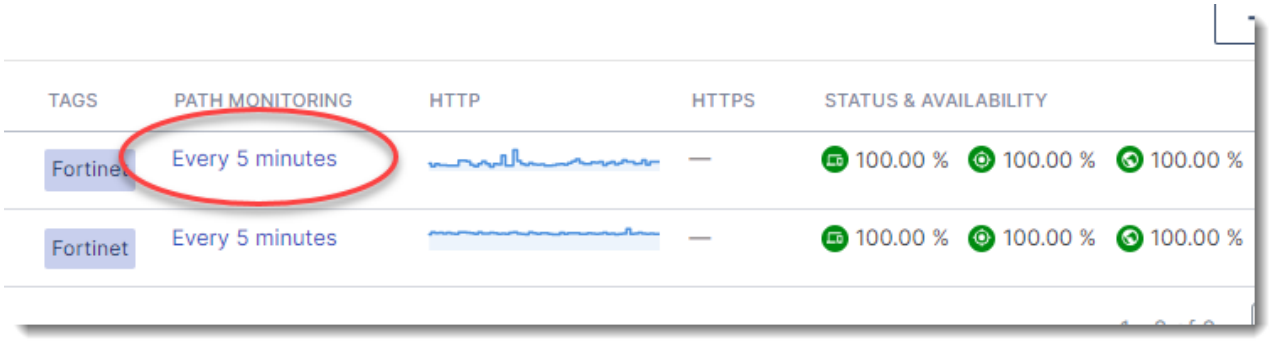


6. Click **Save**.

Change an existing configuration

To change an existing configuration, perform the following steps:

1. From the navigation menu Select **Monitoring** > **Applications**.
2. Open an **Application**.
3. Click **Instances**.
4. Choose the instance that you want to update then click the Path Monitoring link. **Configure**. The configuration drawer will slide out.

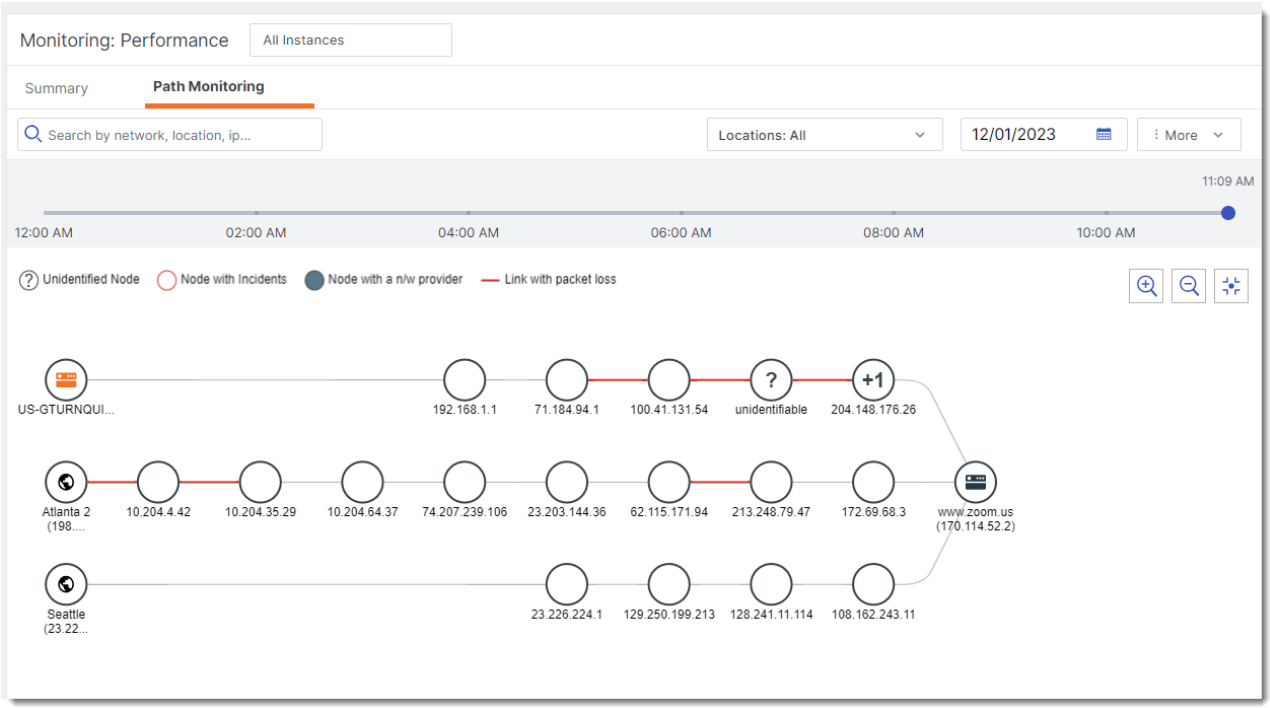


5. From here you can disable Path Monitoring or change the frequency.
6. When you are done, click **Save**.

View Path Monitoring

To see the configured path monitoring, perform the following steps:

1. 1. From the navigation menu Select **Monitoring** > **Applications**.
2. 2. Open an **Application**.
3. 3. Select **Performance**.
4. 4. Click the **Path Monitoring** tab.



Timescale selector, filters, and toggles

Timescale selector and date

The timescale selector and date selector allow you to see the results of a path from all of the sources to the target instance at any point of time. If there are multiple instances in an application, the path for each target is shown as well. Select a date or slide the selector to any point in the timescale to see the results.

Filters and toggles

The following filters and toggles are available:

- **Source** – Toggle to display source icons
- **Identified Node** – Toggle to display nodes that have been identified
- **Unidentified Node** – Toggle to display not that are not identified

- **Node with incidents** – Toggle to display nodes with active incidents
- **Link with packet loss** – Toggle to display links or paths with packet loss. These links are in red.
- **Target** – Toggle to display targets
- **Node Name/IP** – Hide/unhide the name or IP address of nodes
- **Legend** – Hide/unhide the toggles
- **Locations** – Select which locations are shown/hidden

You can also use the search box to search for and display specific networks, locations, or targets.

View node details

You can view the details of a source, destination, or hop node by moving the cursor to a particular node.

The following details are displayed:

- Type of node (source, destination, or hop)
- Provider
- Location
- For source nodes, the number of hops taken to reach the destination is also displayed.

View link details

You can view the details of link by moving the cursor to a particular link.

The following details are displayed:

- Packet loss
- Latency

Links with packet loss are highlighted in red.

SD-WAN application monitoring

SD-WAN overview

SD-WAN is a software-defined approach to managing Wide-Area Networks (WAN). It consolidates the physical transport connections, or underlays, and monitors and load-balances traffic across the links. VPN overlay networks can be built on top of the underlays to control traffic across different sites.

For more information, see this article.

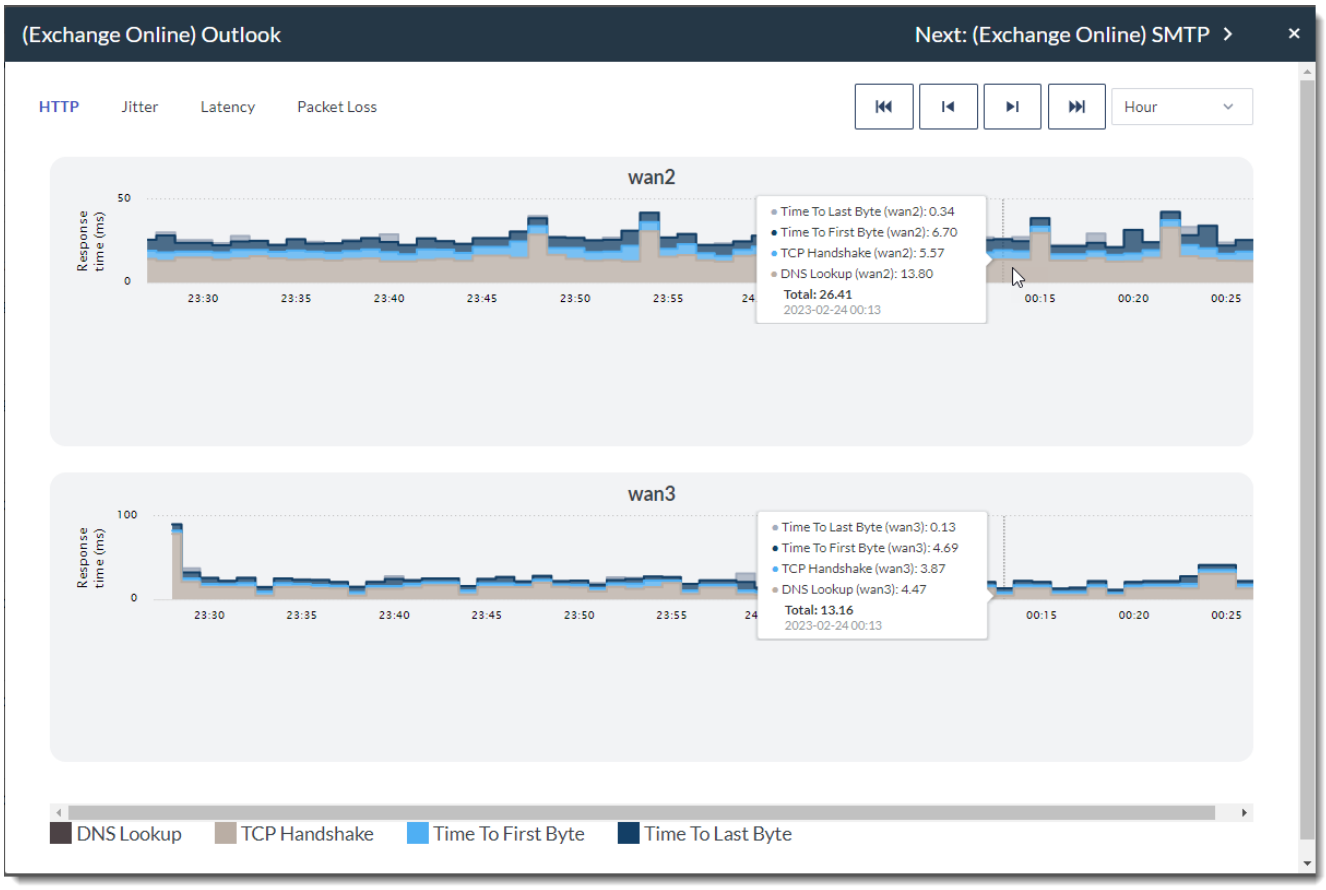
FortiMonitor monitoring approach

SD-WAN application monitoring works in tandem with DEM application monitoring to provide continuous end-to-end per-application performance monitoring from multiple vantage points, including one or more SD-WAN devices. SD-WAN synthetic checks simulate a connection to an application to provide real-time metric collection and monitoring for key network and application layer performance indicators. Synthetic monitoring is done per-application over SD-WAN underlays, thus augmenting Fortinet Secure SD-WAN monitoring.

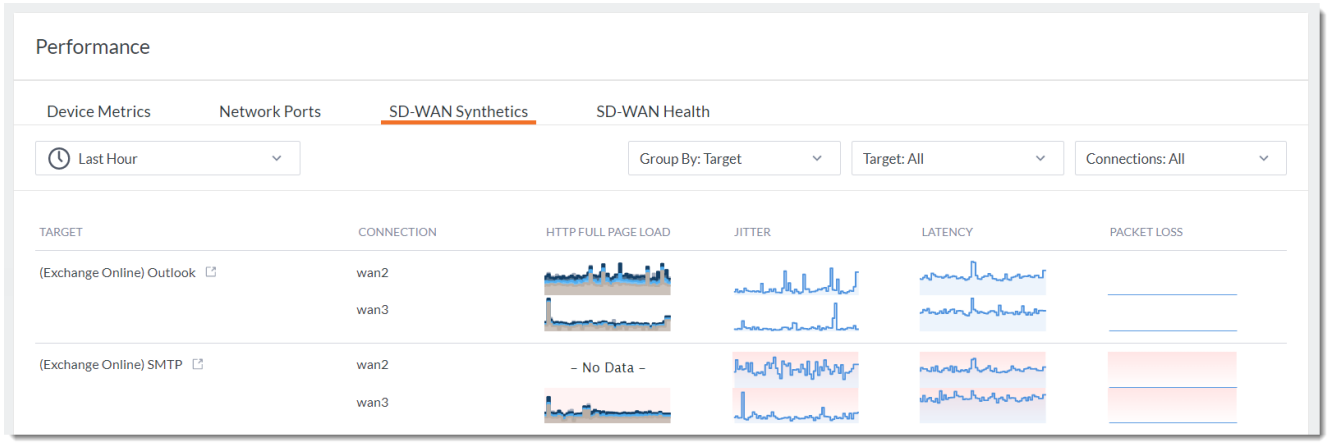
Synthetic checks and network visibility

FortiMonitor adds visibility to the following to distinguish server response issues:

- DNS Lookup
- TCP Handshake
- SSL Handshake
- Time to First Byte
- Time to Last Byte



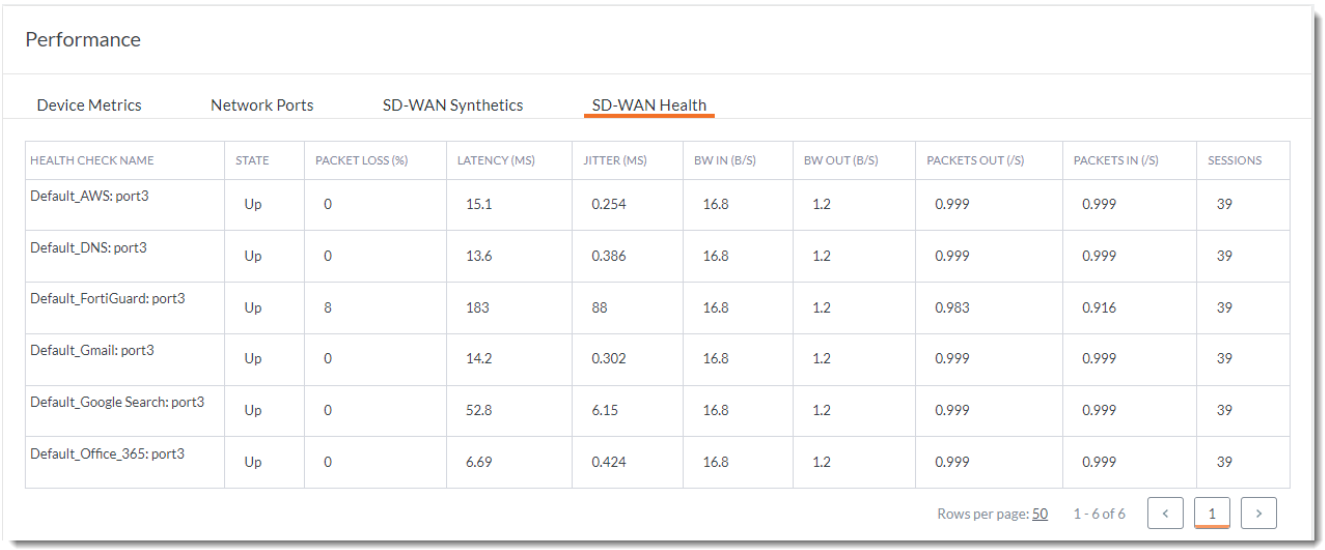
The following image shows a summary of an application's performance.



In addition to continuous synthetic monitoring, FortiMonitor can also monitor SD-WAN link-level metrics such as:

- Packet loss
- Latency
- Jitter
- Bandwidth in
- Bandwidth out

- Packets out
- Packets in
- Sessions



Get started with SD-WAN application monitoring

To get started, go through the following procedures in order:

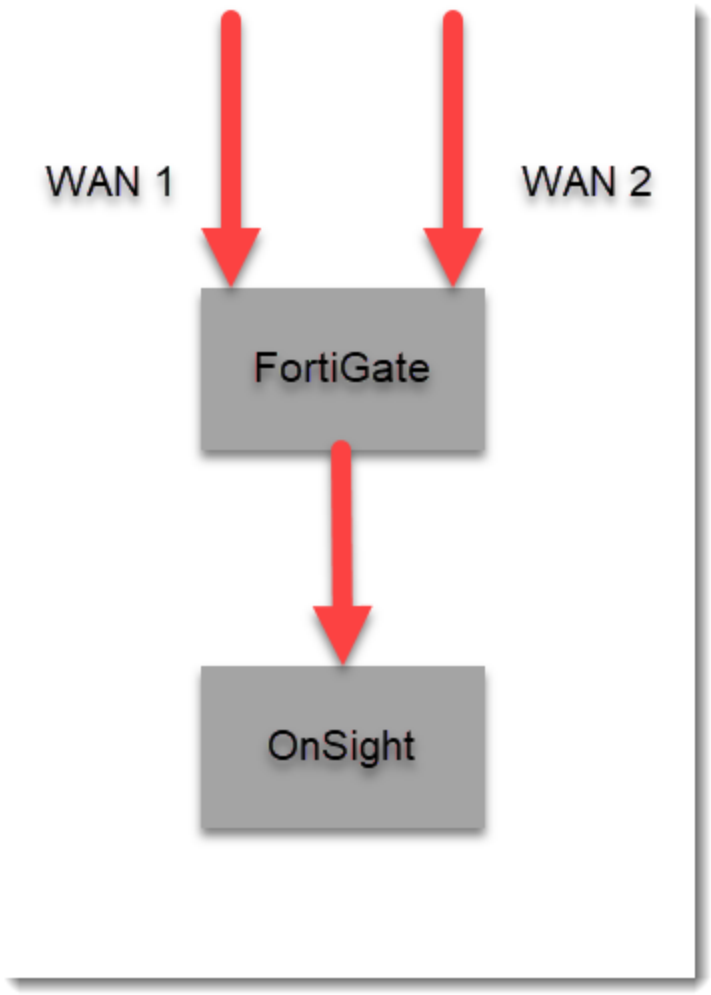
1. FortiGate and OnSight configuration for SD-WAN synthetic monitoring
2. Set up SD-WAN application monitoring

FortiGate and OnSight configuration for SD-WAN synthetic monitoring

This article provides the steps on how to set up an OnSight for SD-WAN synthetic monitoring.

Introduction

SD-WAN underlay monitoring is achieved by assigning additional IP addresses to the OnSight, one per underlay, which are then routed out through the FortiGate using a single underlay. Synthetic checks will then be automatically configured to originate from each of the source IPs in order to test each underlay.



To set up monitoring, you need to create a reserved IP on the FortiGate for each of the underlays. These need to exist within the same subnet as the primary OnSight IP (but are separate addresses from the primary OnSight IP) and not already in use by another device on your network.

After assigning the IPs on the FortiGate according to the instructions below, they will be automatically added and configured on the OnSight.

Note: SD-WAN synthetics are currently only available in the context of application monitoring, this feature will be added for individual instances in future releases.

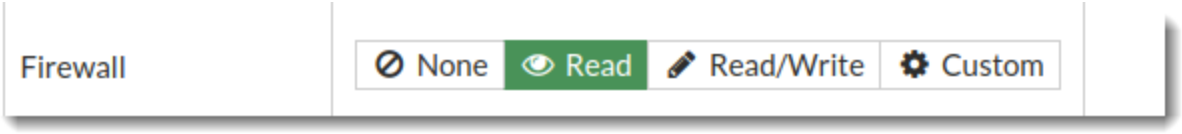
This guide will use the following objects and addresses. In the example below, we are using *10.10.10.11* and *10.10.10.12.* Substitute your chosen IPs in the details below.

SD-WAN Device	OnSight Agent IP	Firewall Address	SD-WAN Rule	SD-WAN Member ID	SD-WAN Zone
WAN1(port1)	*10.10.10.11*	FMR_IP1	Manual (#1)	1	virtual-wan-link
WAN2(port2)	*10.10.10.12*	FMR_IP2	Manual (#2)	2	virtual-wan-link

Note: While this example is presented using two underlays, it can be extended to use three or more.

Prerequisites

- An OnSight vCollector.
- Security Fabric must be configured in FortiMonitor.
- SD-WAN must be configured on the FortiGate device.
- The SD-WAN FortiGate must be monitored by FortiMonitor.
- The Admin profile used in the Security Fabric setup must have the read permission on firewall addresses.



Initial steps

1. Install an OnSight vCollector.
2. Onboard the FortiGate and configure Security Fabric. Note that SD-WAN must be configured on the FortiGate.

FortiGate configuration

This section details the steps on how to configure FortiGate.

Configure firewall objects

Create a firewall address object for each OnSight agent IP.

1. Log in to FortiGate.
2. Launch the CLI console. If you need more information on how to use the CLI console, see Using the CLI.
3. Configure firewall objects for each WAN connection and the OnSight's default IP.
   In this example, The IP addresses *10.10.10.11*, and *10.10.10.12* are used. Replace them with the correct OnSight IPs. Similarly, you can also replace *FMR_IP1* and *FMR_IP2* with your own values.
4. From the CLI, enter the following command:

```
config firewall address
    edit "FMR_IP1"
        set subnet 10.10.10.11 255.255.255.255
    next
    edit "FMR_IP2"
        set subnet 10.10.10.12 255.255.255.255
    next
end
```

Create SD-WAN rules

SD-WAN rules must first be set up to use SD-WAN synthetic monitoring. A specific IP address is needed for each WAN connection.

In this step we will create a manual SD-WAN rule to steer each OnSight IPs monitoring traffic through its respective WAN underlay.

- OnSight monitoring traffic will be marked using the **agent-exclusive** command. This command marks traffic as FortiMonitor health checks.
- Each OnSight firewall object will be manually steered to its respective WAN underlay.

1. Launch the CLI console.
2. Enter the following command to create the SD-WAN rules:

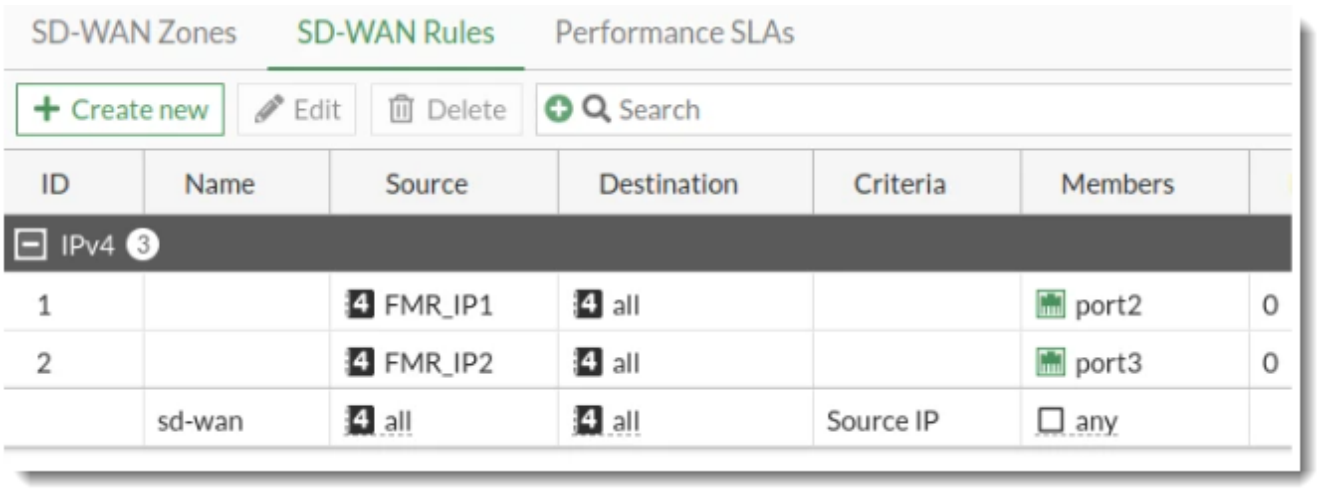
```
config system sdwan
    config service
        edit 1
            set dst "all"
            set src "FMR_IP1"
            set priority-members 1
            set agent-exclusive enable
        next
        edit 2
            set dst "all"
            set src "FMR_IP2"
            set priority-members 2
            set agent-exclusive enable
        next
    end
end
```

Note: The command `set agent-exclusive enable` is a FortiOS dependency of 7.2.4 or later.

In this example, *rules 1* and *2* are used but you may configure any available rule ID.

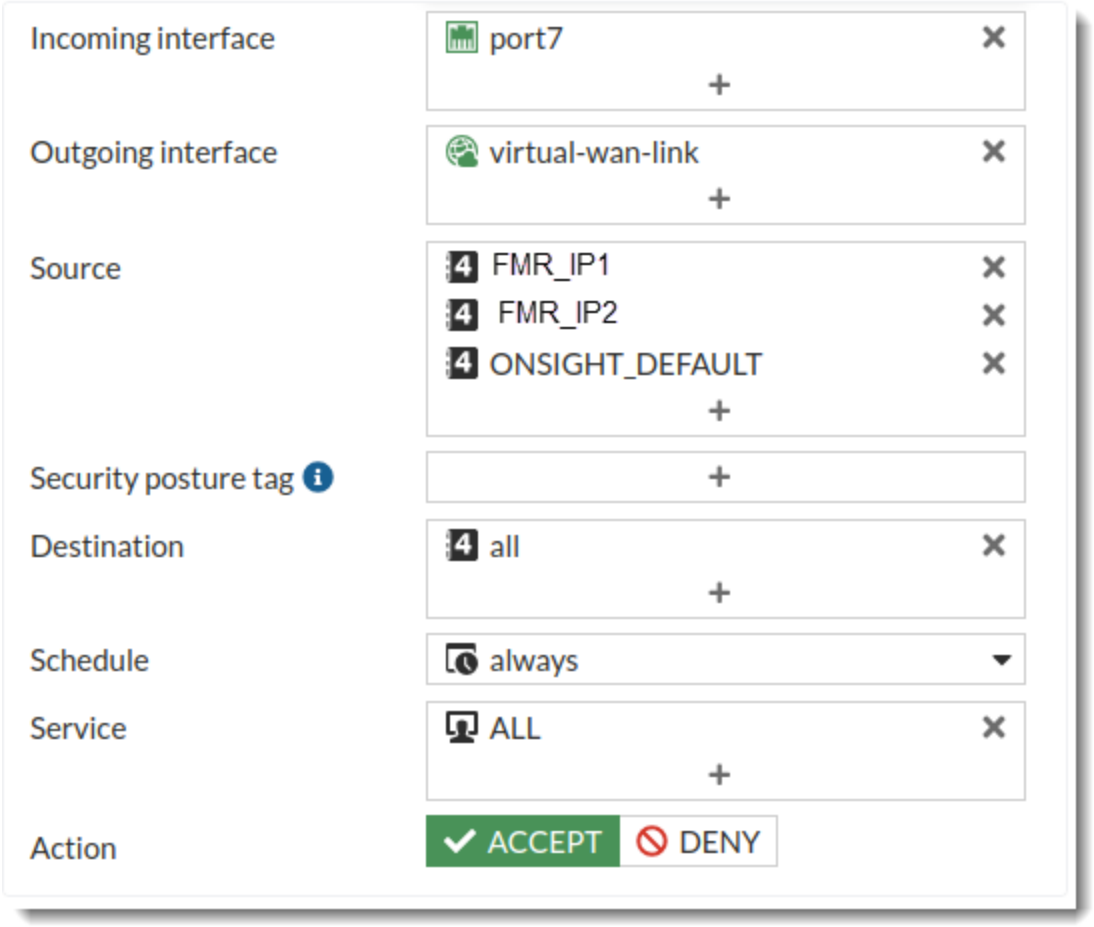
The Source (src) and Destination (dst) parameters are used to identify traffic. If the traffic is from a specific source address ,its associated SD-WAN rule will be used. See SD-WAN rules overview for more information.

Note that the SD-WAN rules created for SD-WAN synthetic monitoring must be set to the highest priority. You can do this by dragging the rules to the top of the list.



Add the firewall addresses to the firewall policy

Add *FMR_IP1* and *FMR_IP2* to the Firewall policy from the OnSight to virtual-wan-link, so that the traffic from OnSight can travel to the internet.



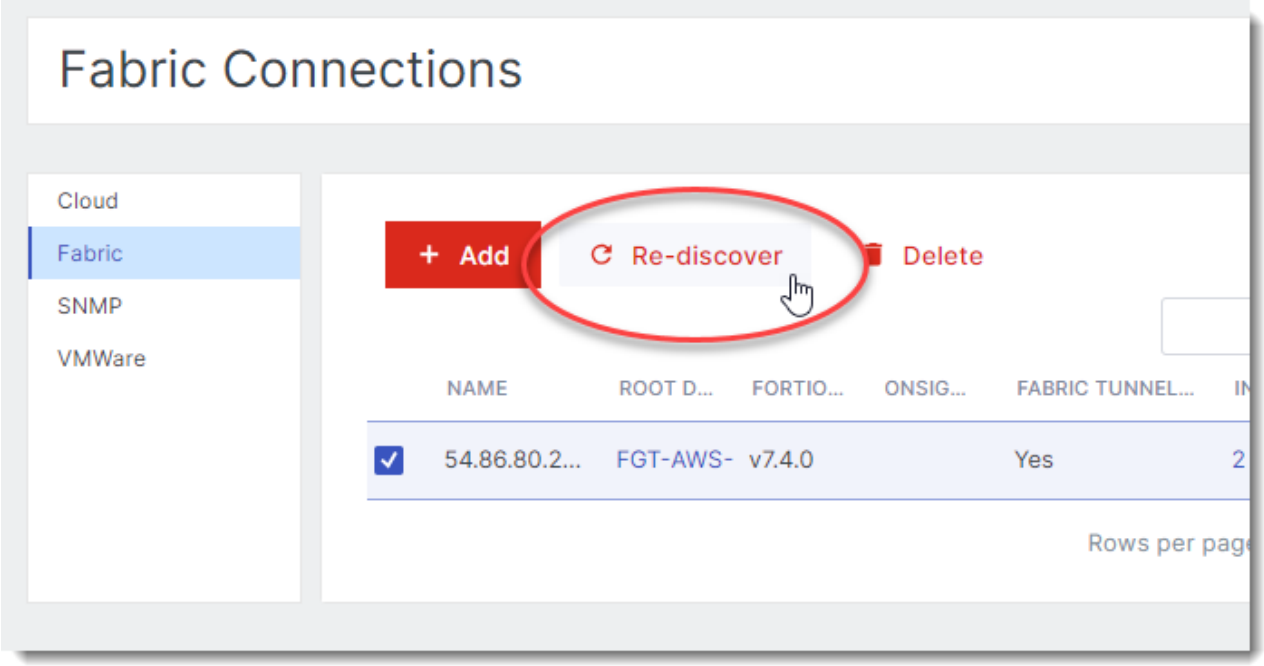
- Incoming interface - The interface connected to the OnSight.
- Source - Add *FMR_IP1* and *FMR_IP2*. If the Source is **All**, there is no need to add the addresses.

FortiMonitor configuration

Enable SD-WAN synthetic monitoring

Once the configuration on the FortiGate has been completed:

Go to **Monitoring** > **Infra setting** > **Fabric** then run a rediscovery on the fabric integration.

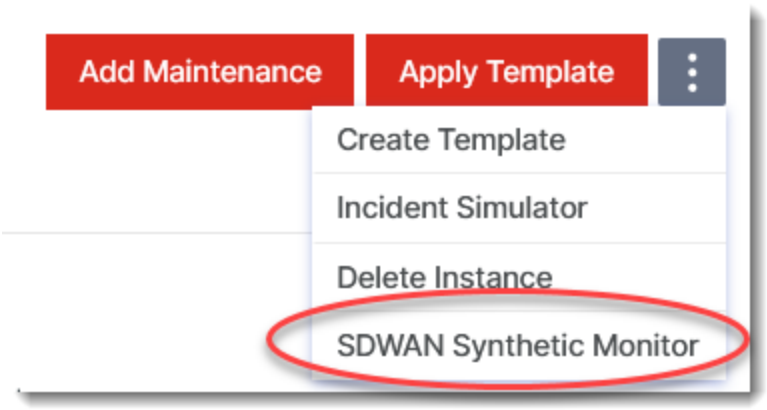


Rediscovery module performs the following:

- Retrieves the firewall address information including the FortiMonitor IP, OnSight agent IPs, and SD-WAN rules.
- Validates the values.
- If there are no issues, the SD-WAN synthetic monitoring menu option will be enabled.
- After enabling SD-WAN synthetic monitoring. the OnSight will then assign the additional IP addresses.

1.
2. Open the FortiGate details page in the Control Panel.

3. Click **SD-WAN Synthetic Monitor**.

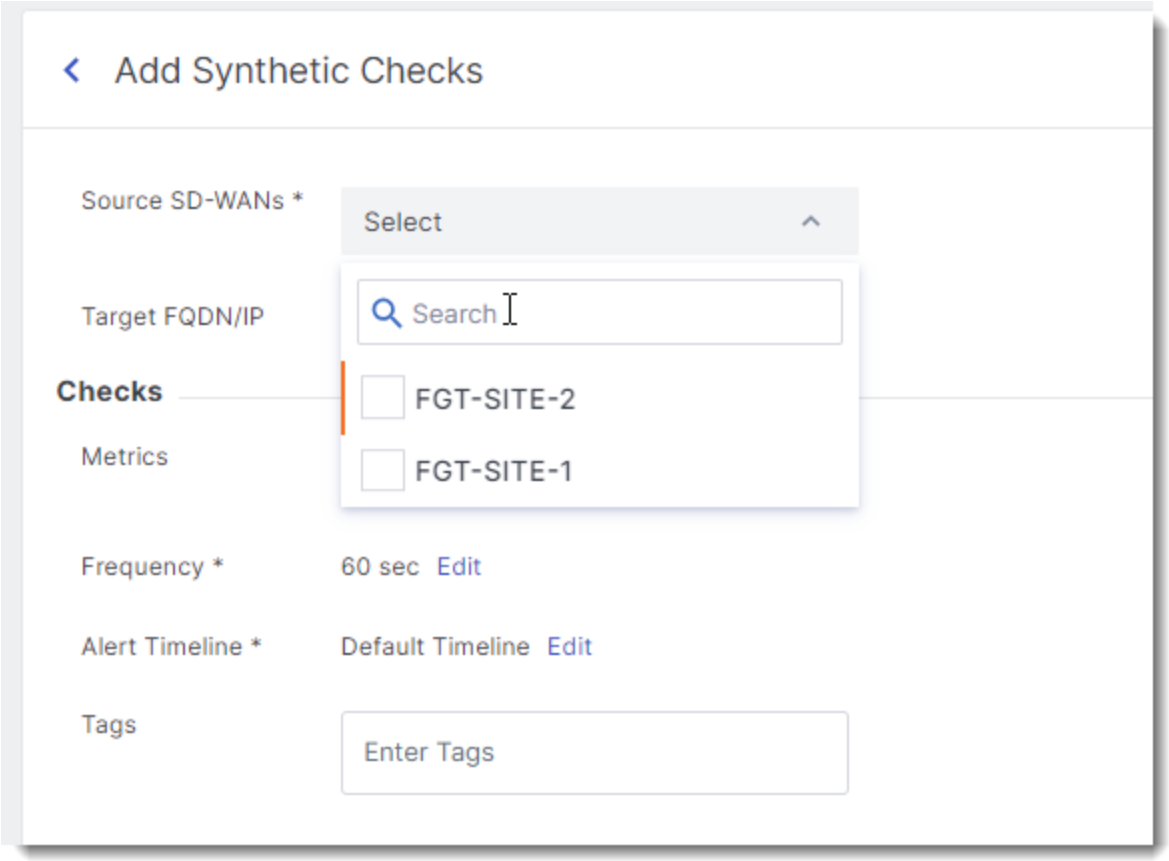


4. On the modal that appears, click **Enable SDWAN Synthetic Monitor**.

Add SD-WAN monitoring to an application

1. Go to **Monitoring** > **Application**.
2. Select the application where you want to add SD-WAN monitoring.
3. Select **Locations**.
4. Select **Add Locations** > **SD-WAN Devices**.

5. In the **Source SD-WANs** drop-down field, select the FortiGate you configured.

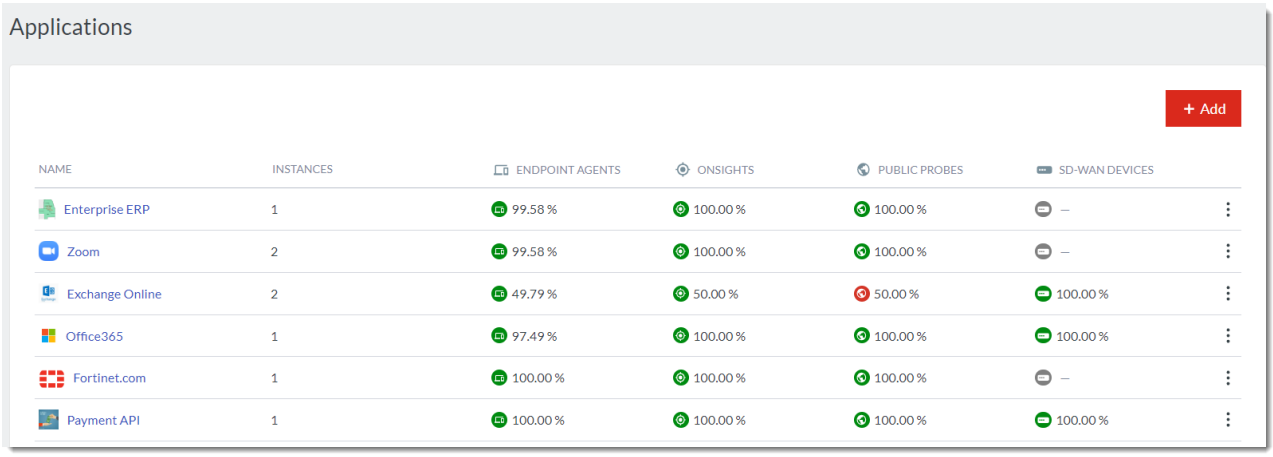


6. Click **Save**.
   Data should begin to flow in about 5-10 minutes.

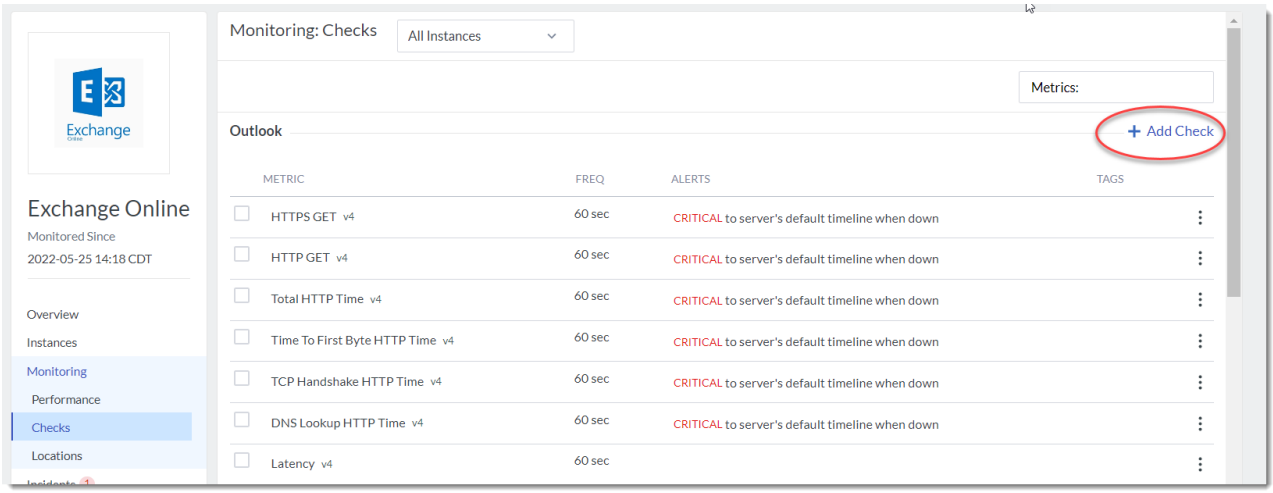
Set up SD-WAN application monitoring

Perform the following steps to set up application synthetic monitoring over SD-WAN underlays:

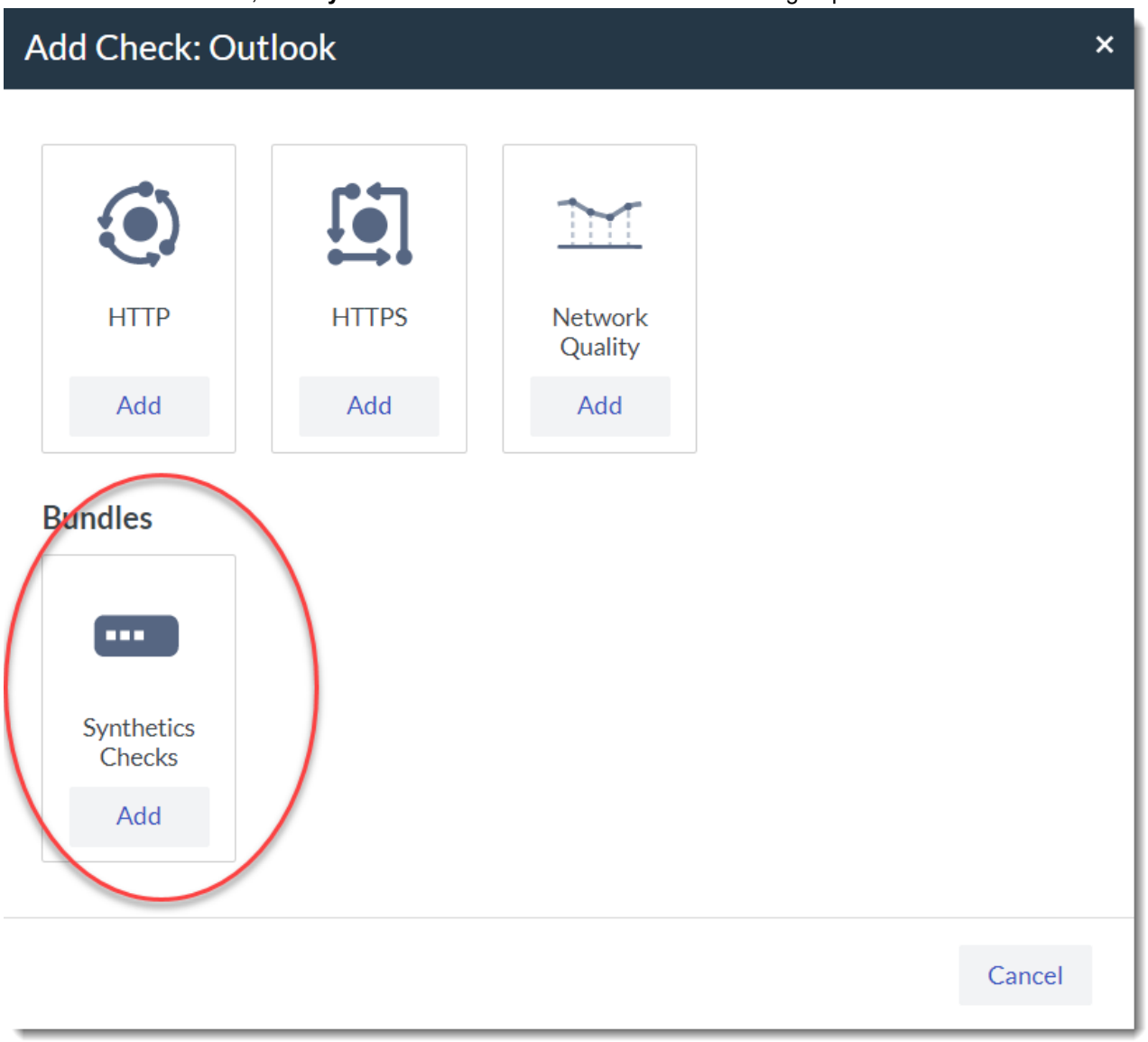
1. From the navigation menu, click **Monitoring** > **Applications**.



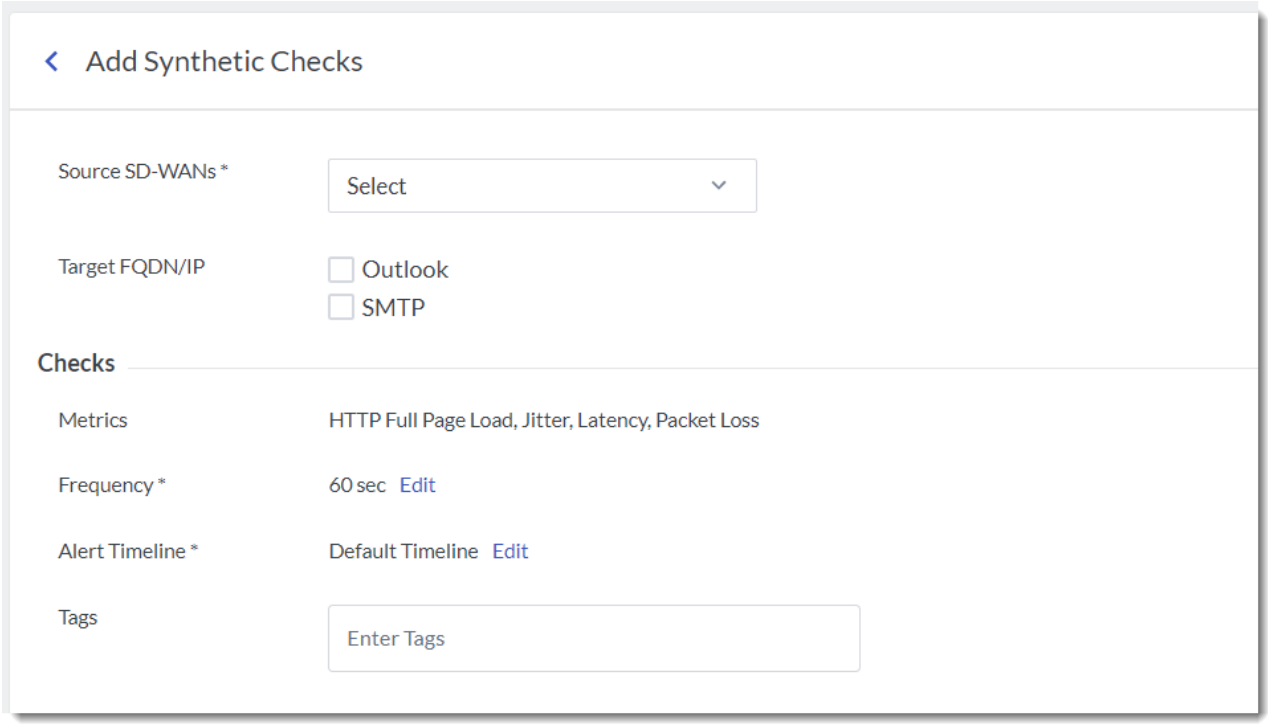
2. Select an application or create a new one.
3. Click the application to open its page.
4. Go to **Monitoring** > **Checks**.
5. Click **Add Check**.



6. In the **Add Check** modal, click **Synthetic Checks** located under the **SD-WAN** group.



7. Select one or more **Source SD-WANs** and **Target FQDNs/IPs**.

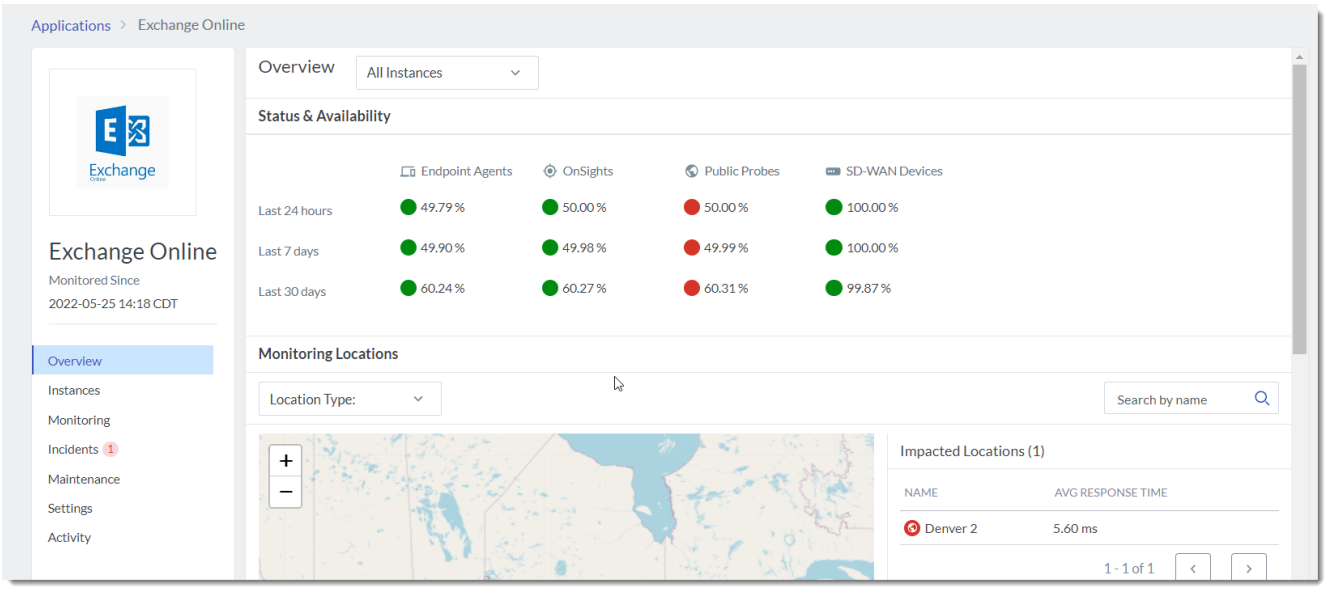


You also have the option to edit the check frequency and the Alert Timeline.

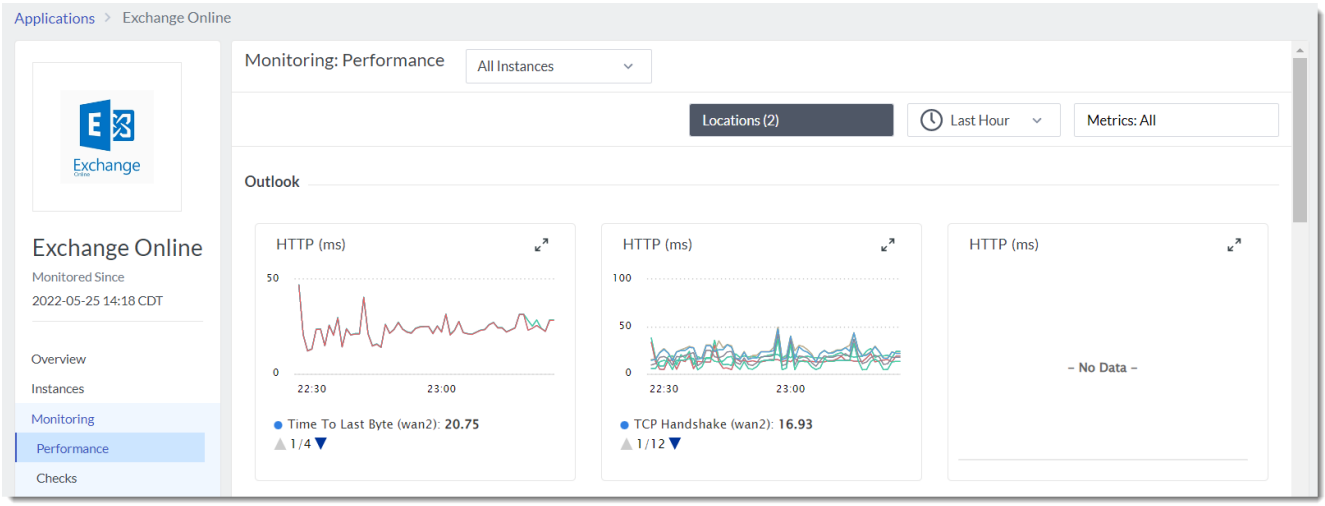
8. Click **Save**.

Verify the configuration

To verify that you have successfully configured SD-WAN application monitoring, go back to the application's page. Under the **Status & Availability** section, you should be able to see the status of the SD-WAN devices.



Next, go to **Monitoring** > **Performance** then select SD-WANs as a location filter. Verify that metrics are being collected.

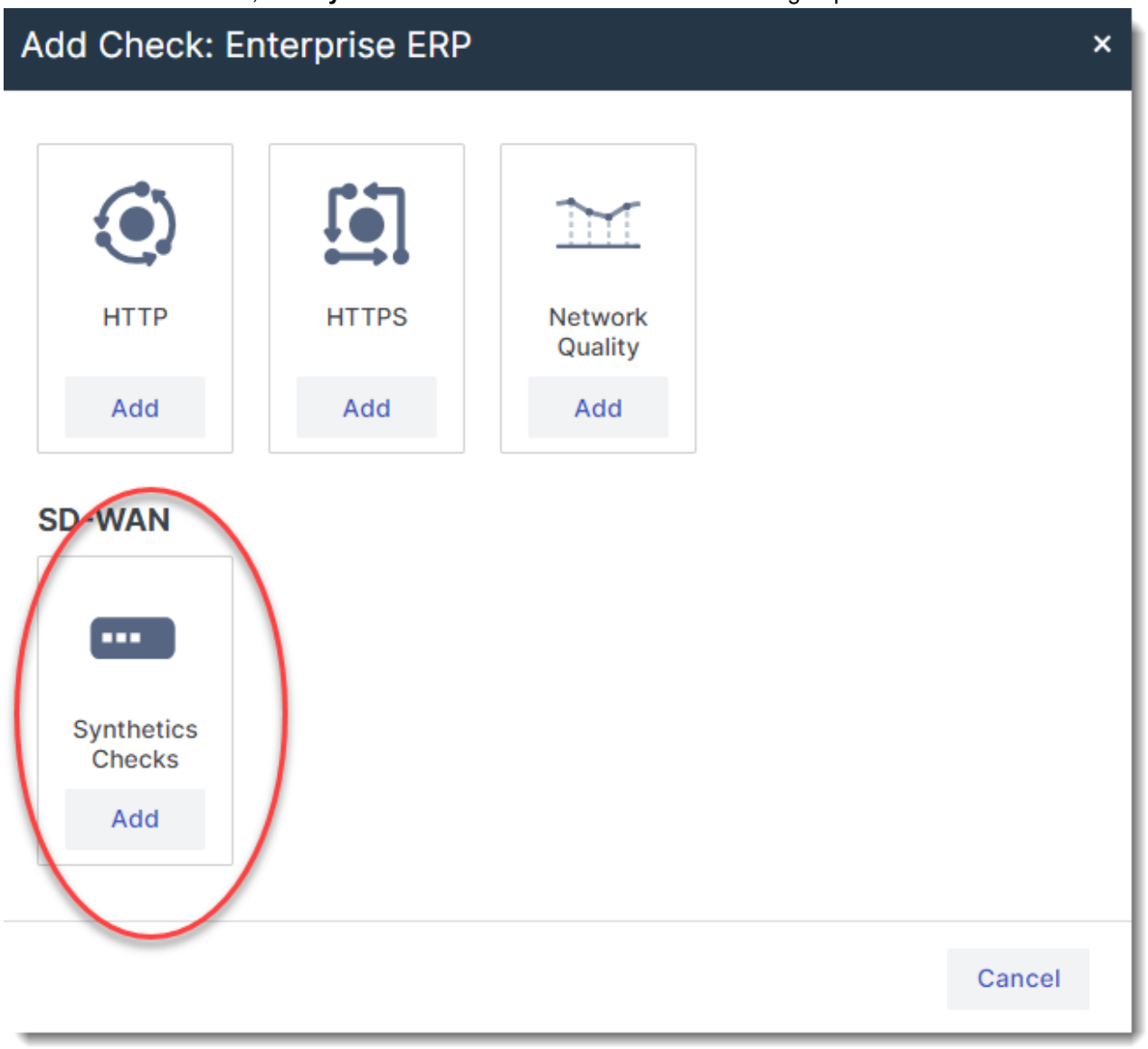


Add SD-WAN synthetic checks to an existing application

To add SD-WAN synthetic checks to an existing application, perform the following steps:

1. On the **Applications** page, select an application.
2. Click **Checks**.
3. Click **Add Check**.

4. In the **Add Check** modal, click **Synthetic Checks** located under the **SD-WAN** group.



5. Select one or more **Source SD-WANs** and **Target FQDNs/IPs**. You also have the option to edit the check frequency and the Alert Timeline.
6. Click **Save**.

Security Fabric

The Fortinet Security Fabric is a cybersecurity platform that interconnects different security solutions to deliver protection and visibility to your entire network architecture. See Fortinet Security Fabric | Administration Guide for more information.

Monitoring different Fabric environments

Fortinet environment

The FortiMonitor Fabric integration works best with network environments that are entirely composed of Fortinet devices. Connecting FortiMonitor to the FortiGate associated with the Fabric will provide FortiMonitor with access to all of the performance data and metadata to each and every device in the Fabric. By allowing FortiMonitor to join the Fabric, devices that are behind the FortiGate can be discovered, whether they're in the same physical location or connected via VPN. New Fortinet devices added to the network after the initial set up can also easily be discovered and monitored.

With this approach, you no longer have to set up SNMP on every firewall, switch, or access point in the network to gather metric data.

The integration to an active Security Fabric does bring certain prerequisites. These are detailed in Enable Security Fabric monitoring and is only available for FortiOS 7.0 and later.

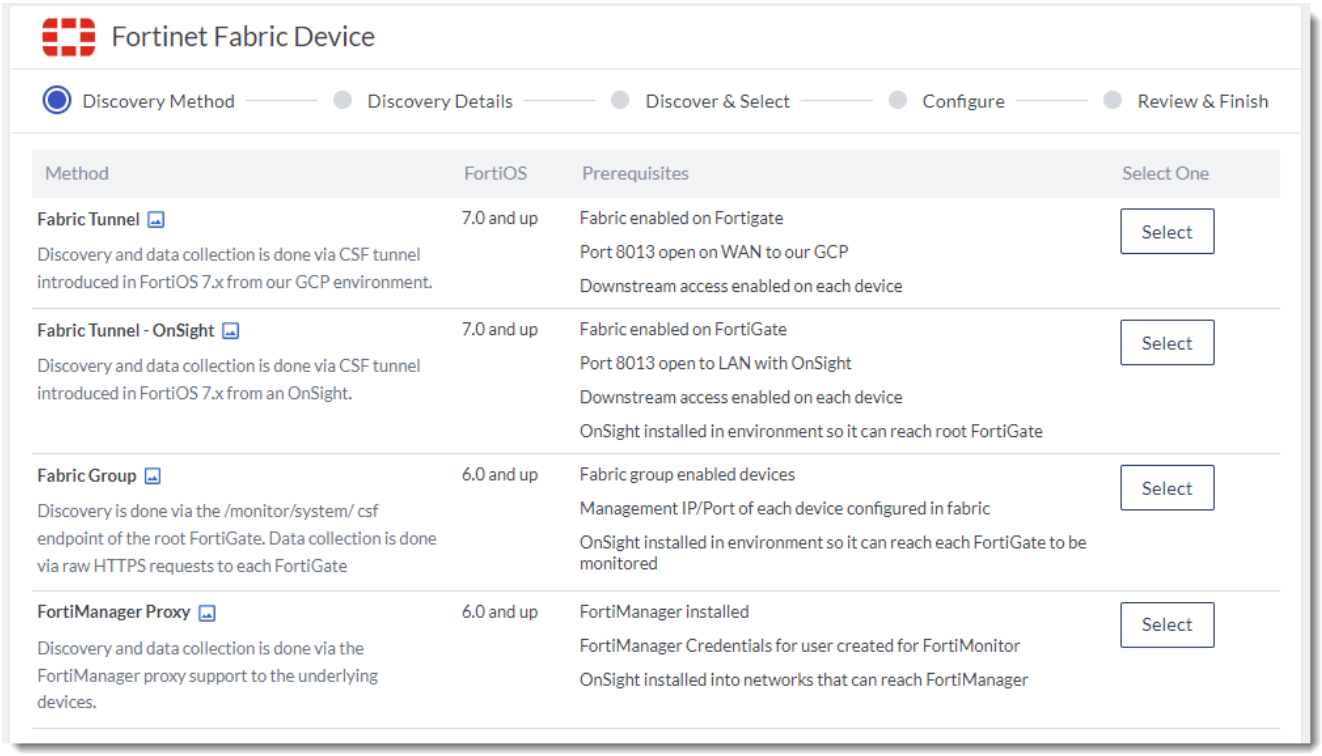
Combination of Fortinet and non-Fortinet devices

For environments that have a combination of Fortinet and non-Fortinet devices, the OnSight vCollector can be used. Note that as mentioned in the previous section, Fortinet devices that are part of the Fabric can be monitored by allowing FortiMonitor access to the Fabric.

For non-Fortinet devices, the OnSight must be deployed behind the firewall on your internal network. OnSight vCollector is the proxy into that network to allow FortiMonitor to collect metric data for your network devices. Windows, Linux and Unix hosts may be monitored with the FortiMonitor Agent. This approach does require that the OnSight vCollector is deployed and configured for maximum network visibility. You also have to either run OnSight discovery to discover all devices in the network or explicitly add all the devices in the network to the OnSight. Access credentials for the OnSight to do SNMP connections to every device in the network must also be configured.

Deployment options

To start onboarding your Fabric environment, click **Add** from the navigation menu. The following deployment options will be displayed. The table below explains each option in detail, including prerequisites and required information.



Deployment method	Description	Supported FortiOS version	Prerequisites	Required from user
Fabric Tunnel	Discovery and data collection is done via the CSF tunnel introduced in FortiOS 7.x from our GCP environment.	7.0 and up	• Security Fabric is enabled on root FortiGate • 8013 open on WAN to our GCP • Downstream access enabled on each device	• Root FortiGate IP/FQDN • Root FortiGate Port (almost always 8013) • Root SN
Fabric Tunnel - OnSight	Discovery and data collection is done via the CSF tunnel introduced in FortiOS 7.x from an OnSight.	7.0 and up	• Security Fabric is enabled on root FortiGate • 8013 open to LAN with OnSight	• Root FortiGate IP/FQDN • Root FortiGate Port (almost always 8013)

Deployment method	Description	Supported FortiOS version	Prerequisites	Required from user
			• Downstream access enabled on each device • OnSight installed in environment so it can reach the root FortiGate	• Root FortiGate SN • A selected OnSight
Fabric Group	Discovery of devices is done via the **/monitor/system/csf** endpoint on the root FortiGate. Collection is done via raw HTTPS queries to each FortiGate.	6.x	• Fabric group enabled devices • OnSight installed in environment so it can reach each FortiGate to be monitored • The management IP/Port of each device configured in Fabric	• Root FortiGate IP/FQDN • Port of root FortiGate HTTPS API • Root FortiGate serial number • Root FortiGate API key • A selected OnSight • After initial discovery, users will be prompted for an API key for each FortiGate discovered.
FortiManager Proxy	Discovery and data collection is performed entirely via the FortiManager proxy to the underlying devices.	6.0 and up	• FortiManager is installed • FortiManager credentials for user created for FortiMonitor • OnSight is installed into network that can reach FortiManager	• Selected OnSight • FortiManager IP/FQDN • FortiManager HTTPS Port • FortiManager SN • FortiManager username and password

Note: FortiGate models with 2 GB RAM cannot be the root of the Security Fabric topology or any mid-tier part of the topology. They can only be configured as downstream devices in a Security Fabric or standalone devices. For more information, see Fortinet Security Fabric.

FortiMonitor integration

FortiMonitor's Security Fabric integration allows you to discover and monitor your devices in your Fabric environment. The full Security Fabric topology and device lists are pulled in and monitored through FortiGate's Security Fabric API.

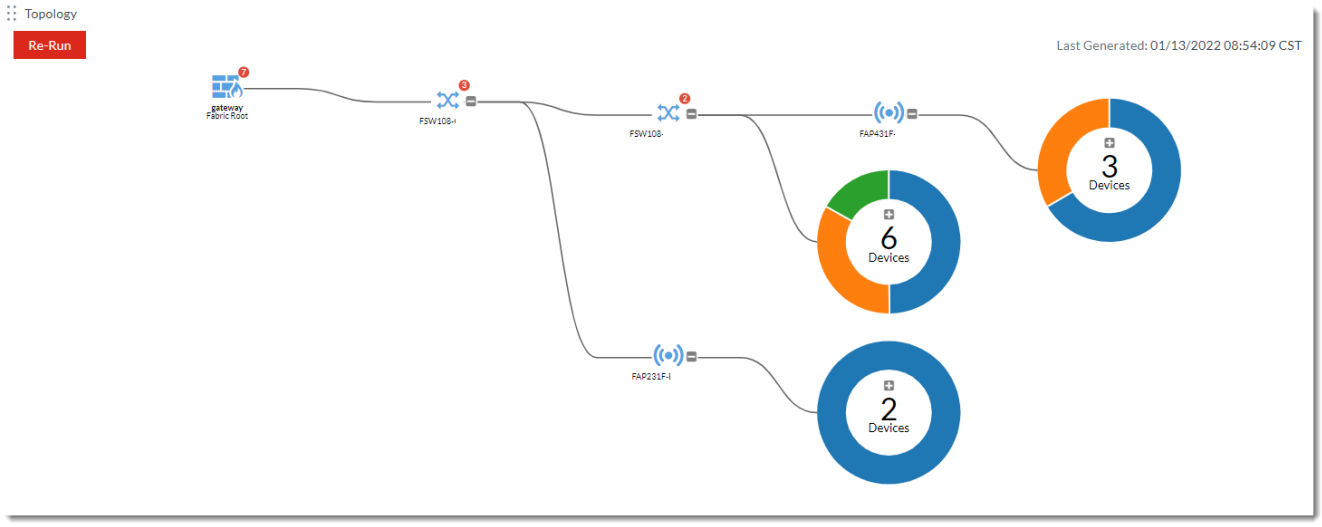
Fabric device monitoring supports the following:

Full template support

Create templates and apply them to Fabric devices that you want to monitor in the same way. Default templates are also available for each type of Fabric device. See Templates to learn how to use templates.

Fabric Topology dashboard widget

The full Security Fabric topology from the FortiGate can be displayed using the Fabric topology widget. For more information, see Fabric topology widget.

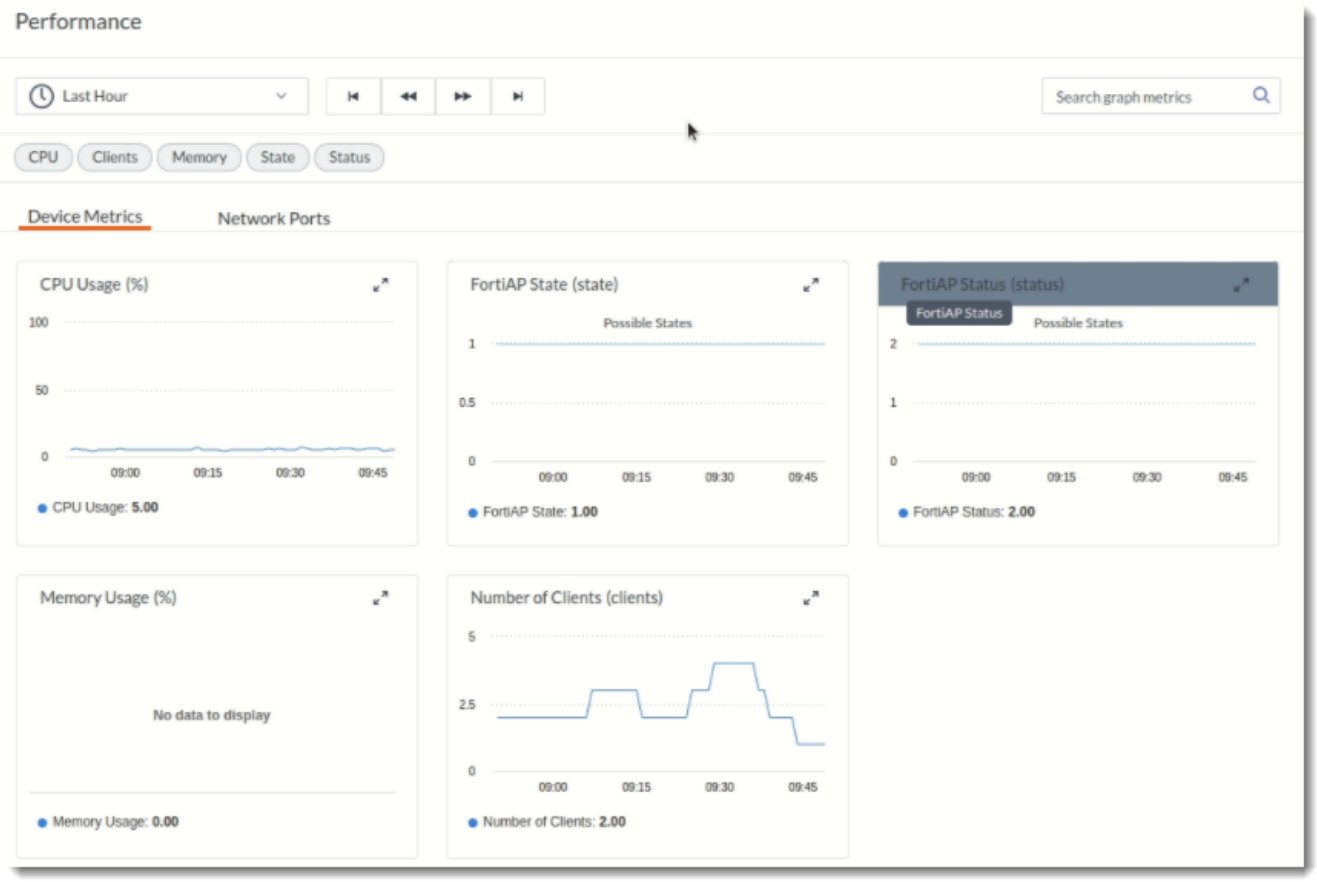


Network Interfaces dashboard widget

The Network Interfaces widget provides a table displaying the status and bandwidth utilization of network device interfaces. For more information, see Network Interfaces.

Network device performance monitoring

The performance of each Fabric device is displayed in the Performance section of the device's **Instance Details page**.



Note on FortiAPs and FortiSwitches

Metric data from FortiAPs and FortiSwitches are gathered from their parent FortiGate. Only FortiAPs and FortiSwitches that are managed by FortiGate are supported.

Supported metrics

In addition to SNMP metrics, the following health checks are also supported:

Device health for FortiGates

- Configured Interface Up/Down
- SD-WAN stats
- CPU/Memory Health
- Security rating
- Client Lists
- DHCP Pool info
- Firewall Bytes
- Firewall Hit Count
- Firewall packets
- Firewall Sessions
- Interface DHCP

Device health for FortiSwitches

- FortiLink Up/Down
- Port Status
- Connected Clients
- CPU/memory Usage
- Errors
- PoE Usage
- State

Device health for FortiAPs

- FortiLink Up/Down
- Traffic Stats
- Connected Clients
- CPU/Memory Usage
- Channel Utilization
- MAC Errors
- TX Errors
- Auth Errors
- DHCP status

Device health for FortiExtender

- LTE RSSI
- LTE SINR

- Signal RSRP
- Signal RSRQ
- Signal Strength
- CPU Usage
- Memory Usage
- Data Usage
- SIM Active

Using FortiManager

- Bandwidth In
- Bandwidth Out
- CPU
- Disk
- Errors In
- Errors Out
- Interface DHCP
- Interface Status
- Memory
- Packets In
- Packets Out
- SD-WAN Bandwidth In
- SD-WAN Bandwidth Out
- SD-WAN Jitter
- SD-WAN Latency
- SD-WAN Packet Loss
- SD-WAN Packets In
- SD-WAN Packets Out
- SD-WAN SLA Targets Met
- SD-WAN Sessions
- SD-WAN Status
- Sessions

Using SNMP

- Active SSL Tunnels
- Active Web Sessions
- Admin Status
- Antivirus Events Triggered
- BGP Peer State
- Bandwidth In
- Bandwidth Out

- Bytes Processed
- CPU Usage
- Cluster Member Bandwidth
- Cluster Member CPU Usage
- Cluster Member Memory Usage
- Cluster Member Packets Processed
- Cluster Member Session Count
- Discard Packets In
- Discard Packets Out
- Disk Usage
- Errors In
- Errors Out
- Explicit Proxy Voliations
- Firewall Packet Matches
- HA Cluster Priority
- HA Sync Status
- IDS/IPS Events Triggered
- IM Connections Blocked
- IPSec Tunnel Status
- IPsec Tunnel Lifetime
- IPsec Tunnel Timeout
- Intrusions Blocked
- Low Mem Usage
- Maximum VDOMs
- Memory Usage
- Operational Status
- Packets Dropped
- Packets Received
- Packets Transmitted
- Processor Usage
- SD-WAN Link Bandwidth In
- SD-WAN Link Bandwidth Out
- SD-WAN Link Bi-Directional Bandwidth
- SD-WAN Link Packet Loss
- SD-WAN Link Packets In
- SD-WAN Link Packets Out
- SD-WAN Link State
- SSL VPN Sessions
- SSL VPN Status
- Session Clash Rate
- Session Count
- Tunnel Bandwidth In
- Tunnel Bandwidth Out

- VDOM Count
- VDOM Enabled
- VDOM Ops Mode
- Viruses Blocked
- WAN Tunnel Count
- fgSysSesCount
- fgVWLHealthCheckLinkJitter
- fgVWLHealthCheckLinkLatency
- fgVpnTunEntLifeBytes

Limitations

Fabric integration with multiple VDOMs for FortiOS prior to 7.4 is not supported due to limitations in API support. The following metrics that support VDOM-level reporting only work on FortiOS 7.4 and above:

- BGP peer metrics
- Firewall policies
- SD-WAN
- IPsec
- SSL VPN

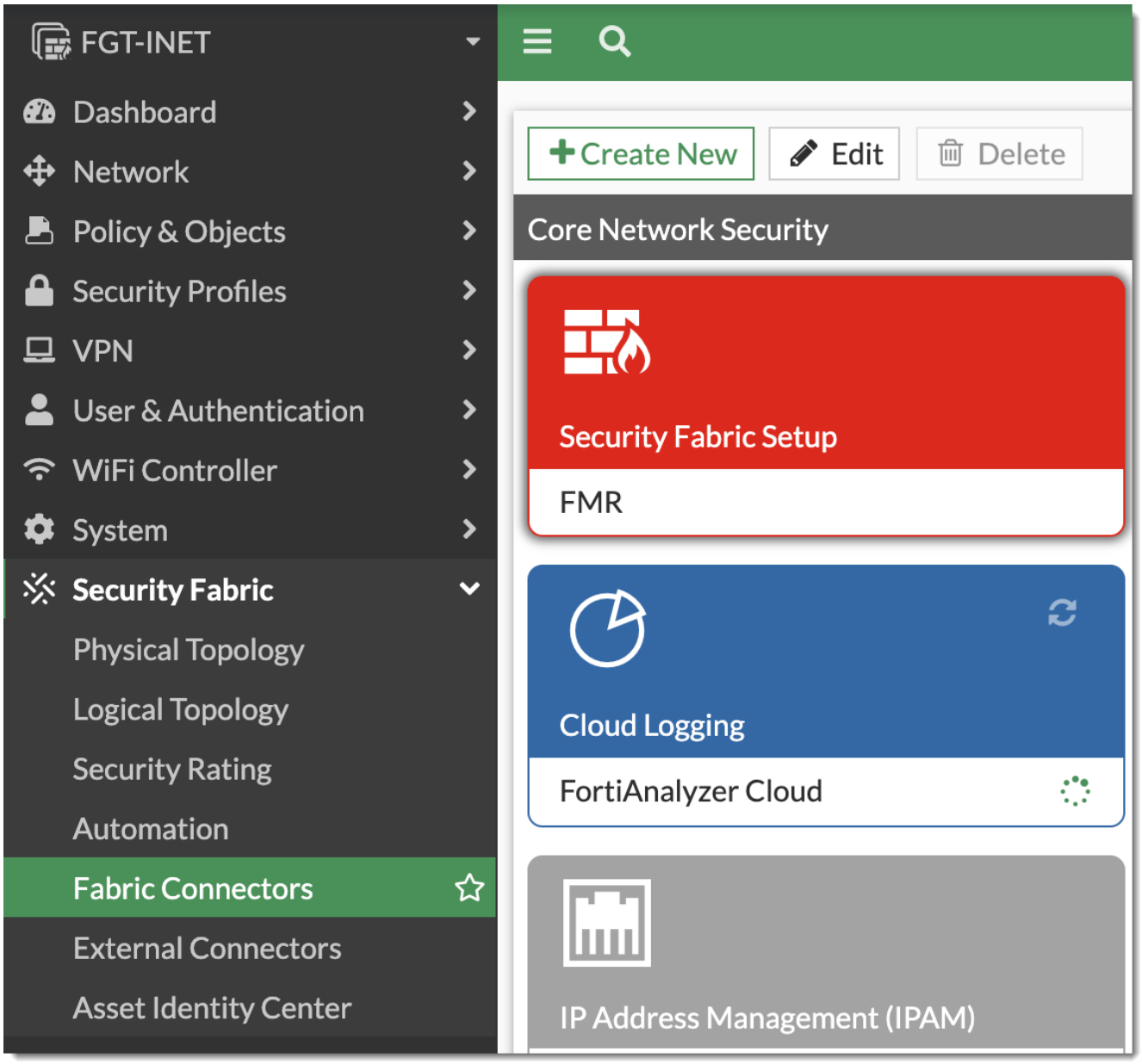
Fabric Tunnel connected to FortiMonitor cloud (FortiOS 7.0 and newer)

This article provides the steps on how to configure a Fortinet Security Fabric environment for monitoring with FortiMonitor. See Security Fabric for an overview of the solution.

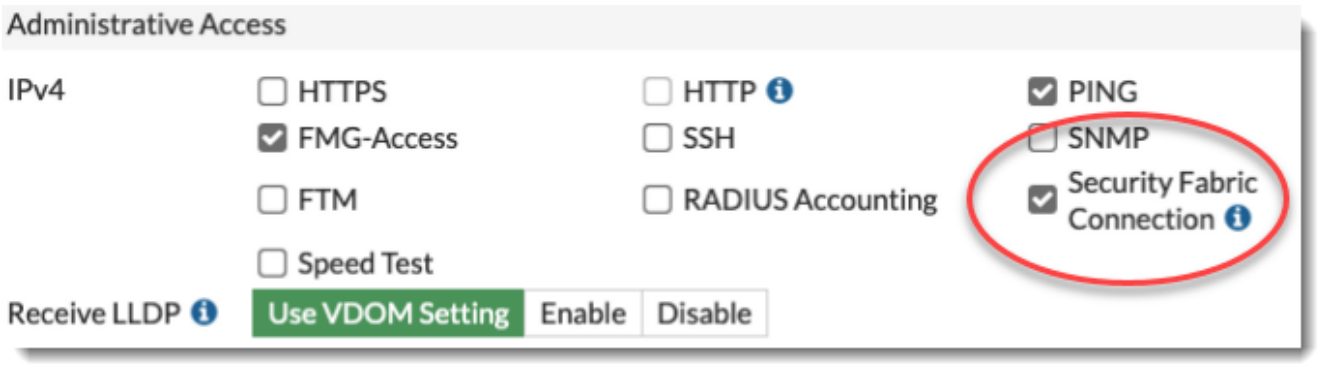
Note: For FortiOS 6.x deployments, see Fabric 6.x integration.

Prerequisites

- FortiGate Firewall, running FortiOS 7.0.0 or later
- Active Security Fabric (Devices should be present under Fabric Root). See Fortinet Security Fabric Administration Guide.



- Administrative access to interfaces within FortiGate. See Interface settings.
- For the public-facing (WAN) interface(s), enable **Security Fabric Connection**. This enables FortiTelemetry and CAPWAP.



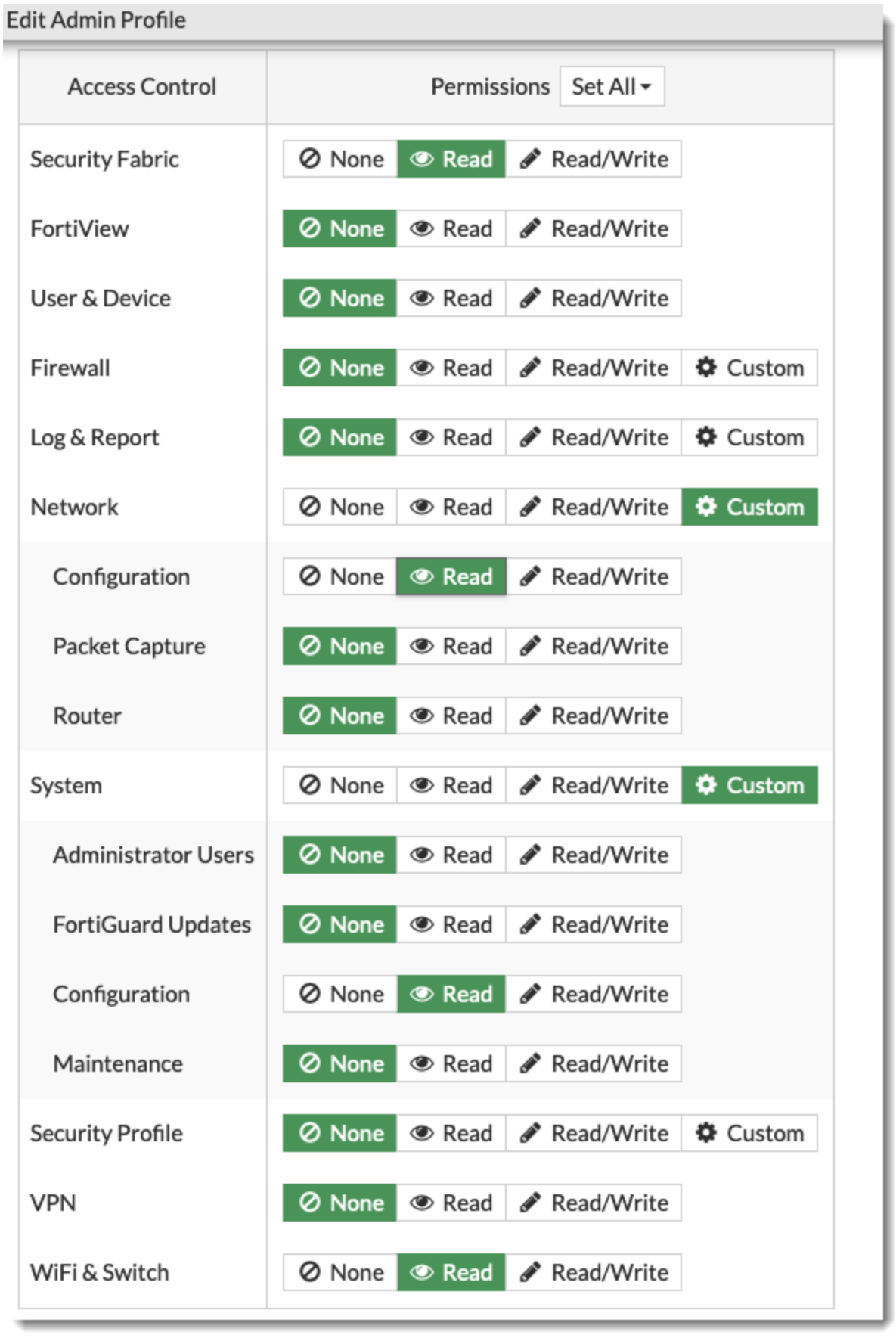
Note:

Security Fabric connections are inbound to TCP port 8013 from these IP addresses:

- 104.197.35.194/32
- 35.185.29.9/32
- GCP IP for EU01 region: 34.159.200.72/32

Configure your firewall to allow inbound traffic to TCP 8013. If you want to limit access to just these two IPs, configure your FortiGate. Note that these IP addresses are subject to change and you will be notified prior to any such change.

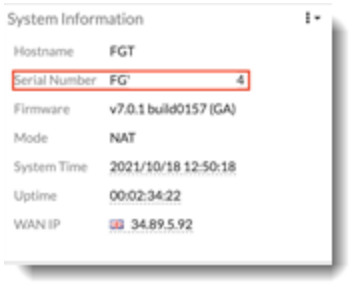
- Create an admin profile by going to **System** > **Admin Profiles** on the FortiGate. For example, **fabric_admin_ro**. It should have the following settings:


Edit Admin Profile
Access Control
Permissions
Set All
Security Fabric
None
Read
Read/Write
FortiView
None
Read
Read/Write
User & Device
None
Read
Read/Write
Firewall
None
Read
Read/Write
Custom
Log & Report
None
Read
Read/Write
Custom
Network
None
Read
Read/Write
Custom
Configuration
None
Read
Read/Write
Packet Capture
None
Read
Read/Write
Router
None
Read
Read/Write
System
None
Read
Read/Write
Custom
Administrator Users
None
Read
Read/Write
FortiGuard Updates
None
Read
Read/Write
Configuration
None
Read
Read/Write
Maintenance
None
Read
Read/Write
Security Profile
None
Read
Read/Write
Custom
VPN
None
Read
Read/Write
WiFi & Switch
None
Read
Read/Write

- Within **Fabric Connectors > Security Fabric Setup**, **Downstream rest API Access** must be enabled, and the Admin profile set to the profile you created with the above permissions, for example: **fabric_admin_ro**



- Take note of the device **Serial Number** in the **Status Dashboard**.



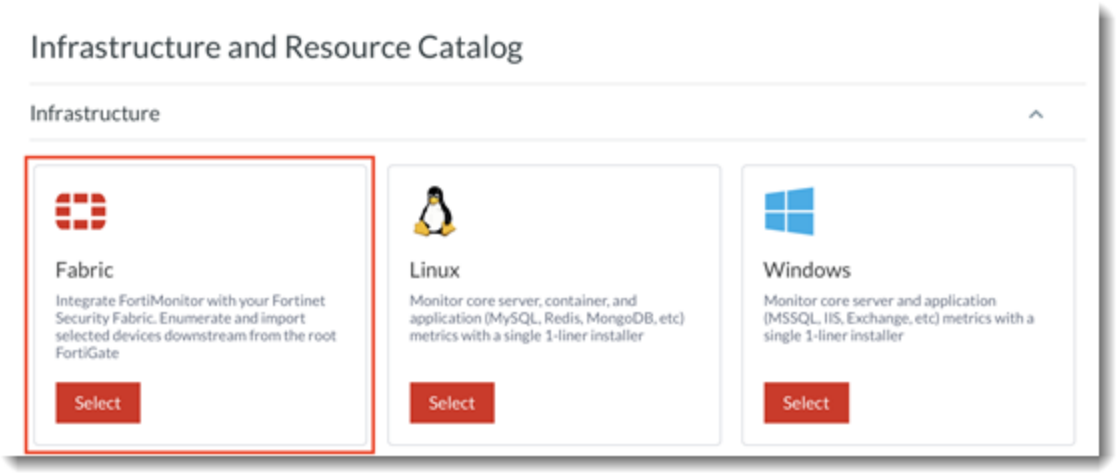
Important note: At this time, once a Fabric environment has been integrated with your FortiMonitor, it may not subsequently be integrated into another FortiMonitor account.

FortiMonitor configuration

After completing the prerequisites, you can now connect FortiMonitor to your Fabric environment.

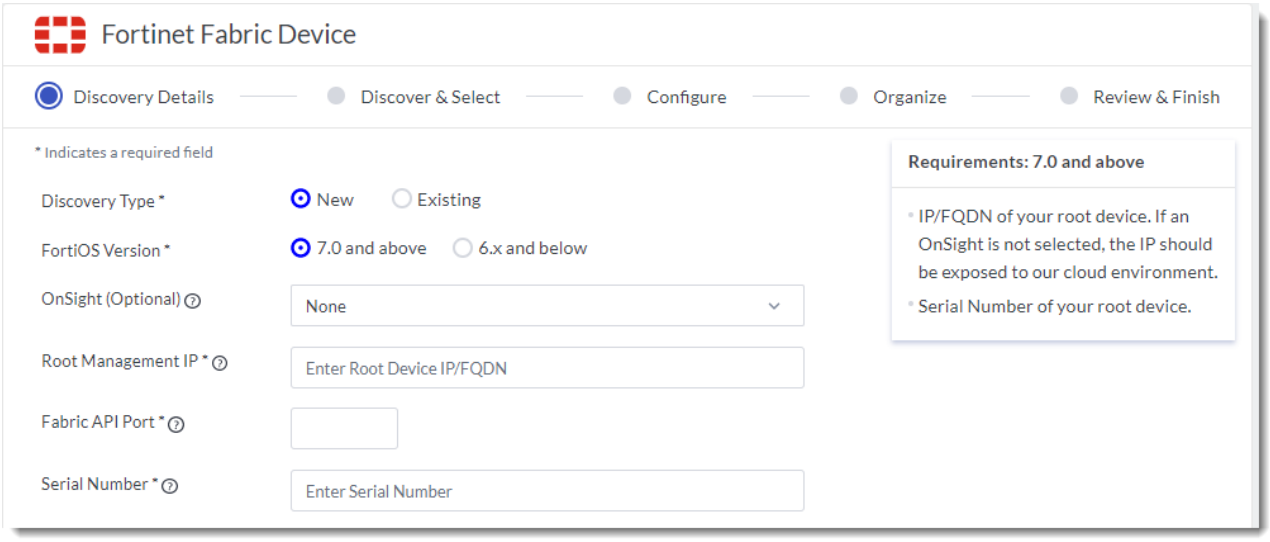
Note: Before connecting FortiMonitor to your Fabric environment, ensure that you have a fully functioning and active Security Fabric environment. Please see **Prerequisites**.

1. Log in to FortiMonitor (https://fortimonitor.forticloud.com/).
2. From the navigation menu, click **Add**. **The Infrastructure and Resource Catalog is displayed**.
3. Select **Fabric** from the **Infrastructure** section of the catalog.



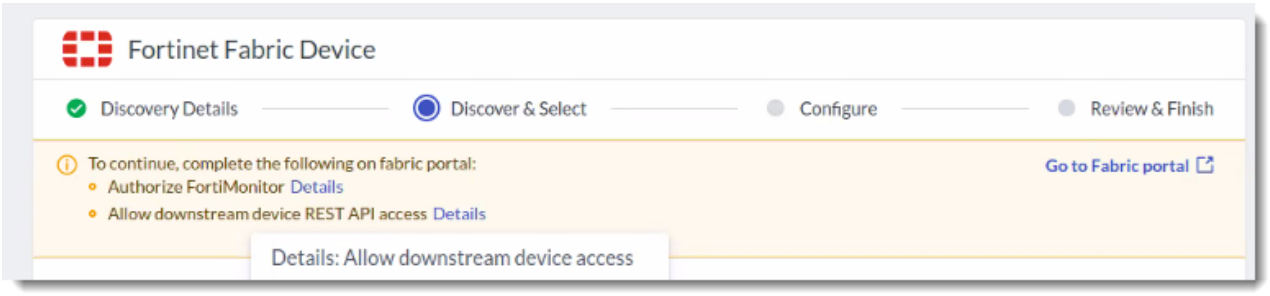
4. Select **Fabric Tunnel**.

5. From here, follow the on-screen prompts.

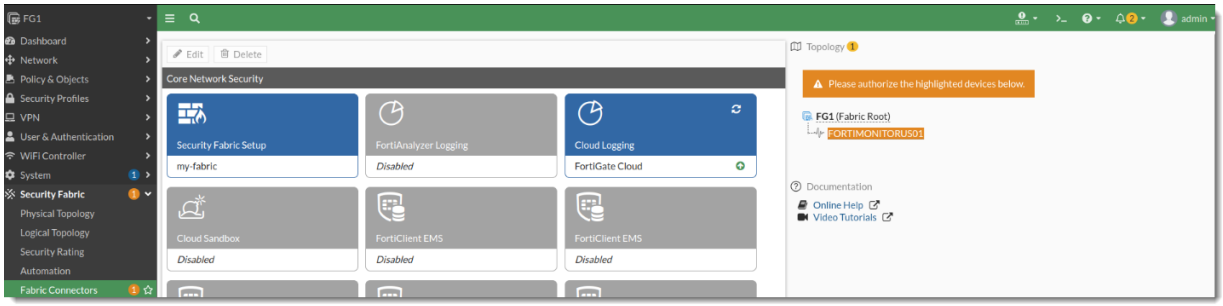


   a. Discovery type - Select either **New** or **Existing**.
   b. FortiOS Version - Select the FortiOS version of the Fabric device
   c. OnSight (Optional) - Select an OnSight. This OnSight will be used to monitor the FortiGates and associated devices. See Fabric 7.x OnSight proxy for more information.
   d. Root Device IP/FQDN - Public IP address where your root FortiGate can be reached from the FortiMonitor Public Cloud.
   e. Fabric API Port - Enter the target port for the Fortinet Security Fabric Connection.
   f. Serial Number - Enter the serial number for the FortiGate Security appliance. See Prerequisites.

6. At this point, the FortiMonitor certificate will require Authorization within FortiOS.



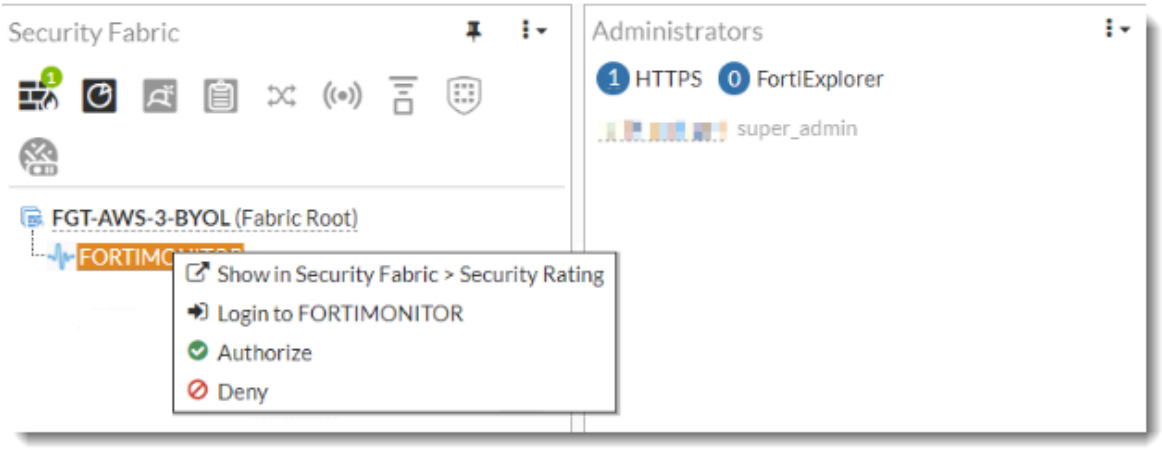
7. Click **Go to Fabric portal** to authorize the certificate.
   a. For FortiOS lower than 7.2.4, perform this step to authorize FortiMonitor. The following screenshot, taken from the FortiGate GUI, shows **FORTIMONITORUS01** under the Fabric Root.



b. For FortiOS 7.2.4 and above, perform the following steps to authorize FortiMonitor.

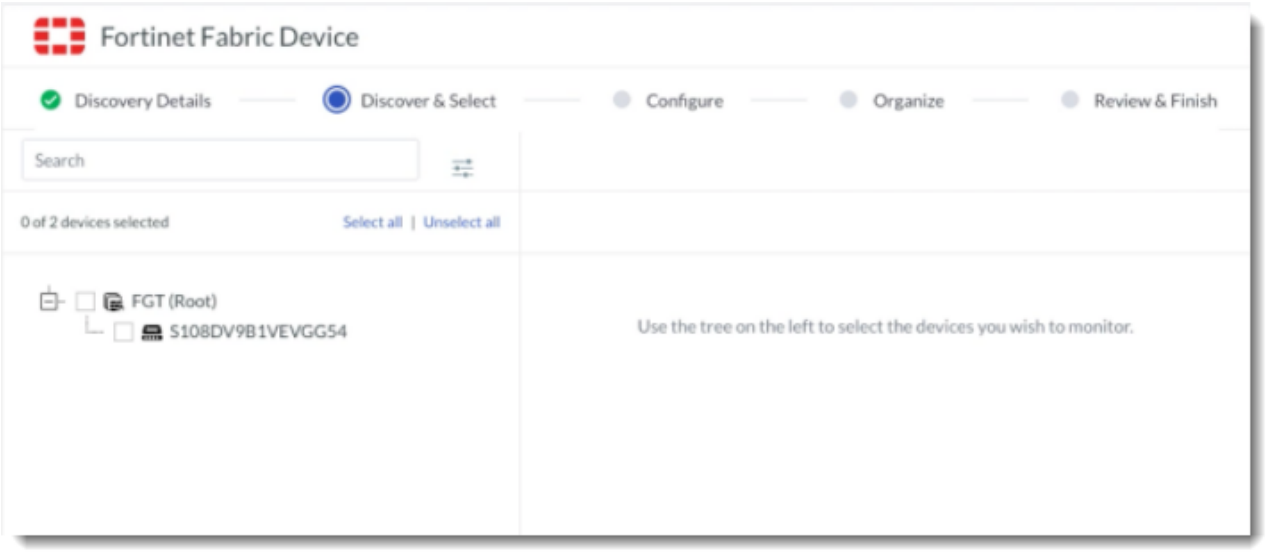
i. Go to **Dashboard** > **Status** and locate the **Security Fabric** widget.

ii. In the topology tree, click the highlighted FortiMonitor and select **Authorize**.

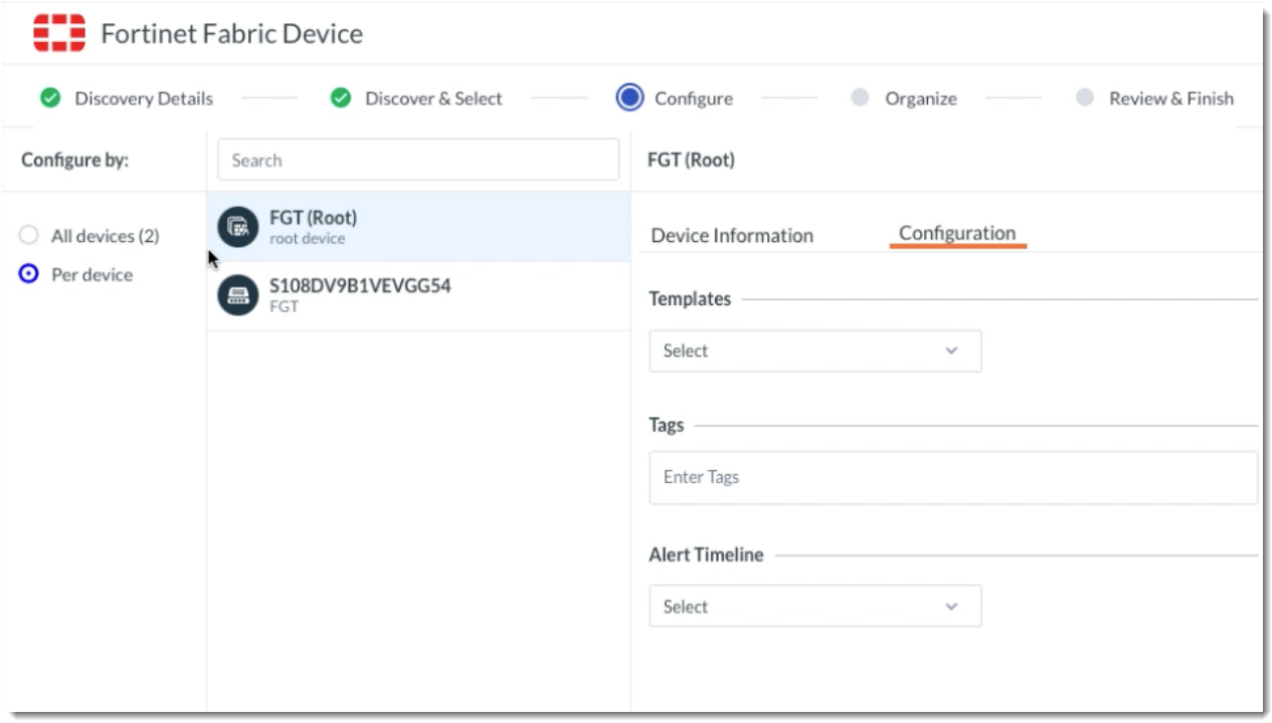


You also have the option to pre-authorize FortiMonitor. For more information on pre-authorization, see Configuring the root FortiGate and downstream FortiGates.

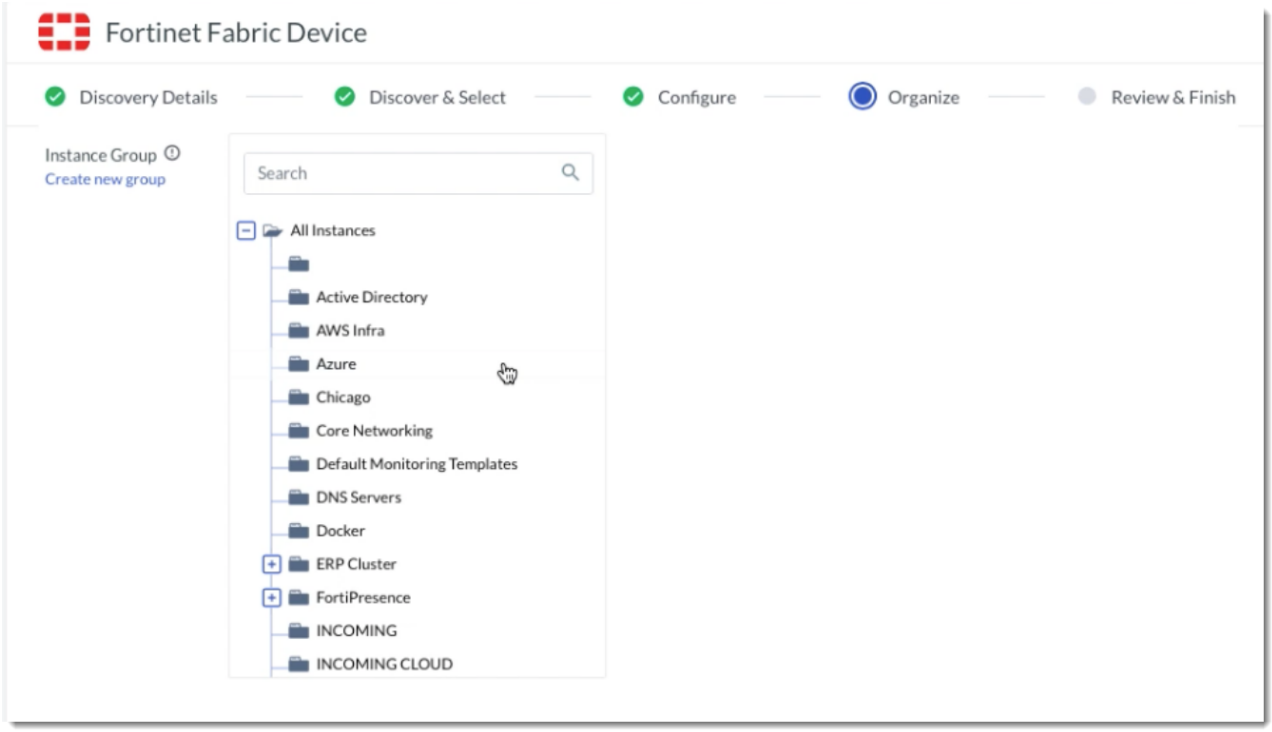
8. Once authorized, you can then begin the process of device selection and import.



9. Once device selection is complete, you may now assign tags, monitoring templates, and alert timelines.

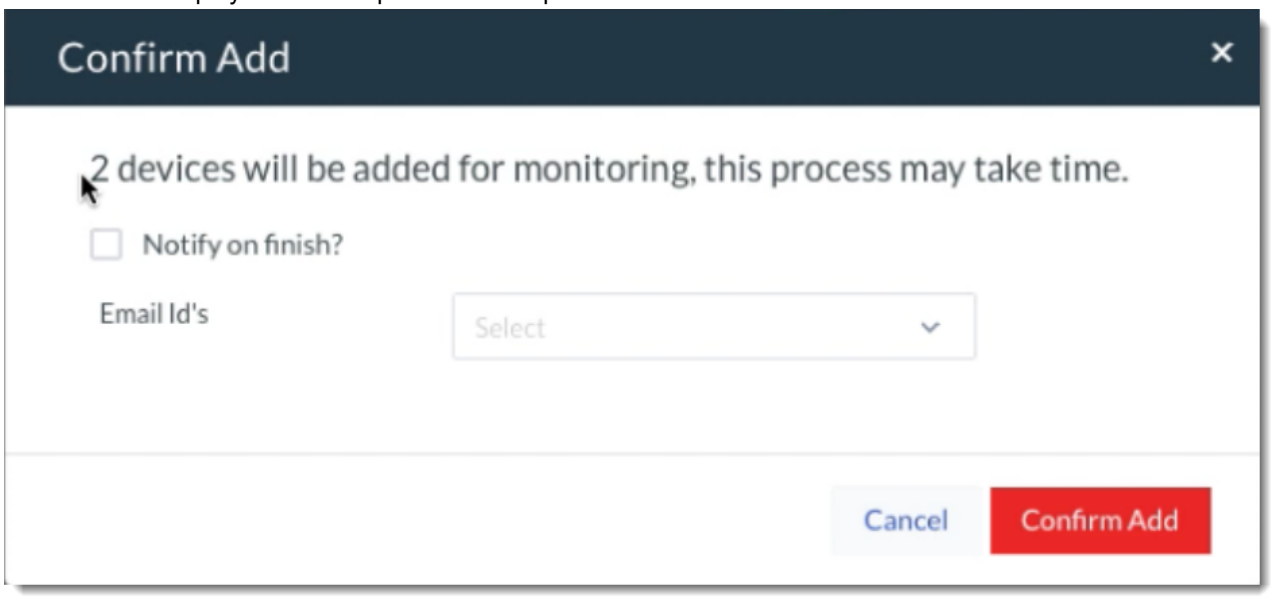


10. Next, configure the **Instance Group**, which is the logical organization of the monitored instances within FortiMonitor.



11. A summary view of the configuration will be displayed before committing the changes.
12. After selecting Finish to add the devices, you have the option to be alerted upon completion.
Note: Depending on the number of devices in your Fabric environment, this process may take a few minutes. A

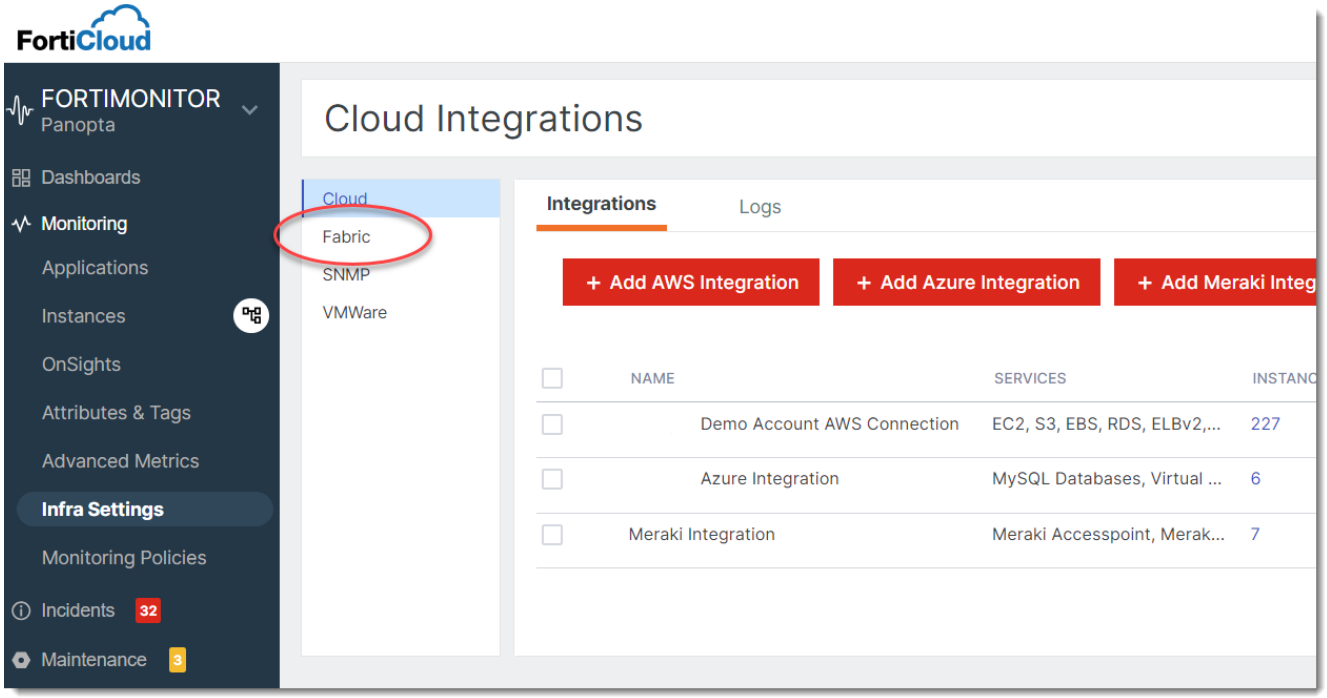
banner will be displayed once the process is complete.



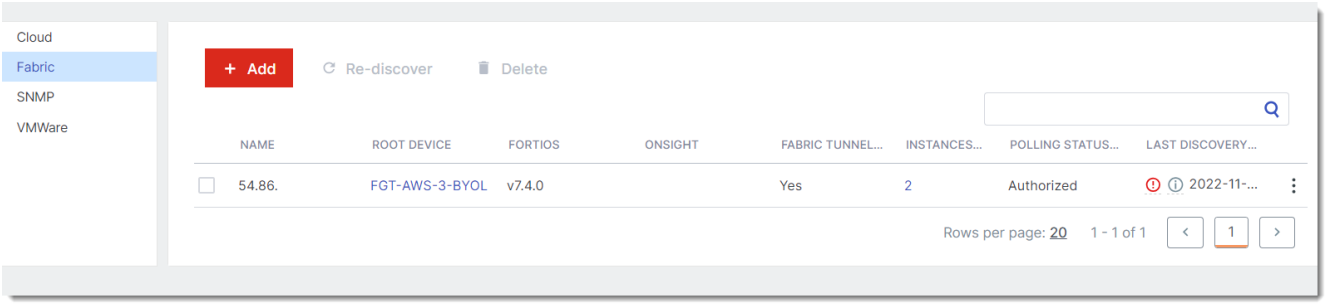
13. Once the process completes, the individual devices may be located on their respective instance pages.
14. You can manage the Fabric integration by going to **Settings > Fabric Settings**.

Manage Fabric connections

Fabric connections can be managed by clicking **Monitoring** > **Infra Settings** > **Fabric**.

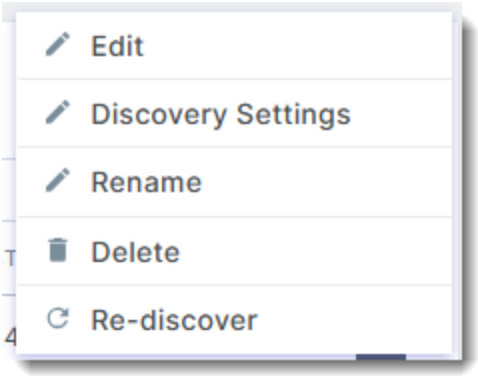


The Fabric Connections management page is displayed.



From this page, you can perform the following actions on each Fabric connection by clicking the **3-dot menu**:

- Edit the configuration of the Fabric connection
- Rename the Fabric connection
- Modify the discovery frequency
- Delete the Fabric connection
- Re-discover the Fabric connection

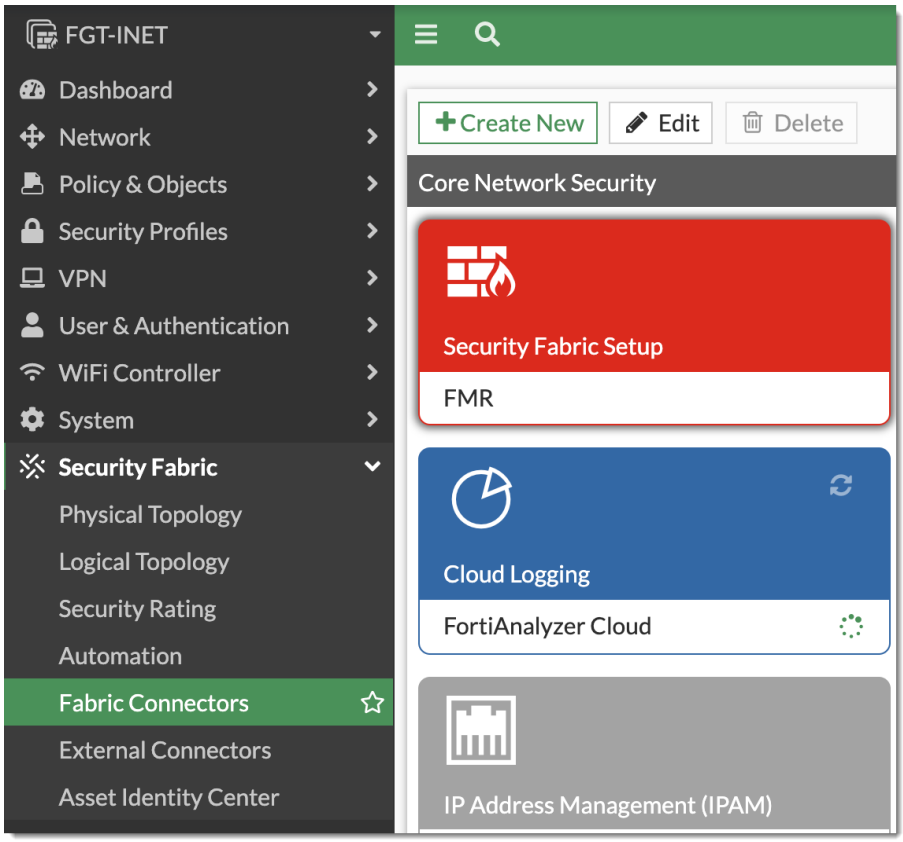


Fabric 7.x OnSight proxy

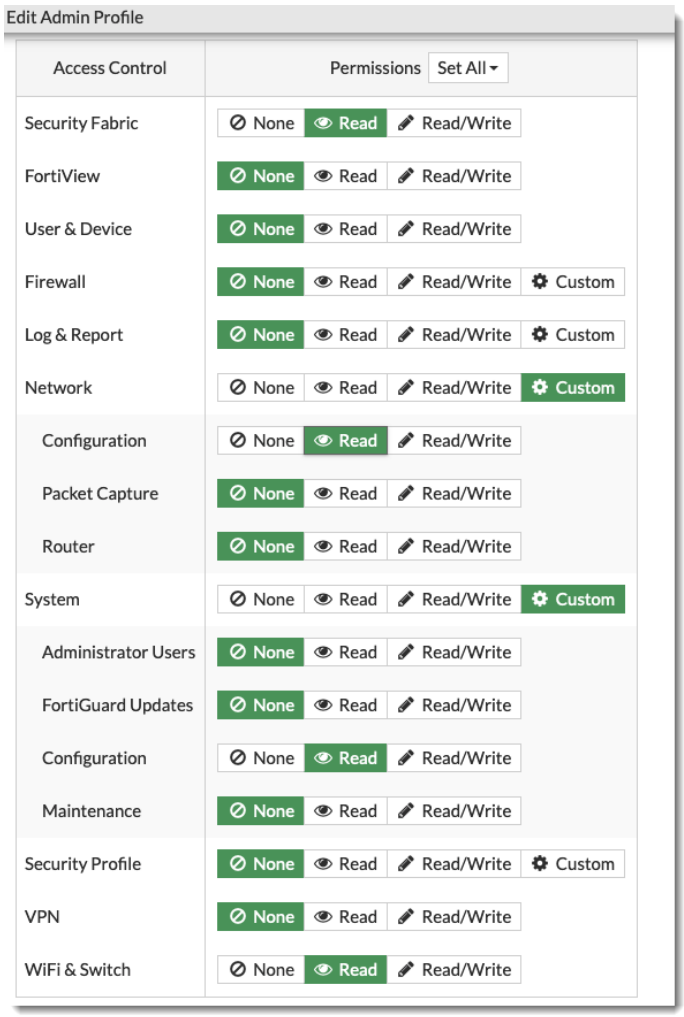
This article provides the steps on how to configure monitoring for Fabric 7.x devices using an OnSight vCollector.

Prerequisites

- FortiGate Firewall, running FortiOS 7.0.0 or later
- Active Security Fabric (Devices should be present under Fabric Root). See Fortinet Security Fabric Administration Guide.



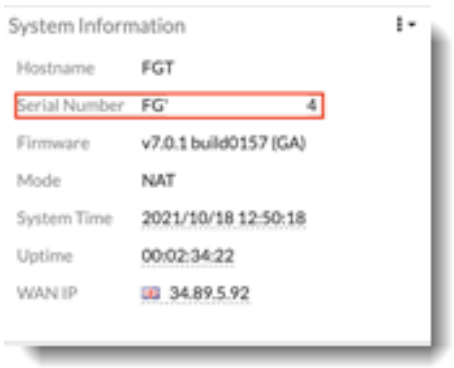
- Security Fabric connections are inbound to TCP port 8013 from the IP address of the OnSight. Configure your firewall to allow inbound traffic to TCP 8013.
- Create an admin profile by going to **System** > **Admin Profiles** on the FortiGate. For example, **fabric_admin_ro**. It should have the following settings:
  Note: By default, the scope of the admin profile is **Virtual Domain** resulting to FortiMonitor only seeing information only from the VDOM it connects to. You must set this to **Global**.



- Within **Fabric Connectors > Security Fabric Setup**, **Downstream REST API Access** must be enabled, and the Admin profile set to the profile you created with the above permissions, for example: **fabric_admin_ro**.
  Note: **Downstream Rest API access** must be enabled on **ALL** FortiGates that you want to monitor.



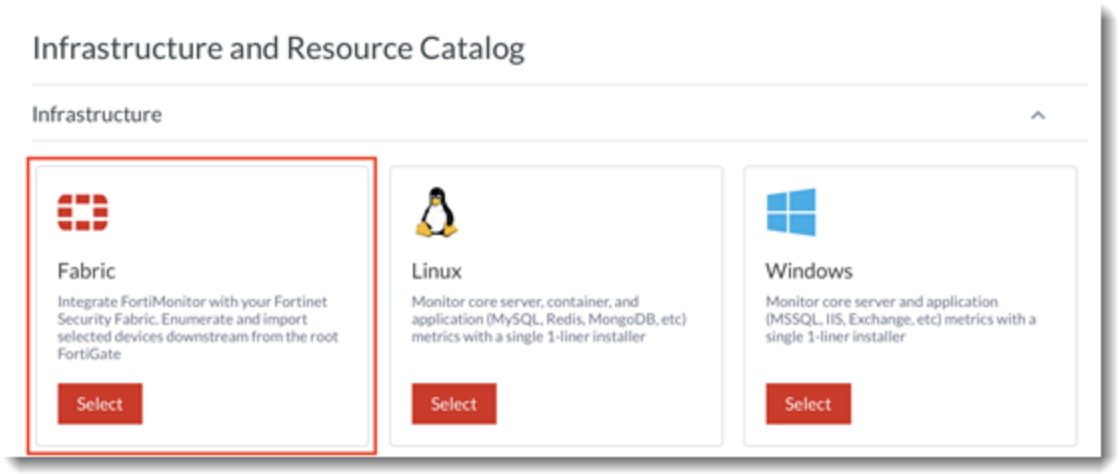
- Take note of the root FortiGate's **Serial Number** in the **Status Dashboard**.



Important note: At this time, once a Fabric environment has been integrated with your FortiMonitor, it may not subsequently be integrated into another FortiMonitor account.

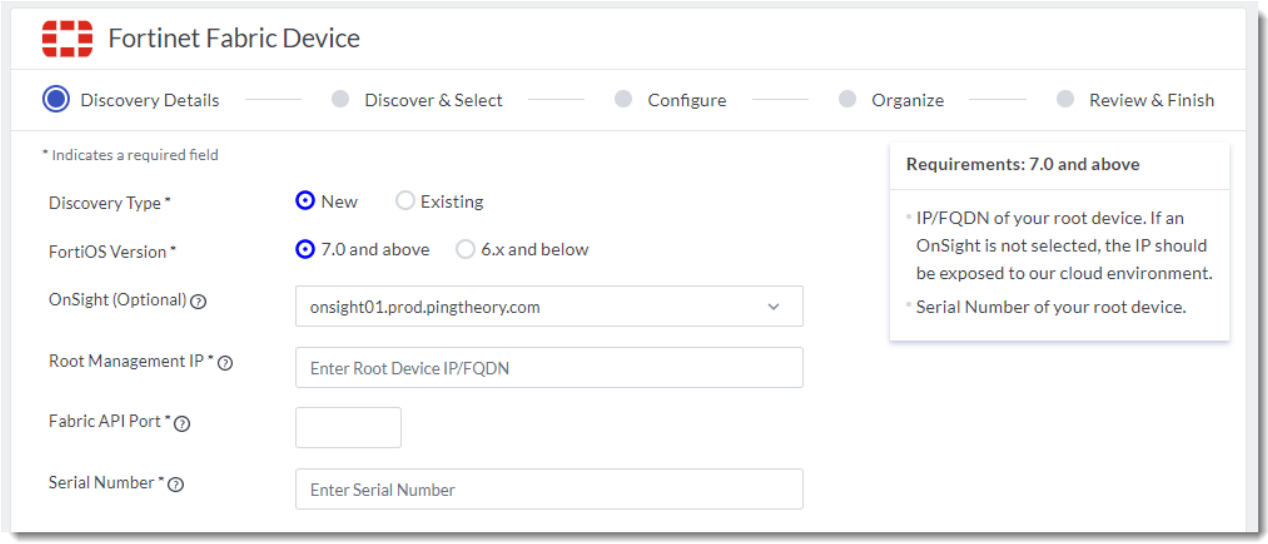
FortiMonitor Configuration

1. Log in to FortiMonitor (https://fortimonitor.forticloud.com/).
2. From the navigation menu, click **Add**. The **Infrastructure and Resource Catalog** is displayed.
3. Select **Fabric** from the **Infrastructure** section of the catalog.



4. Select **Fabric Tunnel - OnSight**.

5. From here, follow the on-screen prompts.



a. Discovery type - Select **New**.

b. FortiOS Version - Select **7.0 and above**.

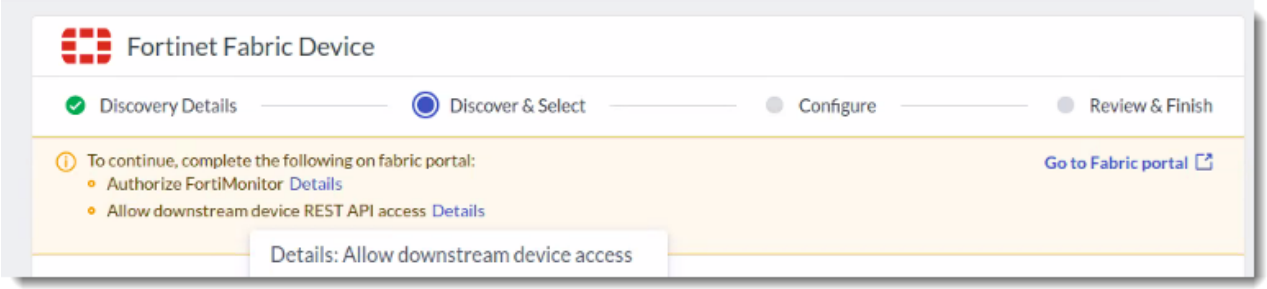
c. OnSight (Optional) - Select an OnSight. This OnSight will be used to monitor the FortiGates and associated devices. See OnSight vCollector for more information.

d. Root Management IP - Enter the IP address where your root FortiGate can be reached from the selected OnSight.

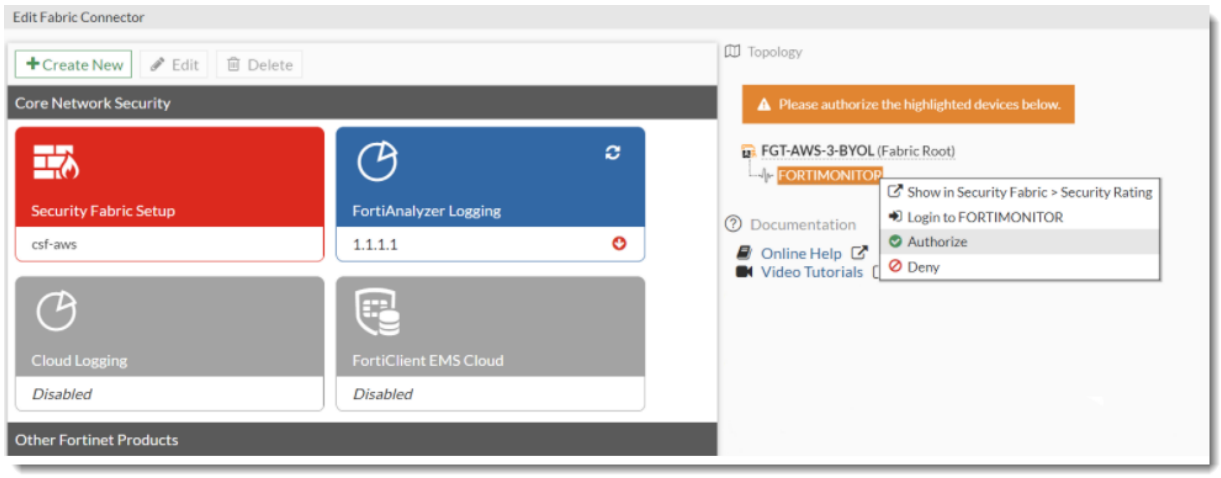
e. Fabric API Port - Enter the target port for the Fabric connection. The default is 8013. To ensure that you have the correct port, run `get system csf` on the root FortiGate.

f. Serial Number - Enter the serial number for the FortiGate Security appliance.

6. At this point, the FortiMonitor Certificate will require Authorization within FortiOS.



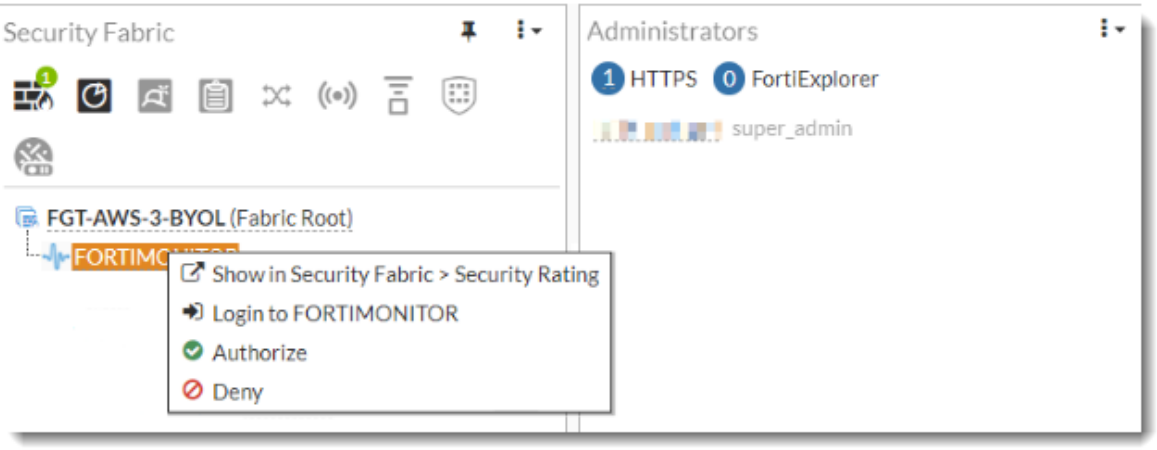
7. Click **Go to Fabric portal** to authorize the certificate.
   a. For FortiOS lower than 7.2.4, perform this step to authorize FortiMonitor. The following screenshot, taken from the FortiGate GUI, shows **FORTIMONITOR** under the Fabric Root.



b. For FortiOS 7.2.4 and above, perform the following steps to authorize FortiMonitor.

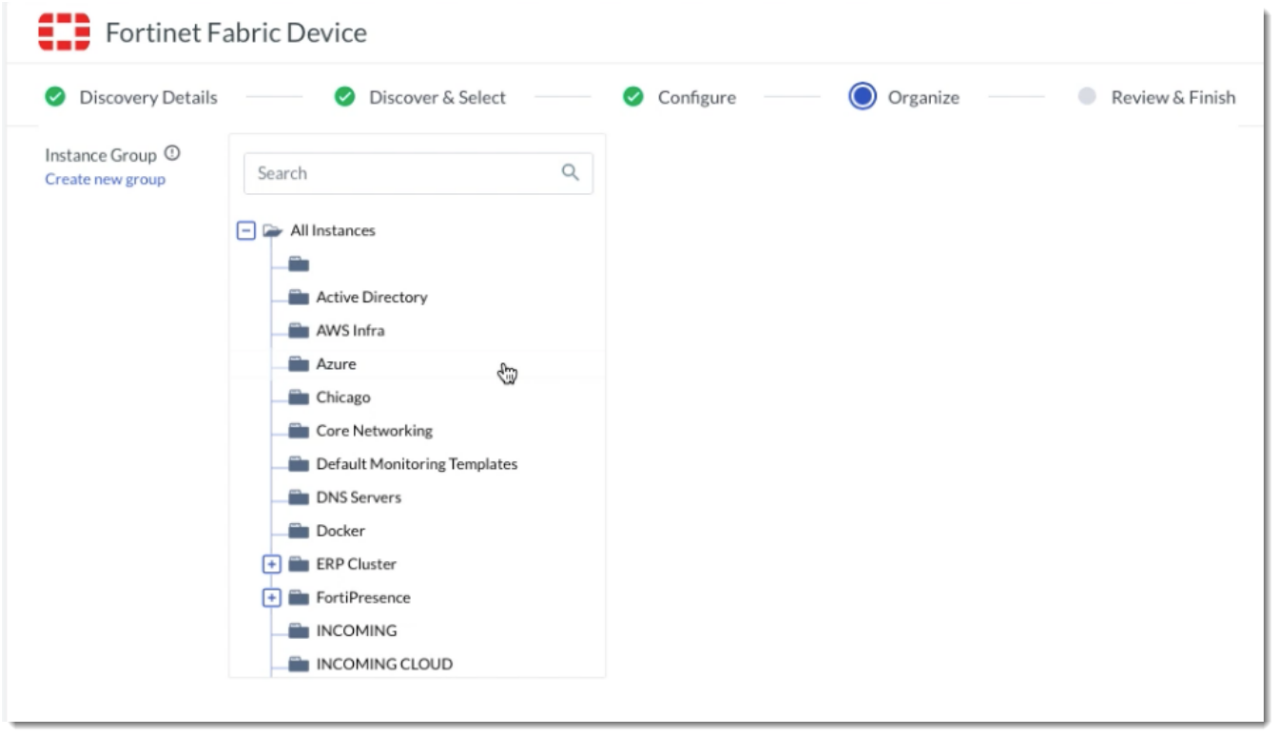
i. Go to **Dashboard** > **Status** and locate the **Security Fabric** widget.

ii. In the topology tree, click the highlighted FortiMonitor and select **Authorize**.

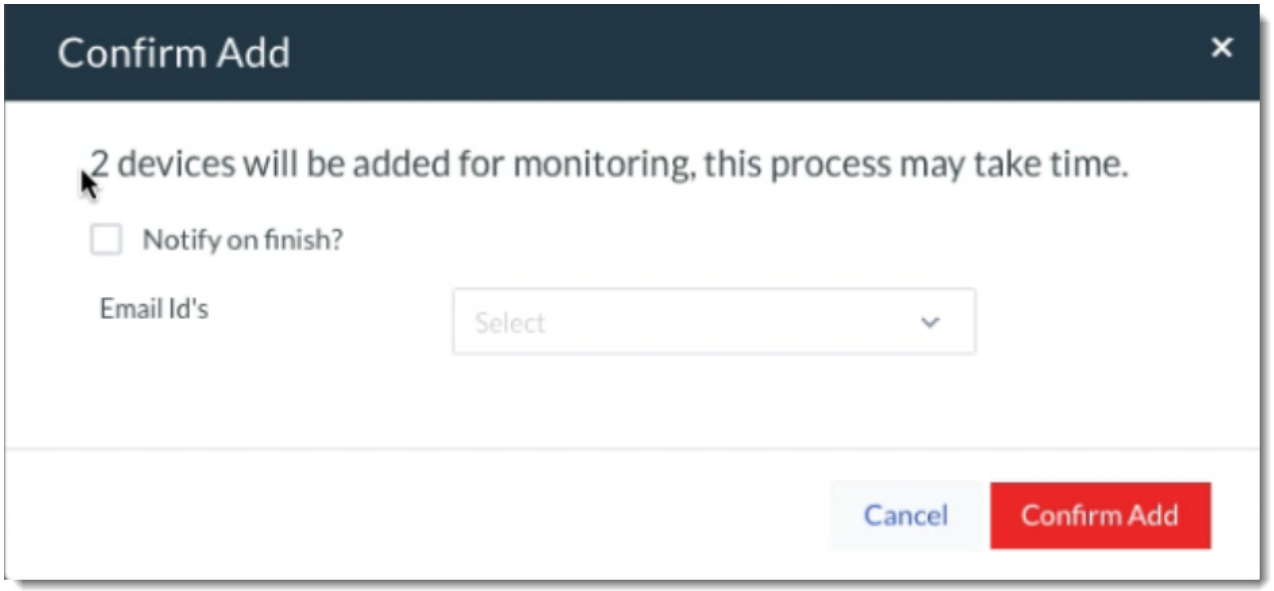


You also have the option to pre-authorize FortiMonitor. For more information on pre-authorization, see Configuring the root FortiGate and downstream FortiGates.

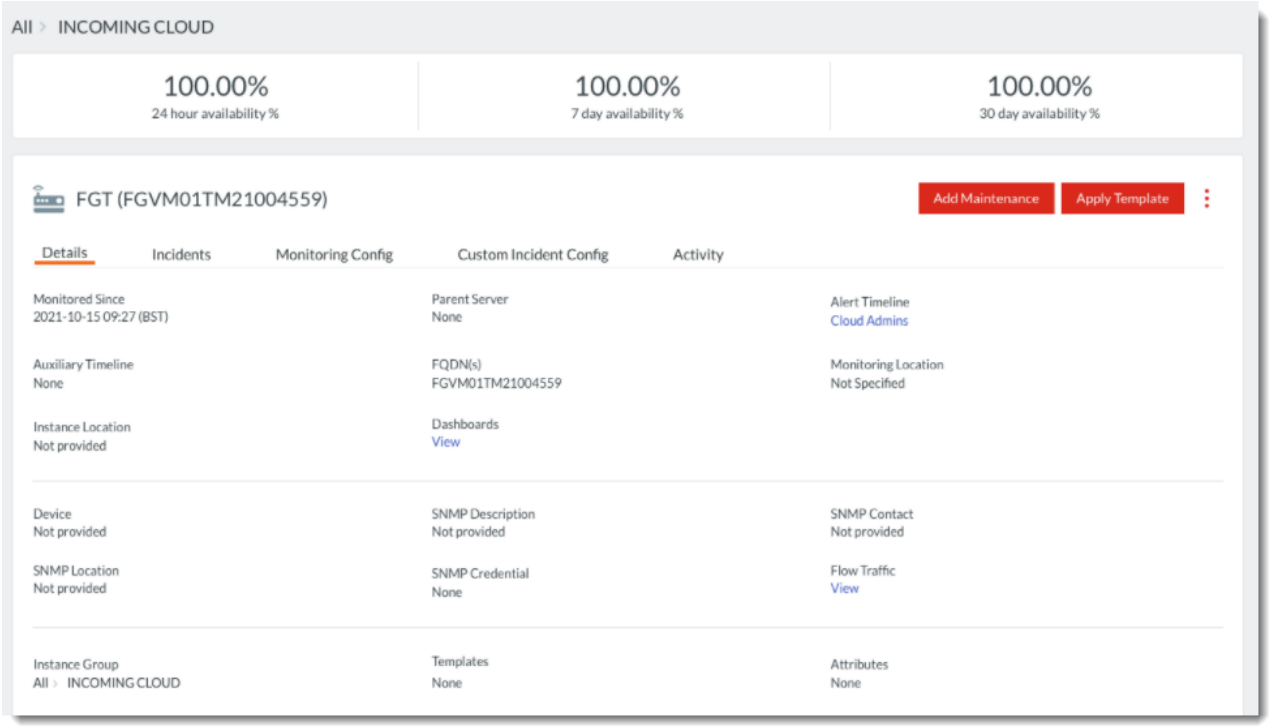
8. Click **Go to Fabric portal** to authorize the certificate. Note that if you selected an OnSight to monitor the FortiGates, the **OnSight name** will be displayed instead of **FORTIMONITOR**.
9. Click **Continue to Discover & Select**.
10. You can now begin the process of device selection and import.
11. Once device selection is complete, you will be required to enter an API key for each of the selected FortiGates.
12. Next, configure the **Instance Group**, which is the logical organization of the monitored instances within FortiMonitor.



13. A summary view of the configuration will be displayed before committing the changes.
14. After selecting **Finish** to add the devices, you have the option to be alerted upon completion.
Note: Depending on the number of devices in your Fabric environment, this process may take a few minutes. A banner will be displayed once the process is complete.



15. Once the process completes, the individual devices may be located on their respective instance pages.



16. You can manage the Fabric integration by going to **Settings** > **Fabric Settings**.

Fabric group

This article provides the steps on how to configure monitoring for Fabric 6.x devices using an OnSight vCollector.

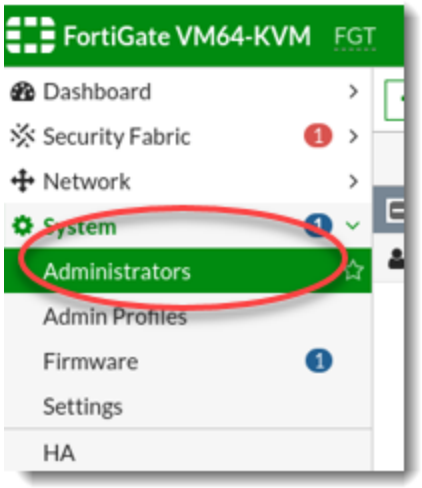
Prerequisites

- Web administration is enabled on all FortiGates
- Active Security Fabric (Devices should be present under Fabric Root). See Fortinet Security Fabric Administration Guide.
- OnSight is able to access the admin port of all FortiGates (port 443)
- The serial of the root FortiGate. To get the serial number, open the console on the root FortiGate then run **get system status | grep Serial**.
- API key and API user for all FortiGates that you want to monitor. Note that any attached devices (switches, APs, extenders) cannot be monitored without an API key from the FortiGate.

Create an API Key and API User

An API key and an API user must be created on each FortiGate.

1. Log in to the FortiGate GUI. See Using the GUI for more information.
2. Go to **System** > **Administrators**.



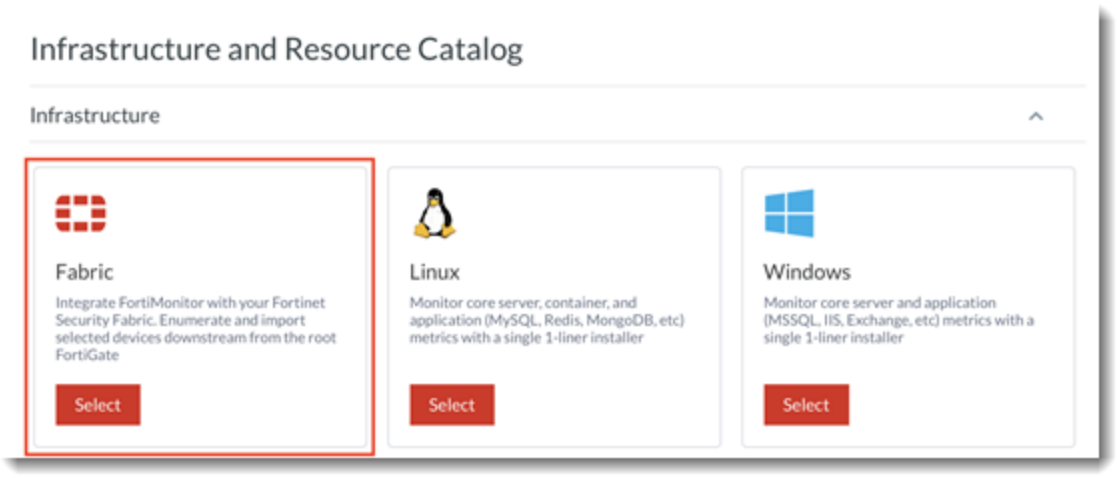
3. Click **Create New** > **REST API Admin**.
4. Enter the following information:
   a. Username - Enter **FortiMonitor**.
   b. Administrator Profile - Select **super_admin_readonly**.
5. Click **OK**. Take note of the generated API key. This will be needed to onboard the FortiGate and its devices to FortiMonitor.

FortiMonitor Configuration

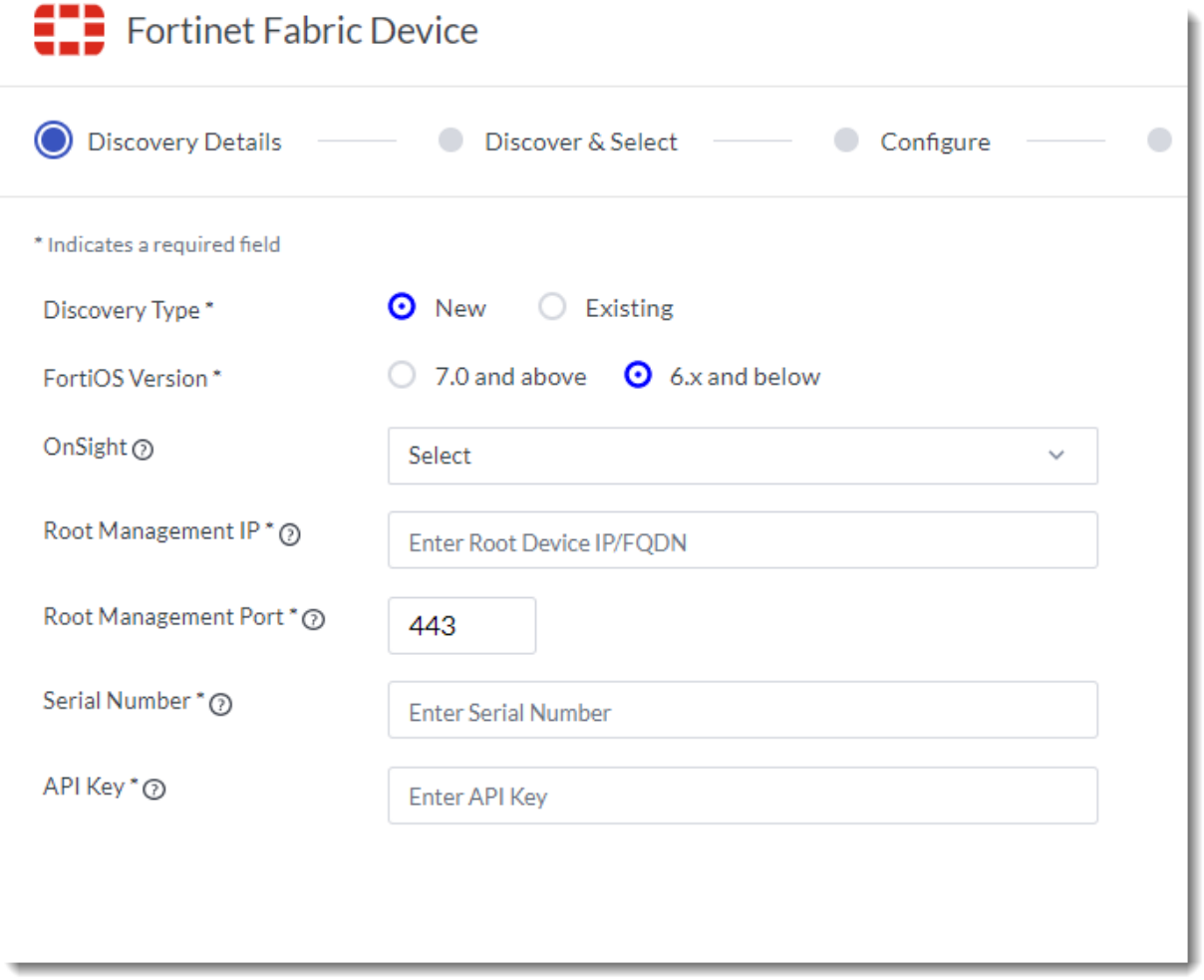
After creating an API key and API user on each FortiGate, you can now connect FortiMonitor to your Fabric environment.

1. Log in to FortiMonitor (https://fortimonitor.forticloud.com/).
2. From the navigation menu, click **Add**. The **Infrastructure and Resource Catalog** is displayed.

3. Select **Fabric** from the **Infrastructure** section of the catalog.



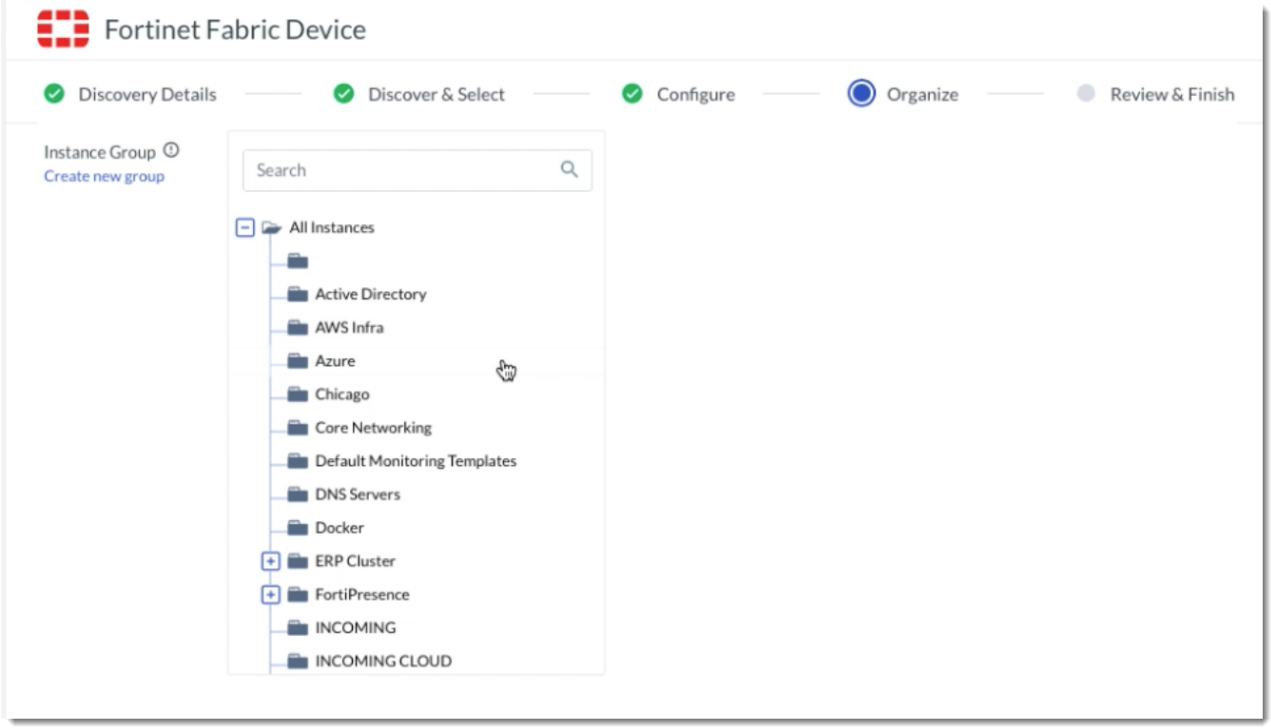
4. Select **Fabric Group**.
5.
   - From here, follow the on-screen prompts.



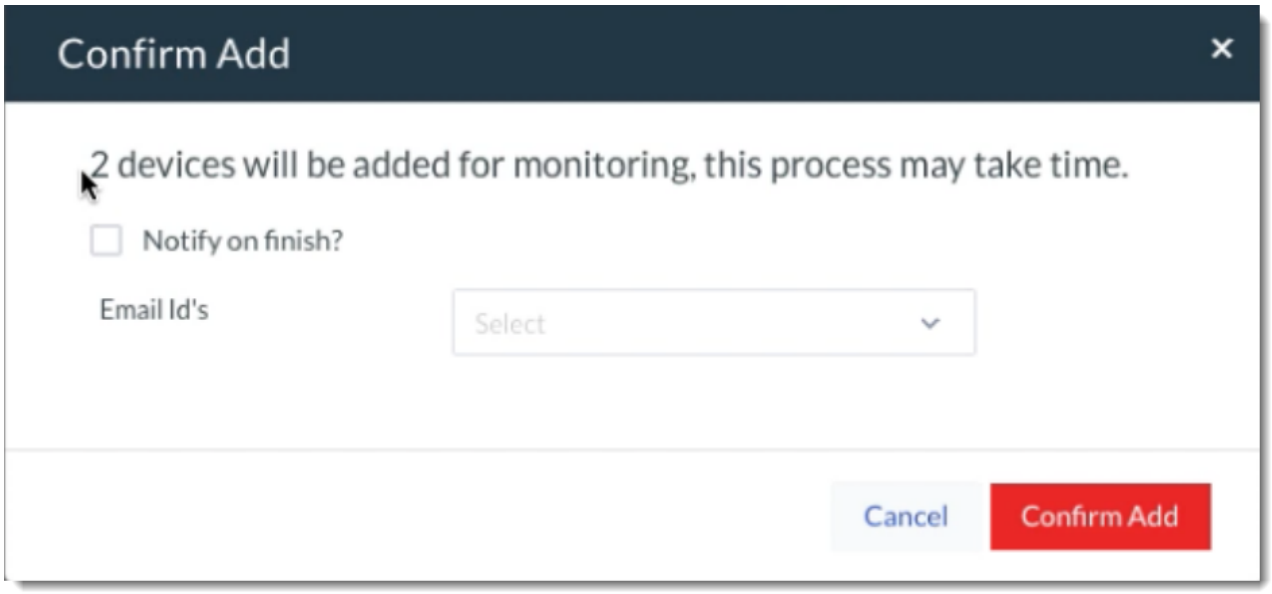
   - Discovery type - Select **New**.
   - FortiOS Version - Select **6.x and below**.

- OnSight - Select an OnSight. This OnSight will be used to monitor the FortiGates and associated devices. See OnSight vCollector for more information.
- Root Management IP - Enter the IP address where your root FortiGate can be reached from the selected OnSight.
- Root Management Port - Enter the management port of the root FortiGate.
  Note: Make sure to click **Specify** on both the **Management IP** and **Management Port** fields on the root FortiGate.
- Serial Number - Enter the serial number for the FortiGate Security appliance. To get the serial number, open the console on the root FortiGate then run:
  `get system status | grep Serial`.
- API Key - Enter the API key for the REST interface of the root FortiGate

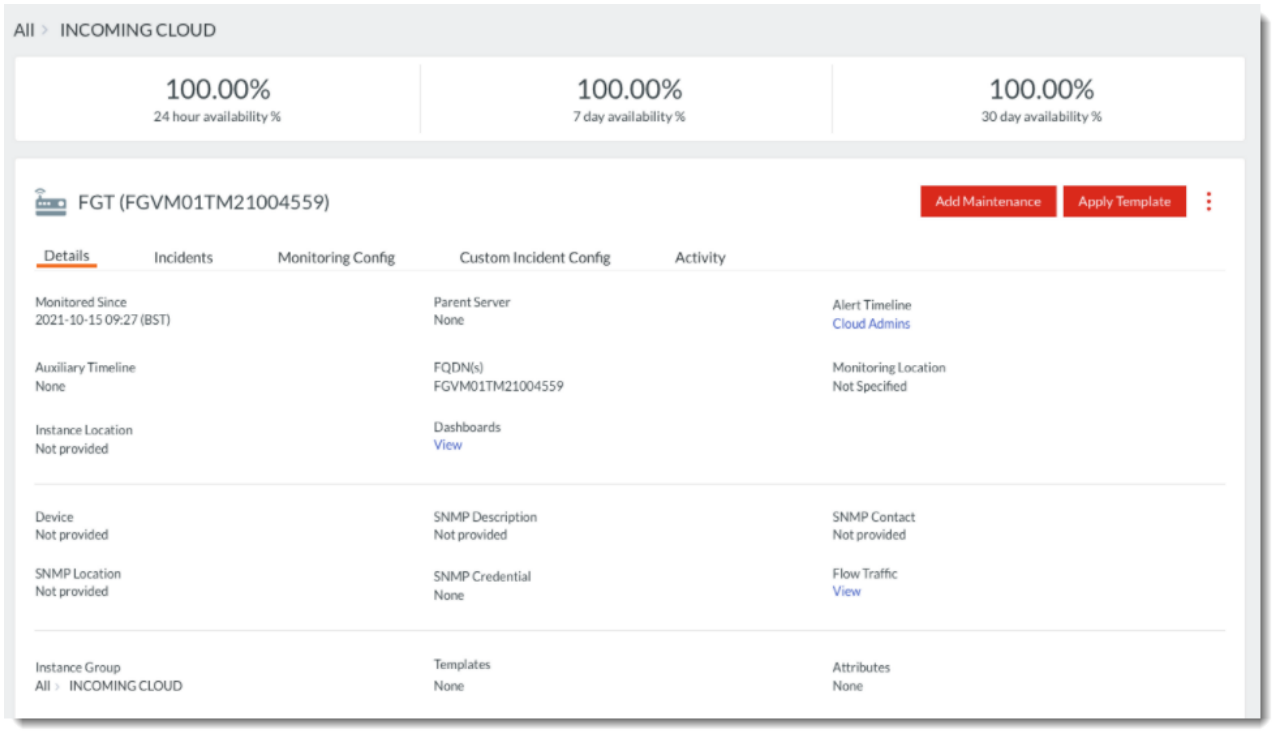
6. Click **Continue to Discover & Select**.
7. You can now begin the process of device selection and import.
8. Once device selection is complete, you will be required to enter an API key for each of the selected FortiGates.
9. Next, configure the **Instance Group**, which is the logical organization of the monitored instances within FortiMonitor.



10. A summary view of the configuration will be displayed before committing the changes.
11. After selecting **Finish** to add the devices, you have the option to be alerted upon completion.
    Note: Depending on the number of devices in your Fabric environment, this process may take a few minutes. A banner will be displayed once the process is complete.



12. Once the process completes, the individual devices may be located on their respective instance pages.



13. You can manage the Fabric integration by going to **Settings** > **Fabric Settings**.

Fabric integration via FortiManager proxy

This article provides the steps on how to configure a Fortinet Security Fabric environment for monitoring with FortiMonitor using the FortiManager as a proxy. See Security Fabric for an overview of the solution.

Note: This solution only works with the FortiManager virtual appliance. FortiManager Cloud is not supported.

Supported Fortinet devices

The following Fortinet devices are supported:

- FortiGate (plus SD-WAN)
- FortiSwitch
- FortiAP
- FortiExtender

Prerequisites

- FortiGate Firewall, running FortiOS 6.0.0 or later
- FortiGate must be authorized in FortiManager
- FortiManager is installed
- A FortiManager administrator account setup to allow FortiMonitor's access. See Create an administrator account.
- Web Service access must be enabled in FortiManager. See Enable Web Service access.
- OnSight is installed into a network that can reach FortiManager

FortiManager configuration

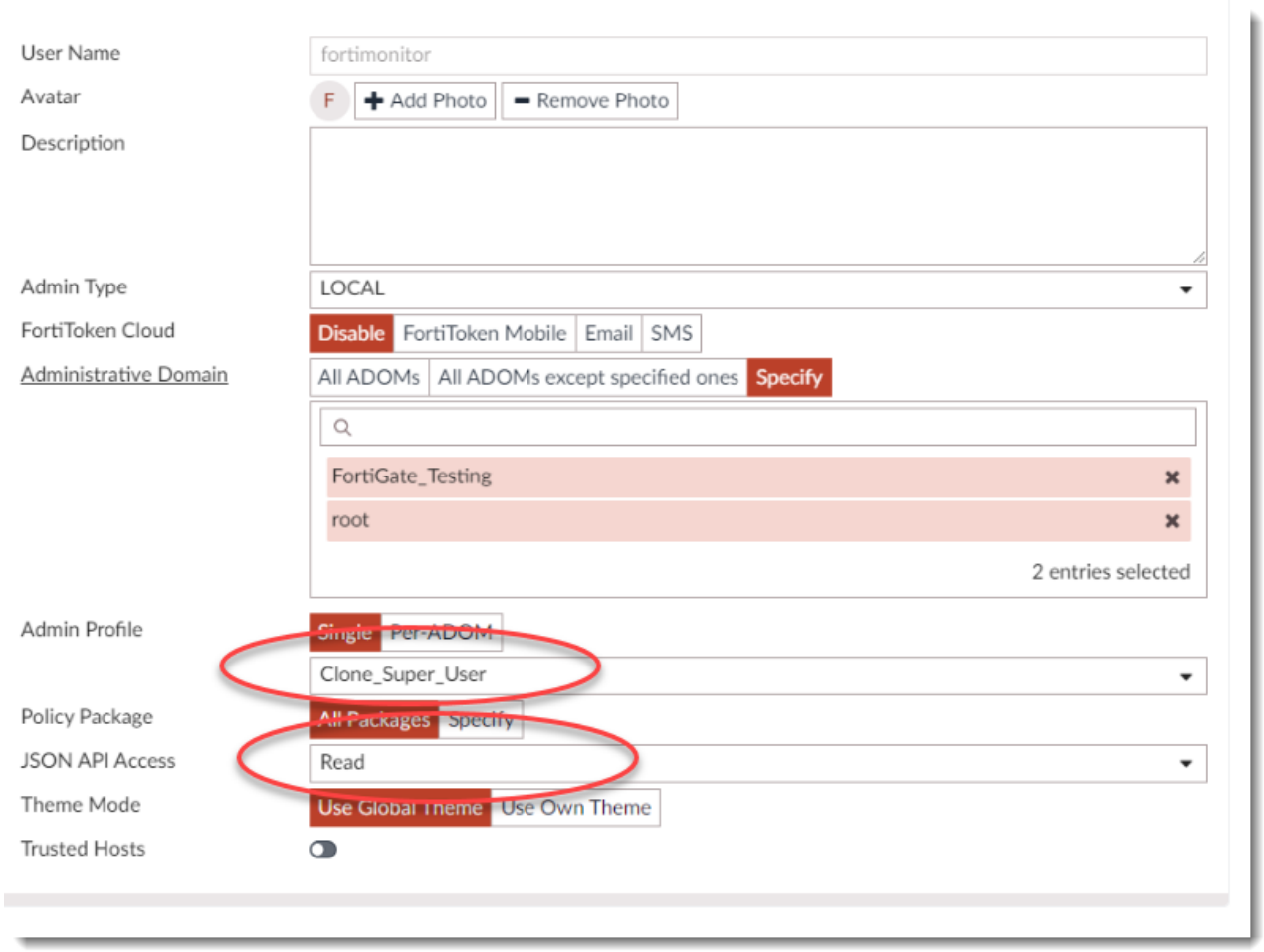
Create an administrator account

An administrator account must be created to allow FortiMonitor to access FortiManager. The steps and field description are detailed in Creating administrators.

When filling out the form, make sure to select the correct **Admin Profile** and **JSON API Access** settings.

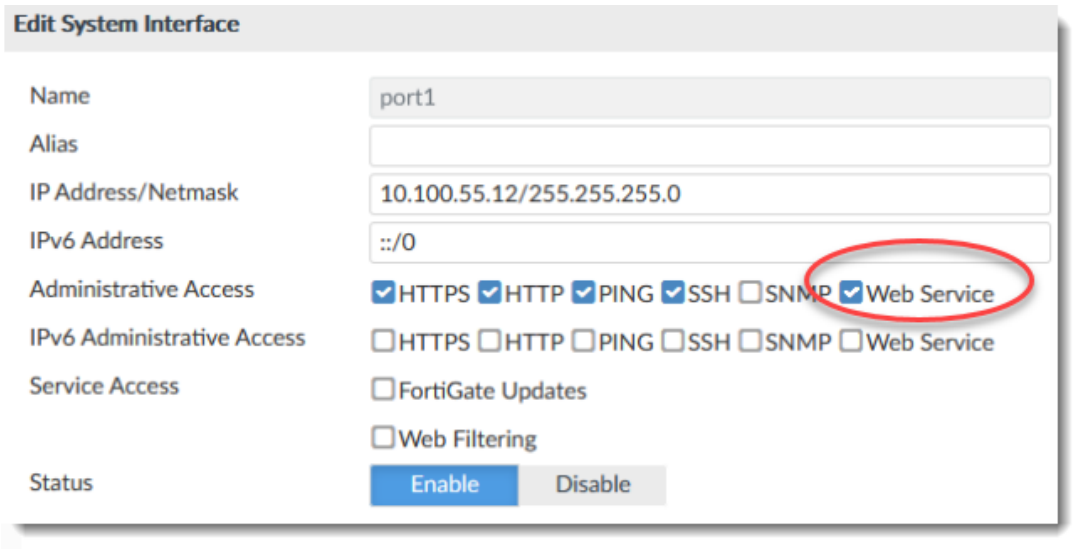
- Clone the **Super_User** profile.

- Admin Profile - Select the **Super_User** profile that you just cloned.
- JSON API Access - **Select Read**.



Enable Web Service access

Web Service access must be enabled for the FortiManager management port.

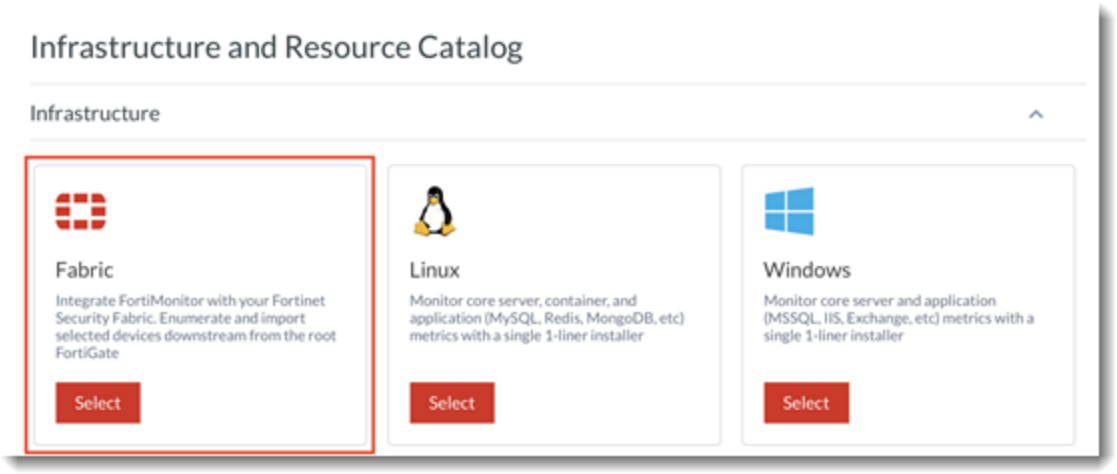


The steps on how to enable the setting can be found in Configuring network interfaces.

FortiMonitor configuration

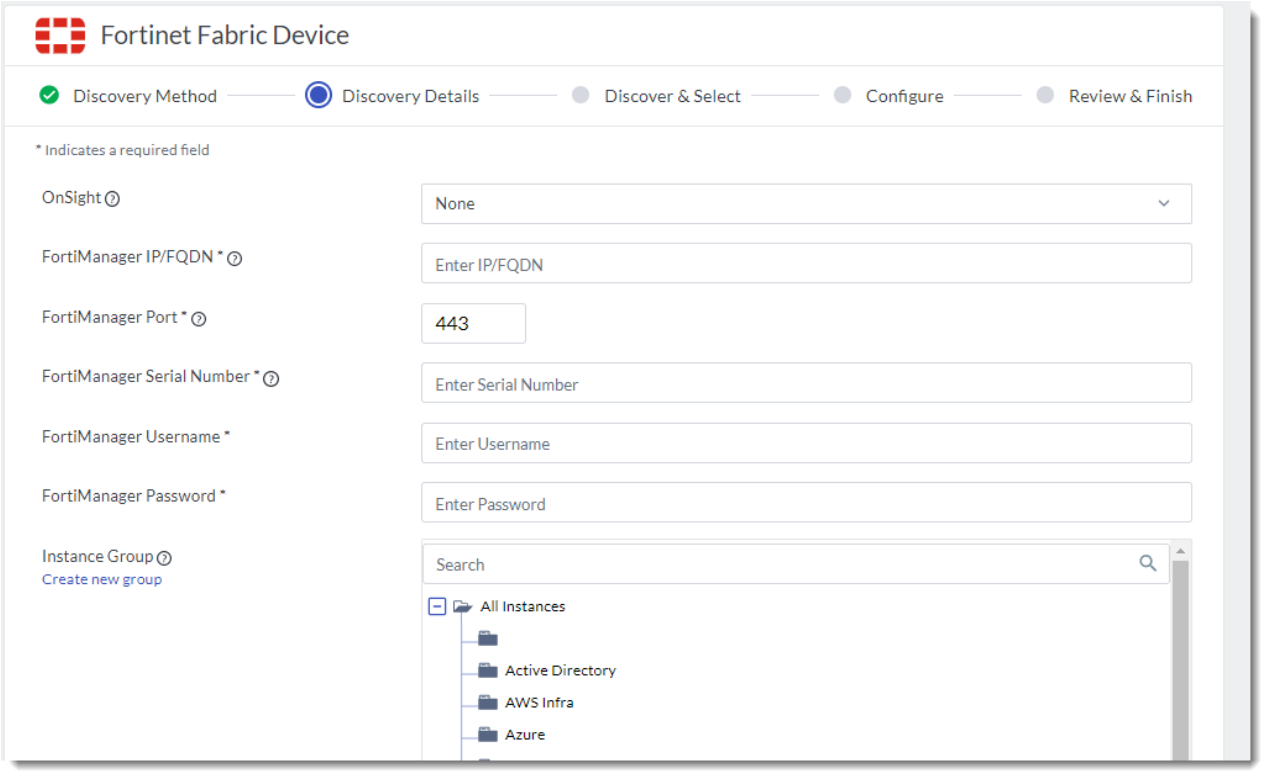
After completing the prerequisites and FortiManager configuration, you can now connect FortiMonitor to your FortiManager for Fabric monitoring.

1. Log in to FortiMonitor (https://fortimonitor.forticloud.com/).
2. From the navigation menu, click **Add**. The **Infrastructure and Resource Catalog** is displayed.
3. Select **Fabric** from the **Infrastructure** section of the catalog.

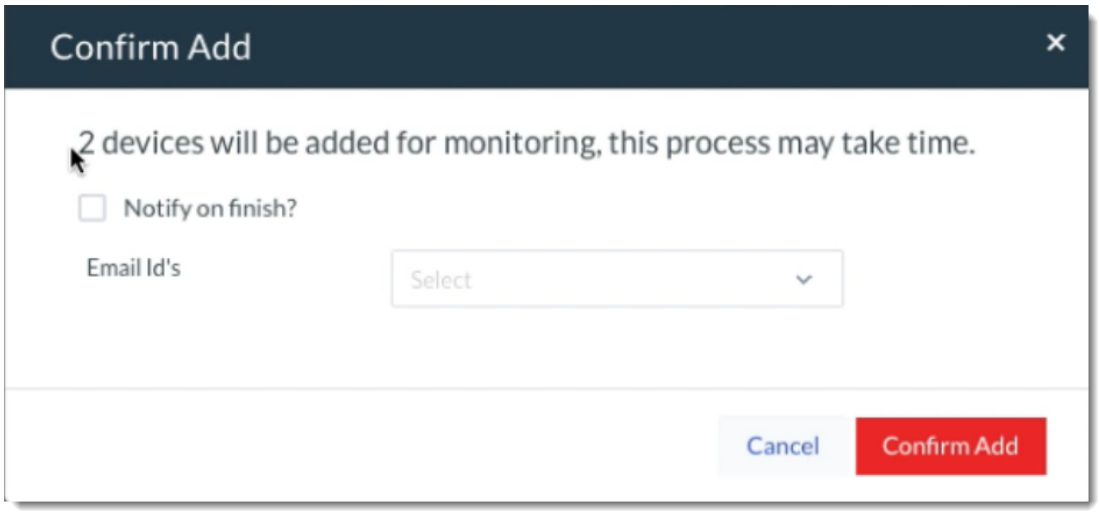


4. Select **FortiManager Proxy**.

5. From here, follow the on-screen prompts.



   - OnSight - Select an OnSight. The selected OnSight will perform the monitoring of the environment.
   - FortiManager IP/FQDN - Enter the **IP address** where FortiManager can be reached from the OnSight.
   - FortiManager Port - Enter the FortiManager **HTTPS port**. This is usually **443**.
   - FortiManager Serial Number - Enter the **serial number**. The FortiManager serial number can be found in the **System Information** dashboard in FortiManager.
   - FortiManager Username - Enter the **FortiManager username** of the administrator account you created.
   - FortiManager Password - Enter the **FortiManager password** of the administrator account you created.
   - Instance Group - Select or create an **Instance Group**. This allows you to define the instance hierarchy and organize your instances. By default, all devices discovered through this wizard are placed in the group Staging Environment.
6. Click **Continue to Discover & Select**.
7. You can now begin the process of device selection and import. Select the devices to import.
8. Once device selection is complete, you may now assign tags, monitoring templates, and alert timelines.
9. A summary view of the configuration will be displayed before you can add the devices.
10. After selecting **Finish** to add the devices, you have the option to be alerted upon completion.
    Note: Depending on the number of devices in your Fabric environment, this process may take a few minutes. A banner will be displayed once the process is complete.

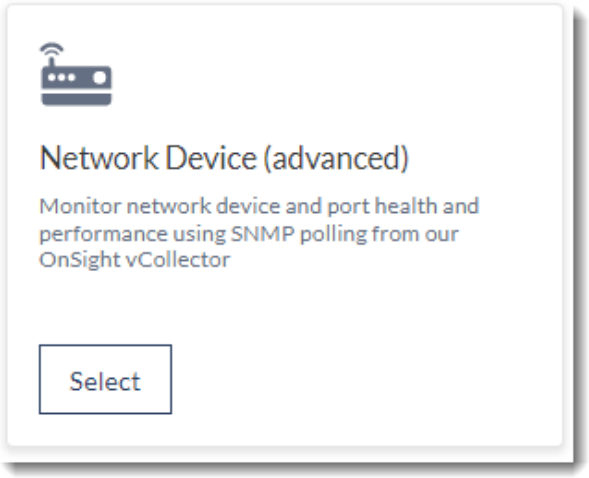


11. Once the process completes, the individual devices may be located on their respective instance pages.
12. You can manage the Fabric integration by going to **Settings** > **Fabric Settings**.

Network device monitoring

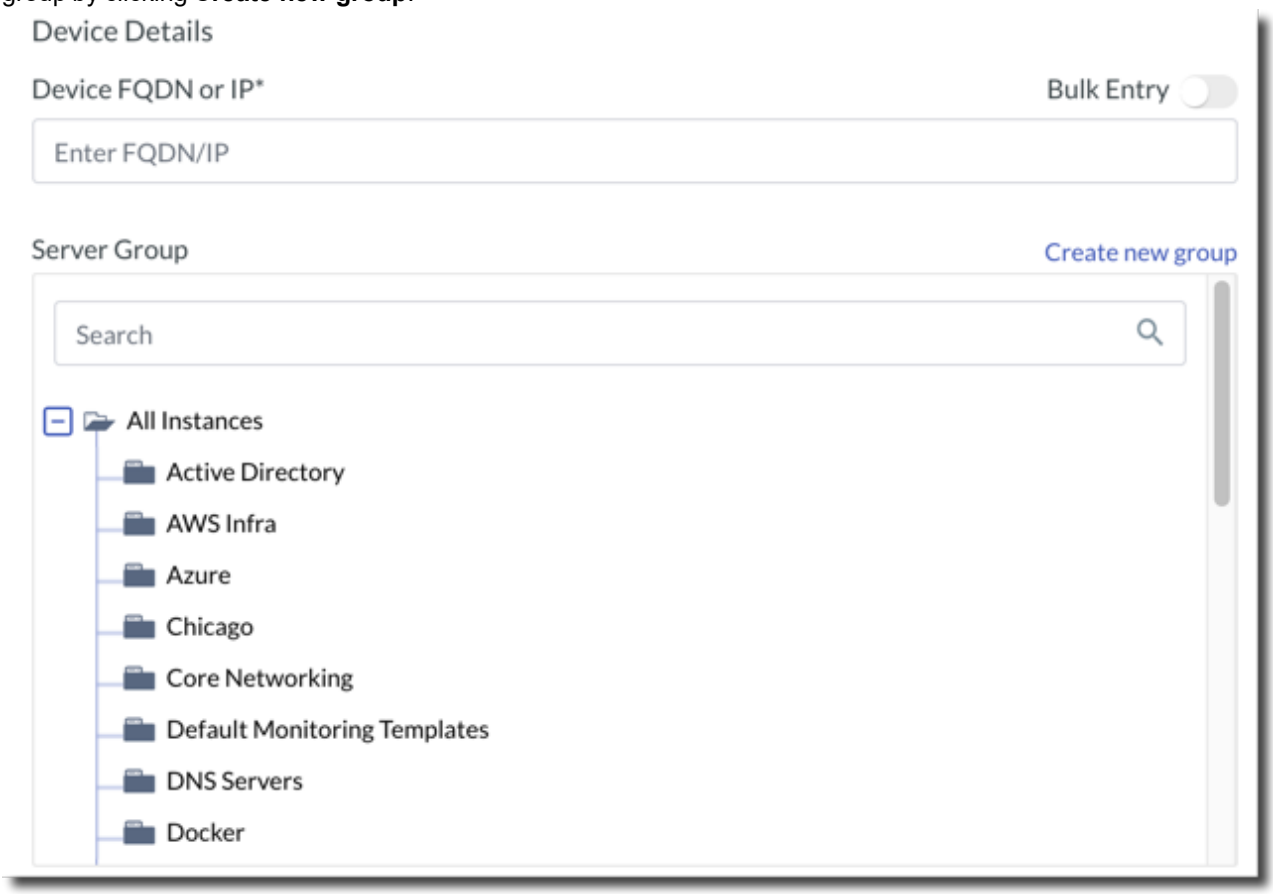
Add a network device manually

Select **+ Add** from the navigation menu then select **Network Device (Advanced)**.



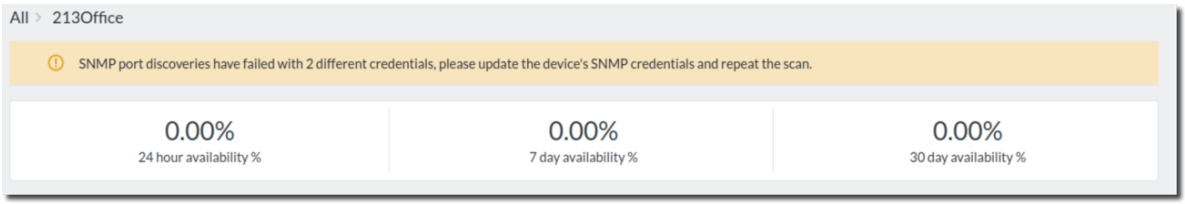
1.
2. Select which OnSight you would like to monitor the device from then click **Next Step**.

Add the FQDN or IP address of the device, and select which group to put the device into. You can also create a new group by clicking **Create new group**.



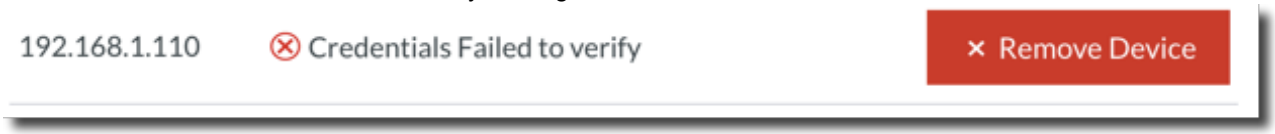
3.

4. (Optional) Enable the **Bulk Entry** option if you want to add more than one.
   a. If no valid SNMP credentials match while adding network devices by bulk, a banner showing the status will be displayed. This banner will also show if the device has an assigned credential and if the last scan that was ran failed.



5. (Optional) Enable the **Process through Monitoring Policy Workflow** option to send the network devices to the Monitoring Policy Workflow for processing.
6. (Optional) Add attributes
7. Select an existing SNMP credential for the device.

   Once the credentials have been verified, you can proceed with configuring the device (templates, tags).
   If the credentials did not match, you will have the option to choose different credentials. For bulk entries, network devices that have failed can be removed by clicking **Remove Device**.



8.

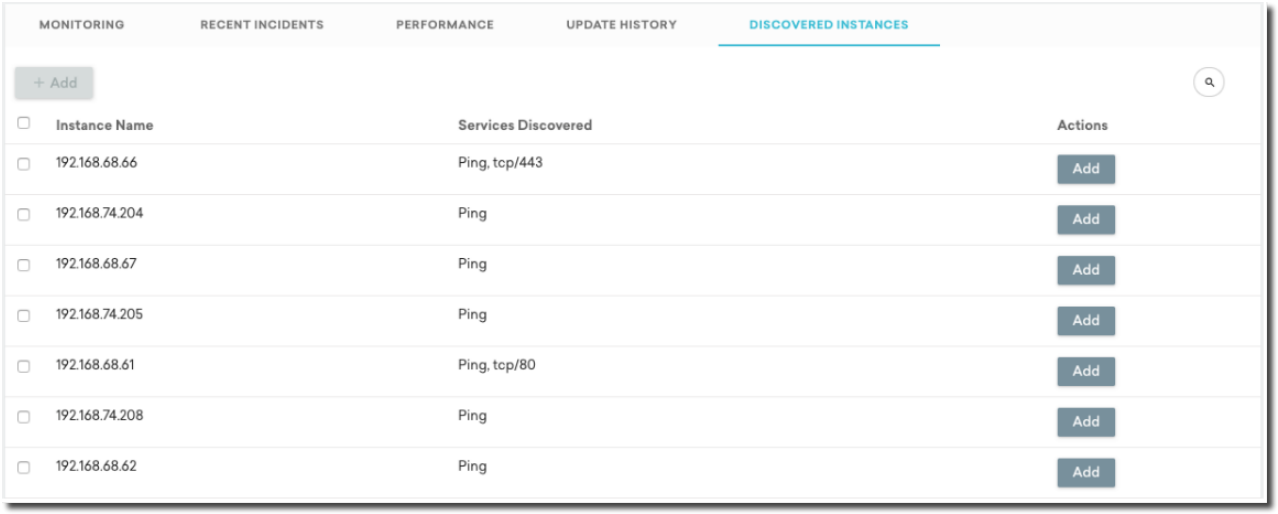
9. Select **Add Device**, the new device will populate under the group you selected.

After adding the network device, the OnSight automatically detects its type and then provisions the standard metrics applicable to the device.

Add network devices with OnSight Discovery

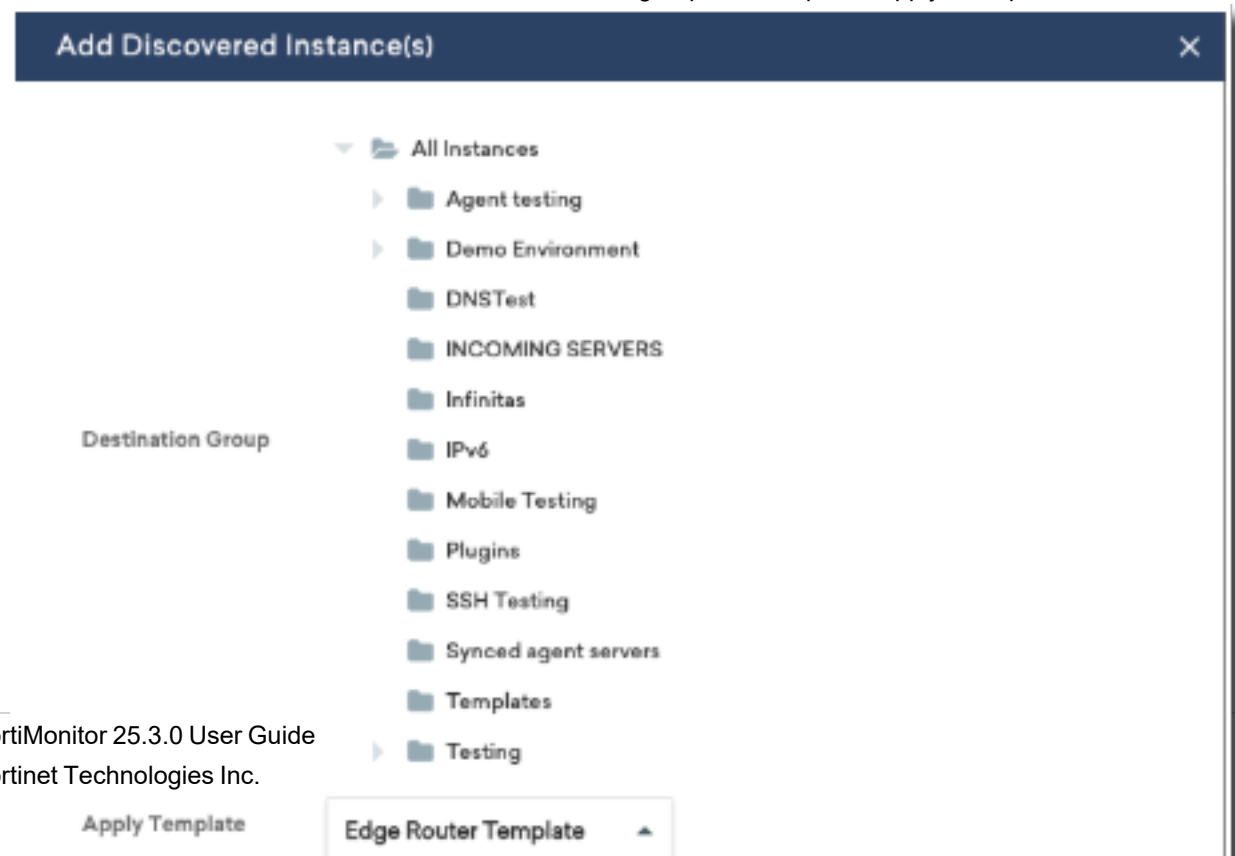
To add network devices that your OnSight vCollector has discovered:

Go to the OnSight page and select **Discovered Instances**. If you do not know how to set up discovery in your account, see OnSight Discovery.

1. 

You will see the IP address or the instance name of the device that had been discovered and the services which were discovered on the device. The checkbox allows you to select multiple instances.

2. Once instances are selected, choose the destination group and if required, apply a template.



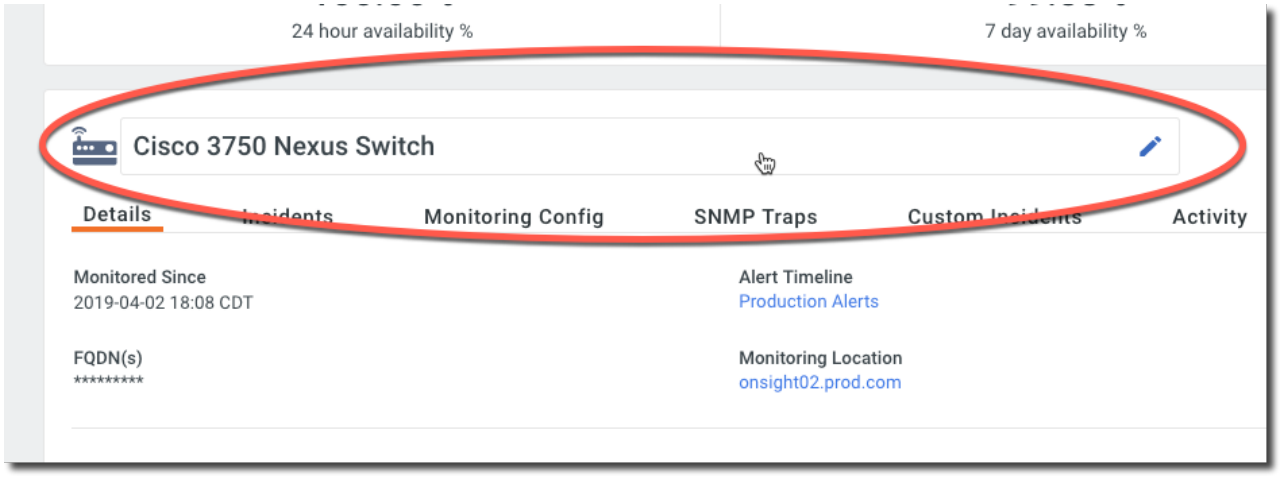
After adding the network device, the OnSight automatically detects its type and then provisions the standard metrics applicable to the device.

You can also monitor SNMP with SNMP traps.

Configure the network device

To configure a network device, perform the following steps:

1. Go to the **Instance Details** page of the network device that you want to configure.
   Select the name of the network device or the **Edit** icon.



2. The **OnSight Configuration** drawer will slide out.
3. Edit the following settings as needed:

Option	Description
General	
Name	Change the name of the network device.
Server Key	The unique key used to reference this device.
Instance Group	Select an instance group for the network device.
Tags	Enter tags that can help define the network device. For example network, cisco. Press Enter after each tag. Known tags will be auto-suggested to you. Tags should not contain spaces.
Attributes	Add key-value pairs that can provide more structure to your instance metadata.
FQDNs and IPs	Add additional interface names. See Additional IP addresses or FQDNs.
Description	Enter a description for the template.
Parent Instance	(Optional) The device upstream from this server that it depends on. If the parent device experiences an outage, this server's outage notifications will be suppressed.

Prometheus Endpoints	Add a Prometheus endpoint to the network device.
SNMP Credentials	Add an SNMP credential to the template.
Select Credentials	The SNMP credential used for this device.
Recurring SNMP Discovery	Enable this option if you want the OnSight to regularly scan the network device. Detected metrics will automatically be provisioned.
SNMP Scan Frequency	This option only appears when the **Recurring SNMP Discovery** option is enabled. Specify how often (minutes) the network device is scanned for new metrics.
Port Metric Provisioning	Enable this option to auto-provision SNMP metrics.

1. Select

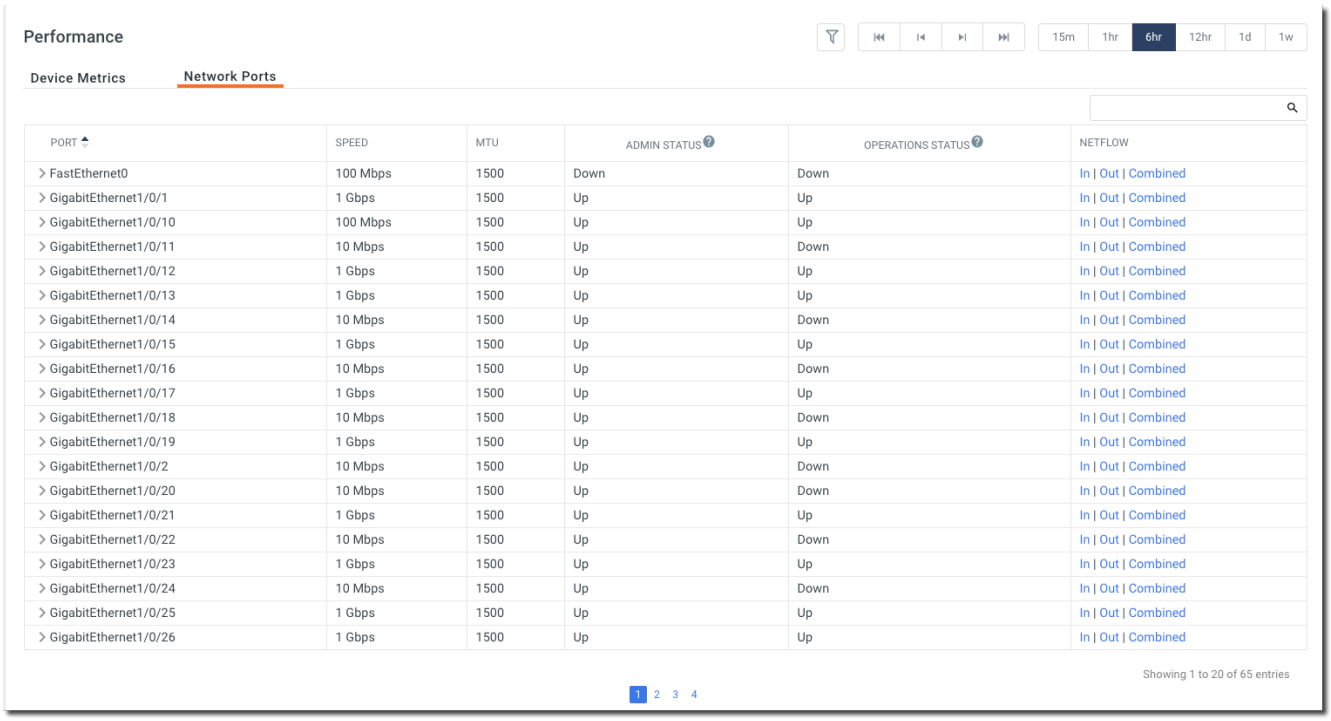
Save Changes.

Monitor metric performance

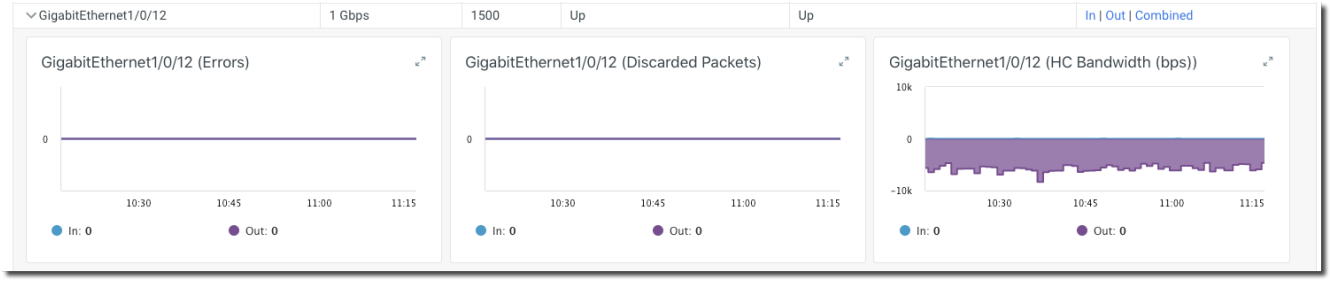
In the **Performance** section of the network device's **Instance Details** page, each metric is presented differently. Device-level metrics use performance line charts along the top.



Port/interface-level metrics use a customized roll up display which shows interface stats in expandable rows.



Click on a specific row to expose all interface-specific performance and health metrics. The graphs will show both inbound/outbound stats stacked on the same graph with zooming ability. Click on the expand icon in the upper right-hand corner of a graph to see a fullscreen display.



Using templates allows you to standardize your thresholds set and alerts and implement them across similar devices.

NetFlow monitoring

This article describes the steps on how to enable NetFlow monitoring in FortiMonitor.

Note: Your firewall rules must be updated to allow the following NetFlow ingest URLs depending on your region:

- US region: nf.us01.fortimonitor.com:443
- EU region: nf.eu01.fortimonitor.com:443

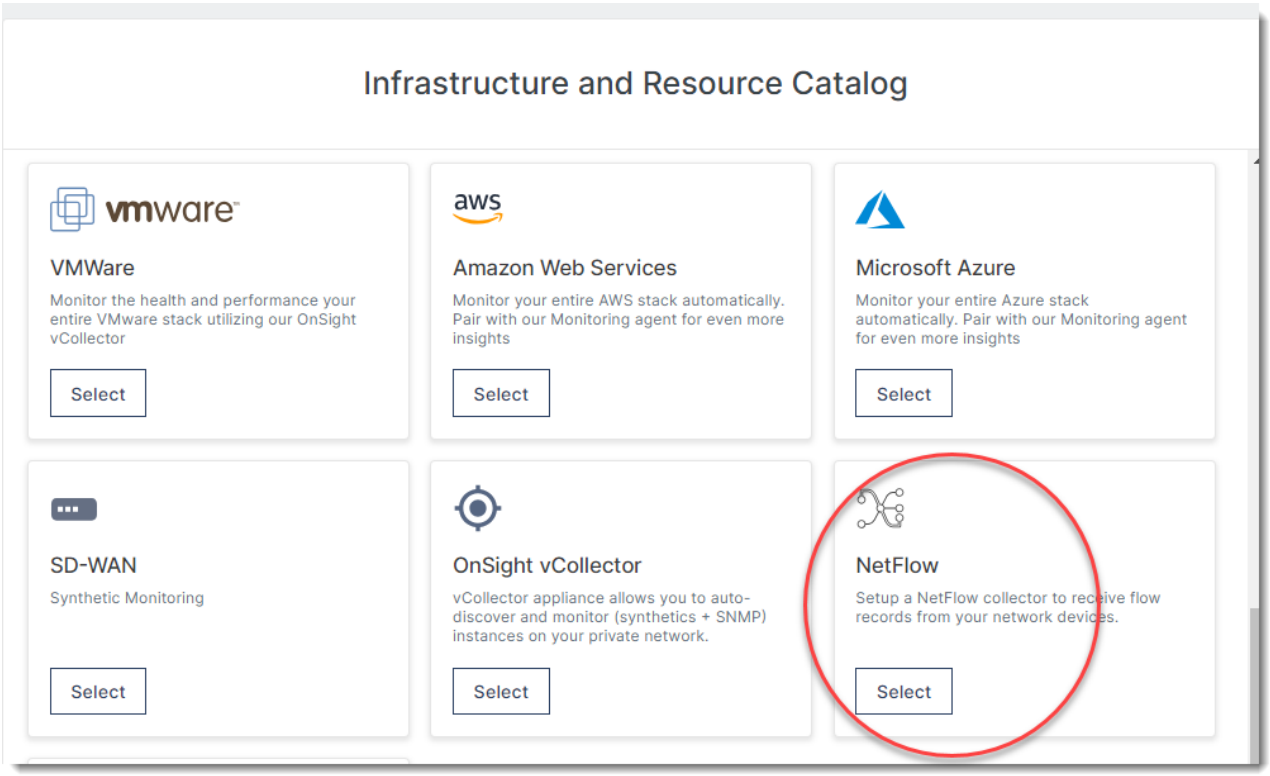
Supported NetFlow versions

- NetFlow v5
- NetFlow v9
- IPFIX (NetFlow v10)
- sFlow

Enable NetFlow monitoring

1. To enable NetFlow monitoring, perform the following steps:
2. From the navigation menu, click **Add**.

3. Select **NetFlow**.



Installation modes

Standalone installer

You can install the NetFlow collector using a VM that will be turned into the NetFlow collector appliance.

Make sure that the VM you are using meets the following requirements before you start enabling NetFlow monitoring:

- VM requirements
  - A Linux VM that will be turned into the NetFlow collector appliance
  - 1 core processor
  - 2 GB RAM
  - 20 GB disk
- OS requirements
  - Recent Ubuntu, RedHat, or Debian distribution.
  - Podman must not be installed.
- NetFlow device must be configured to send flow data

Note: The installer checks to make sure that firewall rules are open, if not you will get a warning to open those to allow flow data in.

To enable firewall access to UDP port 2055 to receive flow packets, run the following command:

```
iptables -A INPUT -p udp -m udp --dport 2055 -j ACCEPT
```

Configure the VM that you will use as the NetFlow collector appliance.

1. SSH into your Linux VM.
2. Root privileges are need for the next steps. Run the following command:
   `$ sudo su -`
3. Enable firewall access to UDP port 2055 to receive flow packets by running:
   `iptables -A INPUT -p udp -m udp --dport 2055 -j ACCEPT`
4. Download the installer by running the following command:
   `curl -fsSL https://repo.fortimonitor.com/install/netflow/install-fortimonitor-netflow.sh > install-fortimonitor-netflow.sh`
5. (**Optional**) Run the installer in**Estimator** mode. This installs the NetFlow estimator and will not register the appliance to FortiMonitor.
6. Run the installer as root. You can choose any name for the NetFlow appliance:
   `bash install-fortimonitor-netflow.sh -customer_key <customer_key> -appliance_name "NetFlow Appliance"`
7. Configure the NetFlow source device.

Virtual appliance

Make sure that the following requirements are met before enabling NetFlow monitoring:

- At least 2 GB of memory (4 GB is recommended)
- At least 30 GB of available disk space
- Minimum network connectivity

Perform the following steps to download and install the NetFlow collector image:

1. Download the NetFlow vCollector image for your hypervisor.
2. After downloading the NetFlow collector image, import it as a virtual machine into your hypervisor. Once your NetFlow vCollector is imported and booted, the VM will go through the normal Linux startup process, finishing with a login prompt.
3. Log in with username **fortimonitor** and password **fortimonitor**.
4. You will then be prompted to set a new password. Important: Do not lose this password. Without it, there is no way to access the NetFlow vCollector for further updates.
5. Register the NetFlow vCollector by running the following command as root:
   `fortimonitor-netflow register <customer_key> "NetFlow Appliance"`
6. This will connect your NetFlow vCollector to the FortiMonitor cloud and begin syncing data.
7. Start it by running the following as root:
   `fortimonitor-netflow start`
8. To run the estimator, run the following command as root:
   `fortimonitor-netflow estimator <cidr-filters-file>`
9. Configure the NetFlow source device.

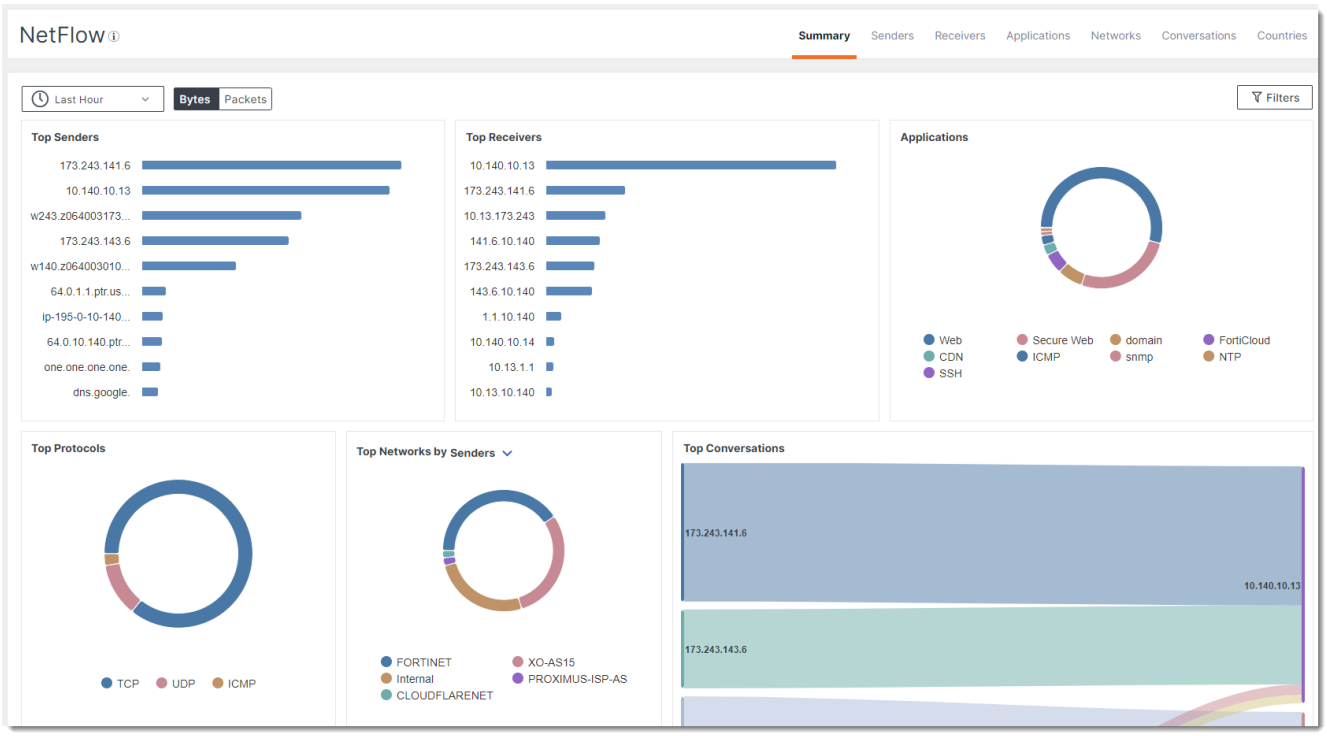
Configure the NetFlow source device

You must configure your NetFlow source device, in this case a FortiGate, to send flow data to FortiMonitor.

1. Log in to the NetFlow source device. For more information on how to configure NetFlow in FortiGate, see https://docs.fortinet.com/document/fortigate/7.4.0/cli-reference/31620/config-system-netflow.
2. Configure NetFlow in FortiGate.
```
# config system netflow
set active-flow-timeout 60
set collector-ip <netflow-collector-ip>
set collector-port 2055
end
```
Where`<netflow-collector-ip>` is the IP address of your NetFlow collector appliance.
3. Enable NetFlow on each interface that you want to monitor traffic for. For example, **wan1**:
```
# config system interface
edit wan1
set netflow-sampler both
end
```

View the NetFlow dashboard

Log into FortiMonitor then go to **Dashboards** > **NetFlow**.



FortiMonitor NetFlow commands

This section describes the commands that can be used with the NetFlow collector appliance.

```
/usr/bin/fortimonitor-netflow <command>
```

Command	Description
register <customer_key>	Register your device with FortiMonitor.
start	Start the container.
restart	Restart the container.
stop	Stop the container.
upgrade	Upgrade to the latest version of FortiMonitor NetFlow.
status	Show the status of the FortiMonitor NetFlow container.
remove	Remove FortiMonitor NetFlow from your device
netflow-uninstall	• Stops and removes the NetFlow container and image • Uninstall the FM agent • Remove Docker volumes • Remove the CLI bash script, logs and any other files added during the installation

Estimator mode

The NetFlow estimator allows you to get your estimated NetFlow volume.

This step is optional.

1. Run the installer in Estimator mode by running the following command:
   `bash install-fortimonitor-netflow.sh -re y`
   This installs the NetFlow estimator and will not register the appliance to FortiMonitor. See Installation parameters for more details.
2. Upload the log files which includes your usage data. This will be used to get your estimated NetFlow volume.
   `fortimonitor-netflow upload-logs <customer_key>`

Switch to Collector mode

If you used the Estimator mode, switch to Collector mode to begin collecting and monitoring flow data. To do this, perform the following steps:

1. Stop NetFlow monitoring by running the following command:
   `fortimonitor-netflow stop`
2. Register your NetFlow collector appliance with FortiMonitor by running the following command:
   `fortimonitor-netflow register -c <customer_key> -a <appliance_name>`
   Where:
   - **<customer-key>** is your own customer key. To obtain your customer key, visit the FortiMonitor control panel and click your **account name** then select **My FortiMonitor Account**.

- **<appliance_name>** is the name of the appliance to be registered. This name will be used to help identify the flows in the NetFlow dashboard in the FortiMonitor control panel.

3. To begin collection, run the following command:
   `fortimonitor-netflow start`

VMware

The VMware integration allows you to collect metrics from your VMware clusters (if vCenter is being utilized), hosts, and datastores. FortiMonitor can perform automatic discovery and monitoring of these components within your VMware deployment.

Make sure that you have the latest version of the OnSight vCollector. To check your OnSight's version, see Manage and configure your OnSight vCollector.

Supported versions

- VMware 7
- VMware 8

Control panel configuration

To start monitoring your VMware instances, perform the following steps:

VMware user account creation

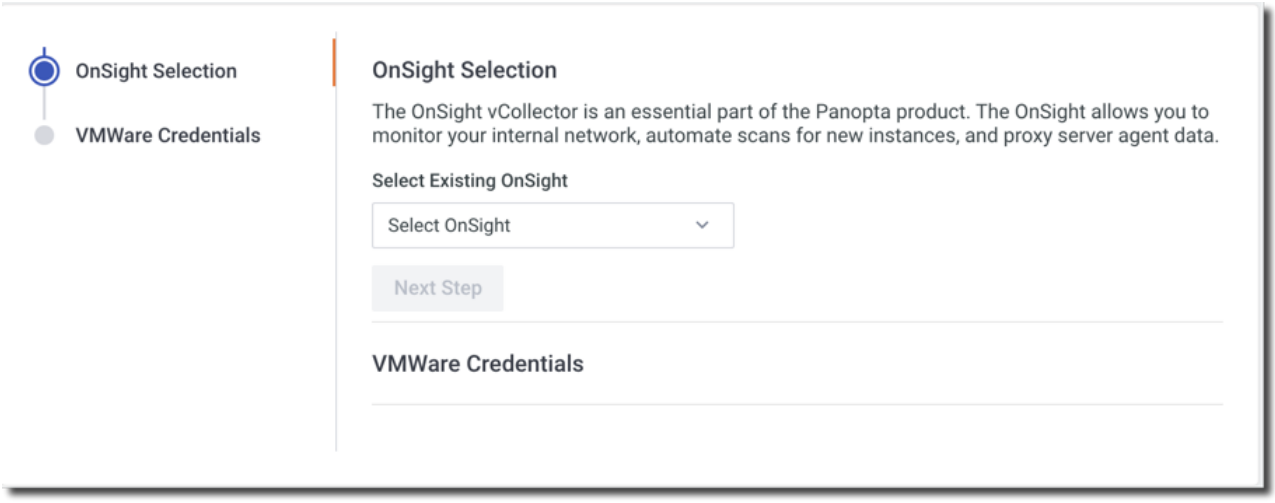
Before configuring the VMware integration in the control panel, create a read-only user for FortiMonitor in your VMware management tool, for example, vCenter Server Appliance. Use these credentials to set up your VMware integration.

1. From the navigation menu, click **Add**. The **Infrastructure and Resource Catalog** is displayed.
   Select **VMware** from the **Infrastructure** section of the catalog. If you do not see this option, please contact our support team to have it enabled.

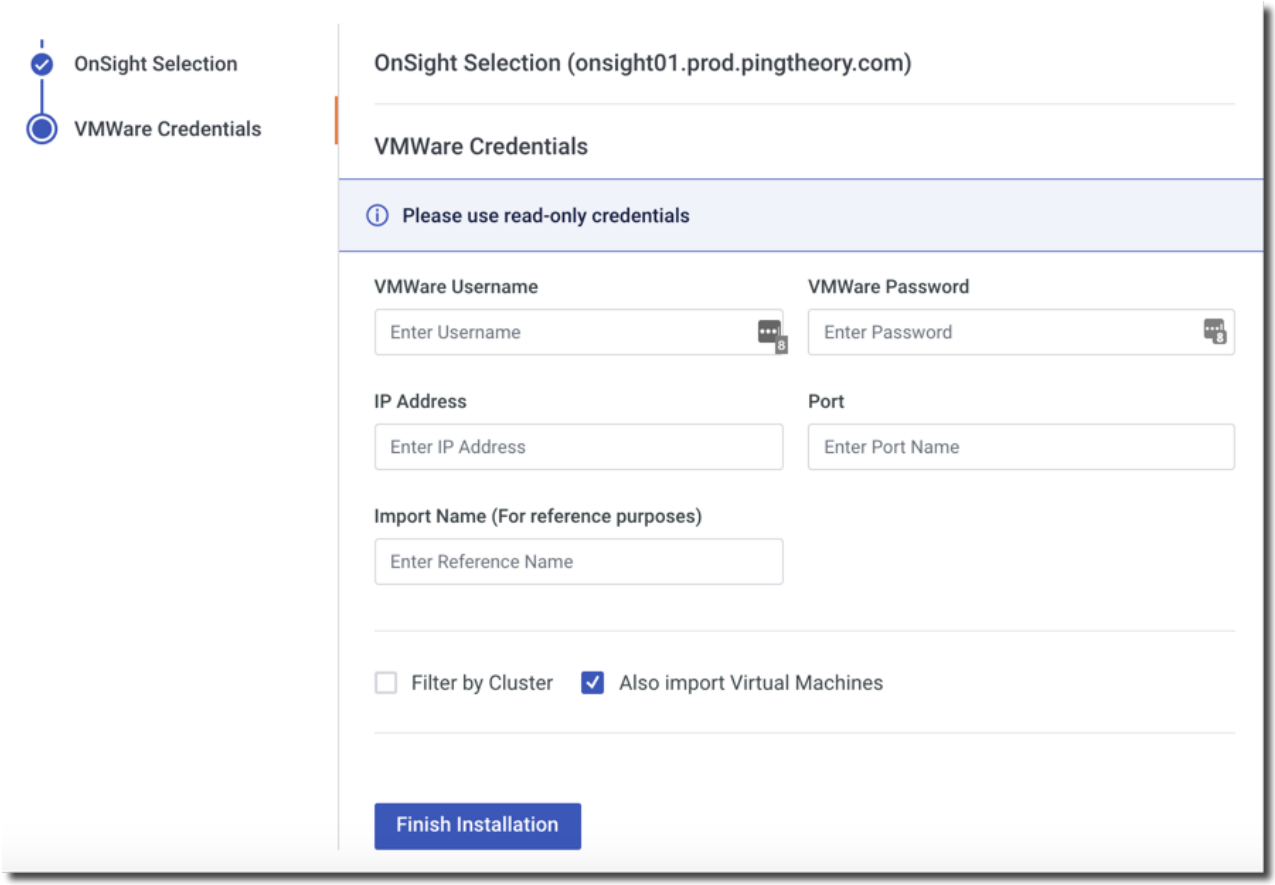


2.

Connect your VMware environment to FortiMonitor by entering your VMware connection information.



3.



The fields are described in the following table:

Field	Description
OnSight Collector	Select an OnSight vCollector instance to use as a monitoring location for this integration.

VMware Username	The username of your FortiMonitor VMware user account.
VMware Password	The password of your FortiMonitor VMware user account.
IP Address	The **IP address** or **host name** of your VMware vCenter.
Port	The port of your VMware Center (Usually Port 443).
Import Name	Enter the name to be associated with this integration.

4. Select **Finish Installation**.

After installation, and every 15 minutes from there on, all of your VMware instances will be automatically discovered and imported to FortiMonitor. All decommissioned instances will be removed from FortiMonitor as well. You can see all the discovered instances in the instance tree.

Instance Tree

If you open the instance tree on the left side of the control panel, you will see that your VMware resources are nested under a grouped named after your VMware vCenter instance.

Infrastructure Map and Dashboards

VMware instances will appear in the I**nfrastructure Maps** and **Dashboards** similar to any other monitored instance.

Supported metrics

Host metrics

The following table describes all of the supported host metrics for VMware.

Metric	Unit	Description
CPU Usage	Percentage	The total CPU resources utilized by the host in percent.
Wait	milliseconds	The CPU wait time in milliseconds.
CPU Usage: Total	Percentage	The aggregate value of the CPU resources used by all of the cores in the host.
Average read requests per second	Coun	The average number of read requests per second.
Average write requests per second	Count	The average number of write requests per second.
Disk Usage	KBps	The average of the data read and written for all of the disks in the host.

Physical device read latency	milliseconds	The physical device read latency in milliseconds.
Physical device write latency	milliseconds	The physical device write latency in milliseconds.
Read rate	KBps	The read rate in kilobytes per second.
Write rate	KBps	The write rate in kilobytes per second.
Active	KB	The amount of memory actively used.
Heap free	KB	The amount of free space in the heap.
Host consumed %	Percentage	The average memory consumed by the host in kilobytes.
Data receive rate: Total	KBps	The aggregate value of the data received rate in kilobytes per second.
Data receive rate	KBps	Data received rate in kilobytes per second.
Data transmit rate: Total	KBps	The aggregate value of the data transmission rate in kilobytes per second.
Data transmit rate	KBps	Data transmission rate in kilobytes per second.

Cluster metrics

The following table describes all of the supported cluster metrics for VMware.

Metric	Unit	Description
CPU Usage: Total	Percentage	The total CPU resources utilized by the cluster in percent.
Host Consumed	Percentage	Amount of host memory consumed by the virtual machine for guest memory.
VM guest reboot count	Count	The total number of times a VM guest has rebooted in a specified period.
VM guest shutdown count	Count	The total number of times a VM guest has shut down in a specified period.
VM guest standby count	Count	The total number of times a VM guest was on standby in a specified period.
VM power off count	Count	The total number of times a VM has powered off in a specified period.

VM power on count	Count	The total number of times a VM has powered on in a specified period.
VM reset count	Count	The total number of times a VM has reset in a specified period.
VM suspend count	Count	The total number of times a VM was suspended in a specified period.

Datastore metrics

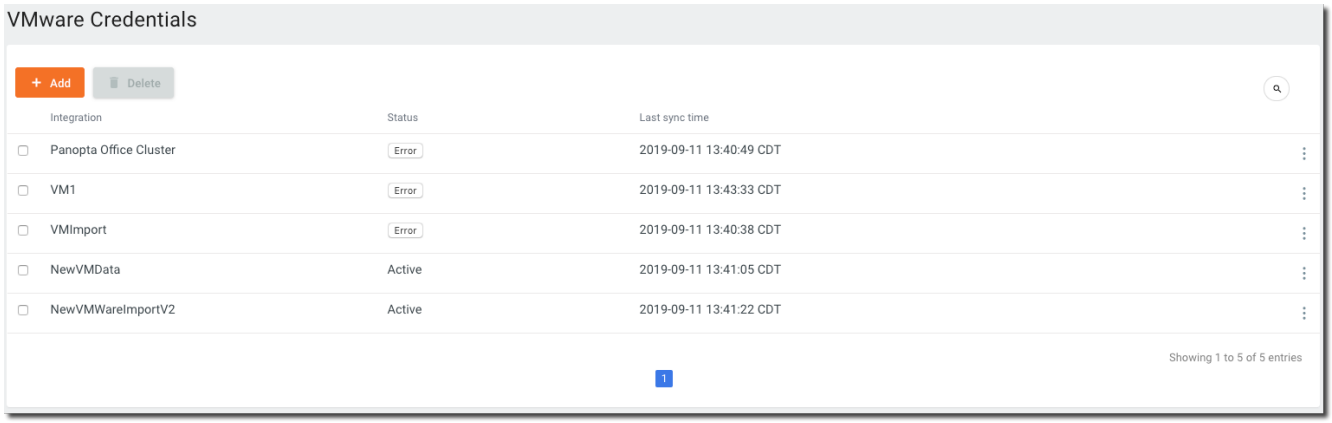
The following table describes all of the supported datastore metrics for VMware.

Metric	Unit	Description
Average read requests per second	Count	The average number of read commands issued per second to the datastore during the collection interval.
Average write requests per second	Count	Average number of write commands issued per second to the datastore during the collection interval.
Space actually used	GB	Total disk space used on all datastores visible to this object.

VMware Credentials page

The **VMware Credentials** page allows you to add, edit, or delete VMware credentials in FortiMonitor.

To access the **VMware Credentials** page, select **Monitoring** > **Infra Settings** > **VMware Settings** from the navigation menu.



Network checks

- Network service checks
- Synthetic monitoring
- Configure incident alert and response time thresholds
- Limit the number of confirmations for an outage
- Dynamic variables

Network service checks

FortiMonitor enables you to monitor servers, web endpoints, and other network-based services from behind your firewall with an OnSight vCollector or FortiMonitor Public Monitoring Probes. These checks provide deeper insight into your applications beyond just basic uptime.

DNS

DNS monitoring is a truly valuable and often neglected tool. If the authoritative DNS servers that are responsible for your server are down (or do not have the correct information) it makes your server effectively invisible to the internet. This type of monitoring checks each authoritative DNS server to see if it is running or if it has the correct IP information for your domain name.

The first thing you need to set up a DNS check is the IP or FQDN of the authoritative DNS servers you would like to monitor. To find the IP addresses of the authoritative DNS servers you can use the whois command (or www.whois.net) in a Linux terminal on the domain name of the server you would like to check. For example typing in whois panopta.com returns (among other things) the list of name servers we are looking for seen below:

```
Name Server: NS1198.DNS.DYN.COM
Name Server: NS2195.DNS.DYN.COM
Name Server: NS3181.DNS.DYN.COM
Name Server: NS4163.DNS.DYN.COM
```

Once you have the IP addresses or FQDN of your name servers add them to your list of servers in the FortiMonitor control panel.

Control Panel Configuration

Select DNS from the monitoring catalog.



There are two options for DNS checks.

Network check

- DNS Name Lookup
- DNS Port

A **DNS port check** is very rudimentary; it will only check to see if the authoritative DNS server is listening through a given port.

A **DNS name lookup** will allow you to test to see if this authoritative DNS server has an IP address for a specific domain name. For additional protection you can add the correct IP address for this domain name, so you will be informed if the authoritative DNS server reports back an incorrect IP.

There are a number of options in the DNS Options tab.

- DNS Record type (A, CNAME, MX, NS, SOA, TXT)
- Match Options for IP address
- Recursive or non-recursive
- DNSSEC Validation

How to monitor DNS on a regionally partitioned DNS record

The good news is that DNS monitoring will still work as normal (so long as you have configured your server with an FQDN), since our monitoring nodes perform local DNS resolution on that domain name.

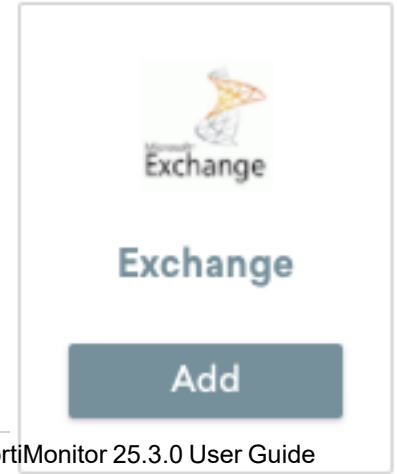
However this can be a confusing approach, so we recommend that you create a DNS service check for each one of your FQDN's IP addresses. This would mean creating a separate server (using the IP address, but not the FQDN) in your FortiMonitor control panel and setting up DNS monitoring for each. By doing this you will know where your DNS outages are coming from.

Exchange - OWA

In addition to the Exchange health metrics which are gathered by the Agent, you can ensure login sessions to OWA (Outlook Web Access) are operating properly as well.

Control Panel Configuration

Select Exchange from the monitoring catalog.



Network check

- OWA session

This check allows you to establish a session with your Exchange server by entering the credentials to the check configuration.

If you have any questions, please contact our support team.

FTP network check

Control Panel Configuration

The FTP check allows you to ensure that your FTP service is operating as expected. You can add an FTP check by selecting FTP from the monitoring catalog.



There are two options, a port check on port 21 and an FTP session check.

- FTP Port
- FTP Session

The session check allows you to set FTP credentials under the **FTP Options** tab.

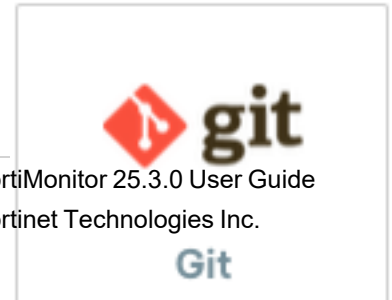
If you have any questions, please contact our support team.

Git

Monitor your Git repositories to ensure they are always accessible to your team with this Git network check. The check issues a git clone, downloads the codes, ensures the download completes and then immediately deletes the content

Control Panel Configuration

Select the Git plugin from the monitoring catalog.



There will only be one option in the check drawer.

- Git clone

The specialized configuration for this network check can be found under the **Git Options** tab.

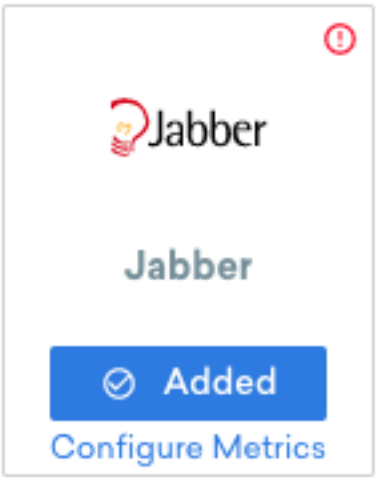
Here you can select the Git protocol, enter the Git Repo URL, and enter the SSH Private Key so the checker has access to download the content.

If you have any questions, please contact our support team.

Jabber

Control Panel Configuration

Monitor your public/private Jabber/XMPP server to ensure clients can connect. Select Jabber from the monitoring catalog.



Select the Jabber Port check.

- Jabber Port

This check monitors port 5222 to ensure that the Jabber service port is responsive.

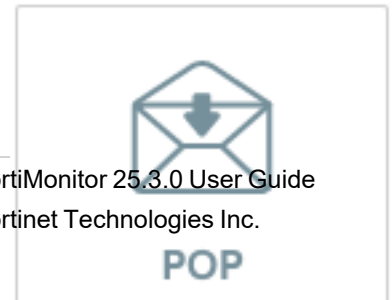
If you have any questions, please contact our support team.

Mail checks

FortiMonitor offers a number of network checks to monitor the various layers of an enterprise mail system. All mail protocols support a TLS/SSL enabled connection as well.

POP Check Configuration

Select POP from the monitoring catalog.



Choose from one of the below options for the POP network checks.

- POP Delivery Time
- POP Port
- POP Session
- POP Secure Delivery Time
- Secure POP Port
- Secure POP Session

POP Delivery Time

POP Delivery Time track the time it takes for an email sent from one of our checkers to be delivered to your inbox. The general POP Delivery Time checks operate over port 110, secure POP Delivery time checks operate over port 995. Each requires a special configuration in the **POP Email Options** tab.

POP Port Check

POP Port checks ensure that the POP Port is available for connections.
General POP Port checks operate over port 110, while secure POP Port checks are run against port 995.

POP Session Check

POP Session checks monitor the POP service to ensure that a session can be established. The general POP Session checks operate over port 110, and secure POP Session checks operate over port 995. There are also specialized POP Email Options that require additional configuration.

IMAP Check Configuration

Select IMAP from the monitoring catalog.



There are a few options to choose from.

- IMAP Delivery Time
- IMAP Port
- IMAP Session

- IMAP Secure Delivery Time
- Secure IMAP Port
- Secure IMAP Session

IMAP Delivery Time

The delivery time check allows you to test the time it takes for an email to go from our checkers to your mailbox. There is a special configuration required for this check that can be found in the **IMAP Email Options** tab.

General IMAP Delivery Time checks operate over port 143.

Secure IMAP Delivery Time checks operate over port 993.

IMAP Port

This check monitors port 143 to see if it is available to receive connections.

Secure IMAP Port checks operate over port 993.

IMAP Session

This check also uses special configuration to test establishing a session with your IMAP server. These specifications can be found under the **IMAP Email Options** tab.

Secure IMAP Session checks operate over port 993.

SMTP Check Configuration

Select SMTP from the monitoring catalog.



There are a couple of options for network checks.

- SMTP Port
- SMTP Session
- Secure SMTP Port
- Secure SMTP Session

SMTP Port

This check monitors the availability of port 25 and notifies you when it becomes unavailable.

SMTP Session

An SMTP Session check allows you to test establishing a session with your SMTP server to ensure functionality.

Secure SMTP Port

Secure SMTP Port checks operate over port 465.

Secure SMTP Session

Secure SMTP Session checks operate over port 465.

The Secure SMTP Port and Session checks use an encrypted connection at the transport layer. If you want to monitor the SMTP Submission services that commonly run on port 587, you should use the SMTP check and set the port to 587.

If you have any questions, please contact our support team.

MySQL network check

Control Panel Configuration

You can add a MySQL Network Check by clicking MySQL in the monitoring catalog.



There are Agent metrics for MySQL as well as Network checks.

- MySQL Port
- MySQL Session

The configuration for a port check is much simpler, as you are only checking port 3306 to see if it is responsive. The configuration for the session check is a bit more advanced as it requires MySQL Database Credentials.

These credentials will allow the checker node (whether part of our public fleet or your own OnSight) to log in and ensure a session can be established.

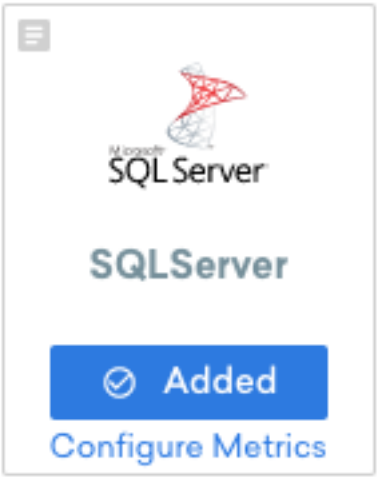
If you have any other questions, please reach out to FortiMonitor Support.

MSSQL network check

Monitor your Microsoft SQL database server from multiple dimensions. Start with internal health/performance metrics using the Agent and then ensure MSSQL client connections are succeeding using a network check from an OnSight vCollector or public monitoring probe.

Control Panel Configuration

To add the network check, select MSSQL from the monitoring catalog.



Select the MSSQL port check option.

- Network Check
- MSSQL Port

This check will monitor whether or not the MSSQL port is available and responsive.

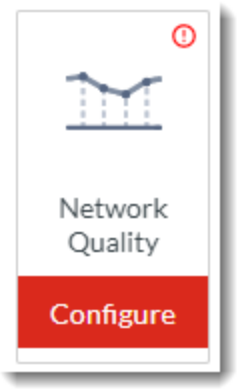
If you have any questions, please contact our support team.

Network quality

Network quality checks allow you to ensure that your network is running as expected. These metrics can be collected from public checkers. In future releases, support will be added for metric data collection using an OnSight vCollector or a DEM Agent.

Control Panel Configuration

You can add an **Network quality** check by choosing the **Network quality** card from the monitoring catalog:



Select a metric from the following options.

Metric	Unit	Description
Jitter	ms	Variation between packet delays
Latency	ms	Delays in network communication
Mean Opinion Score (MOS)	score	MOS score for VOIP monitoring
Packet Loss	%	Percentage of packets lost during transmission

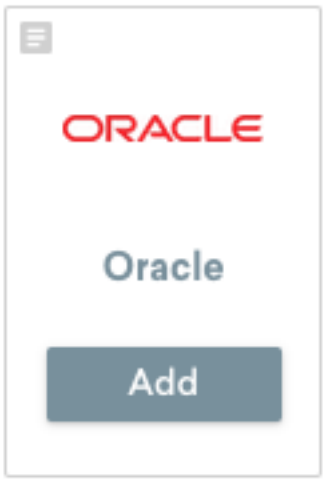
There are a number of other options to configure with the Network checks including the check frequency and from which location you would like the check to run. The check can be run from any of our public monitoring nodes

You can configure the **Alert Timeline** in the **Thresholds & CounterMeasures** tab and select the number of confirmation checks required to confirm an outage.

Oracle network check

Control Panel Configuration

You can add a Oracle Network Check by clicking Oracle in the monitoring catalog.



There are Agent metrics for Oracle as well as Network checks.

- Oracle Port

The Oracle port check sends a TCP/IP connection to the destination server on port 1521 (default port) and ensures the TCP connection is established.

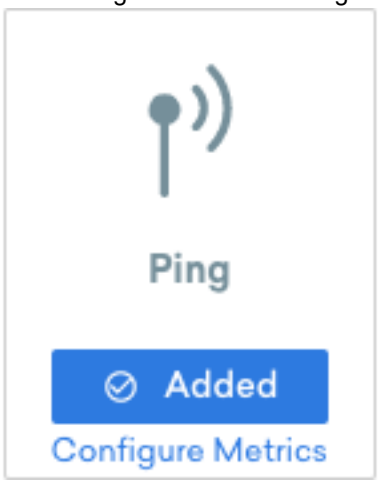
If you have any other questions, please reach out to FortiMonitor Support.

Ping

Ping checks are useful for understanding Layer 3 health and latency separate from your application. It's also the most basic of the uptime checks. In addition to latency, you can optionally monitor packet loss as well.

Control Panel Configuration

Select Ping from the monitoring catalog.



Check types

- Ping
- Jitter

There are a number of other options to configure with Ping including the IP or FQDN you will be checking, and from which location you would like the check to run. The check can run from any of our public monitoring nodes, or an OnSight vCollector if you have one set up. In addition, you can also configure the following:

- Packet Loss Threshold - For ping checks. Enter the percentage of packet loss allowed before an incident is considered..
- Packet Count - For Jitter checks. The number of packets to send.

If you have any questions, please contact our support team.

PostgreSQL network check

Monitor your Postgres database server from multiple dimensions. Start with internal health/performance metrics using the Agent and then ensure Postgres client connections are succeeding using a network check from an OnSight vCollector overview or public monitoring probe.

Control Panel Configuration

To add the network check, select PostgreSQL from the monitoring catalog.



Select the port check option:

- PostgreSQL Port

This check will monitor port 5432 (or whichever port you set to your PostgreSQL service to) to let you know if your PostgreSQL service is responsive.

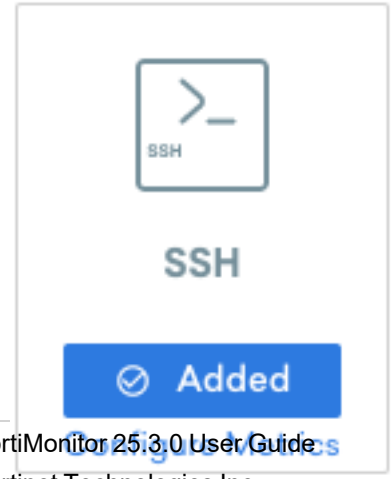
If you have any questions, please contact our support team.

SSH

SSH checks allow you to monitor the SSH service either by checking to see if the port is open, or if a session can be established. SFTP sessions can also be monitored.

Control Panel Configuration

You can add an SSH check by choosing the SSH card in the monitoring catalog.

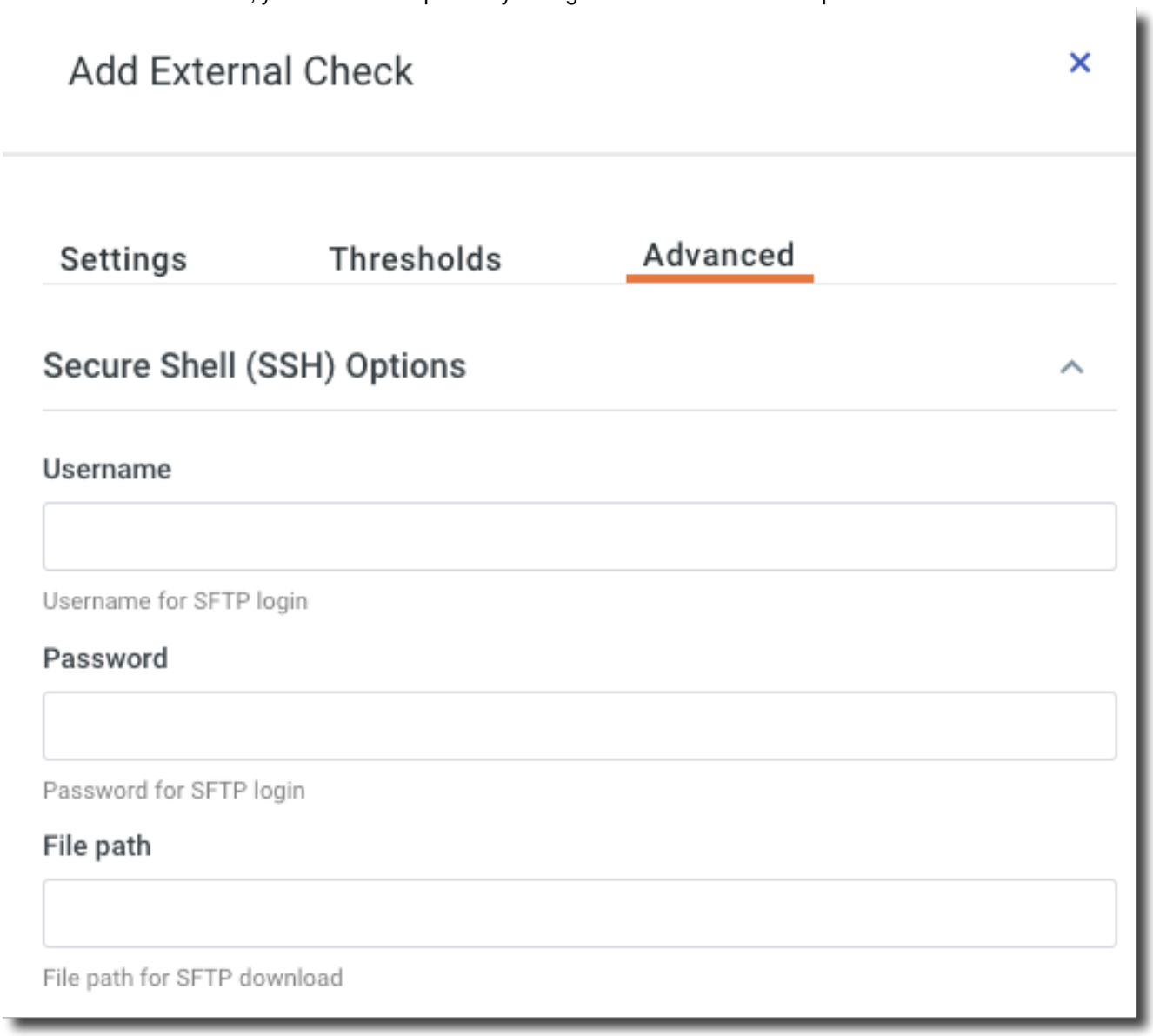


Check Type

- SSH Port
- SSH Session
- SFTP Session

There are a number of other options to configure with SSH including the IP or FQDN you'll be checking, and from which location you would like the check to run. The check can run from any of our public monitoring nodes, or an OnSight vCollector if you have one set up.

For SFTP session checks, you also need to provide your login credentials and the file path for the SFTP location.



You can configure the **Alert Timeline** in the **Thresholds** tab and select the number of confirmation checks required to confirm an outage (we suggest keeping the default).

You can also set up thresholds for the check in the **Thresholds** tab if you want to be alerted if the response time takes longer than X seconds for more than X amount of time.

An SSH Session Check is a little more involved, and includes SSH Options in the **Advanced** tab, allowing you to select a username and password, and a command to execute after a successful login.

If you have any questions, please contact our support team.

SSL expiration check

The SSL certificate expiration metric allows you to create an alert based on the number of days remaining before a certificate's expiration date.

Control Panel configuration

To get started, perform the following steps:

1. From the Control Panel, go to the **Instance Details** page of the instance where you want to add SSL expiration monitoring.
2. In the **Monitoring Config** tab, click **Add Monitoring**. The monitoring catalog will be displayed.
3. Select **HTTPS** from the monitoring catalog.

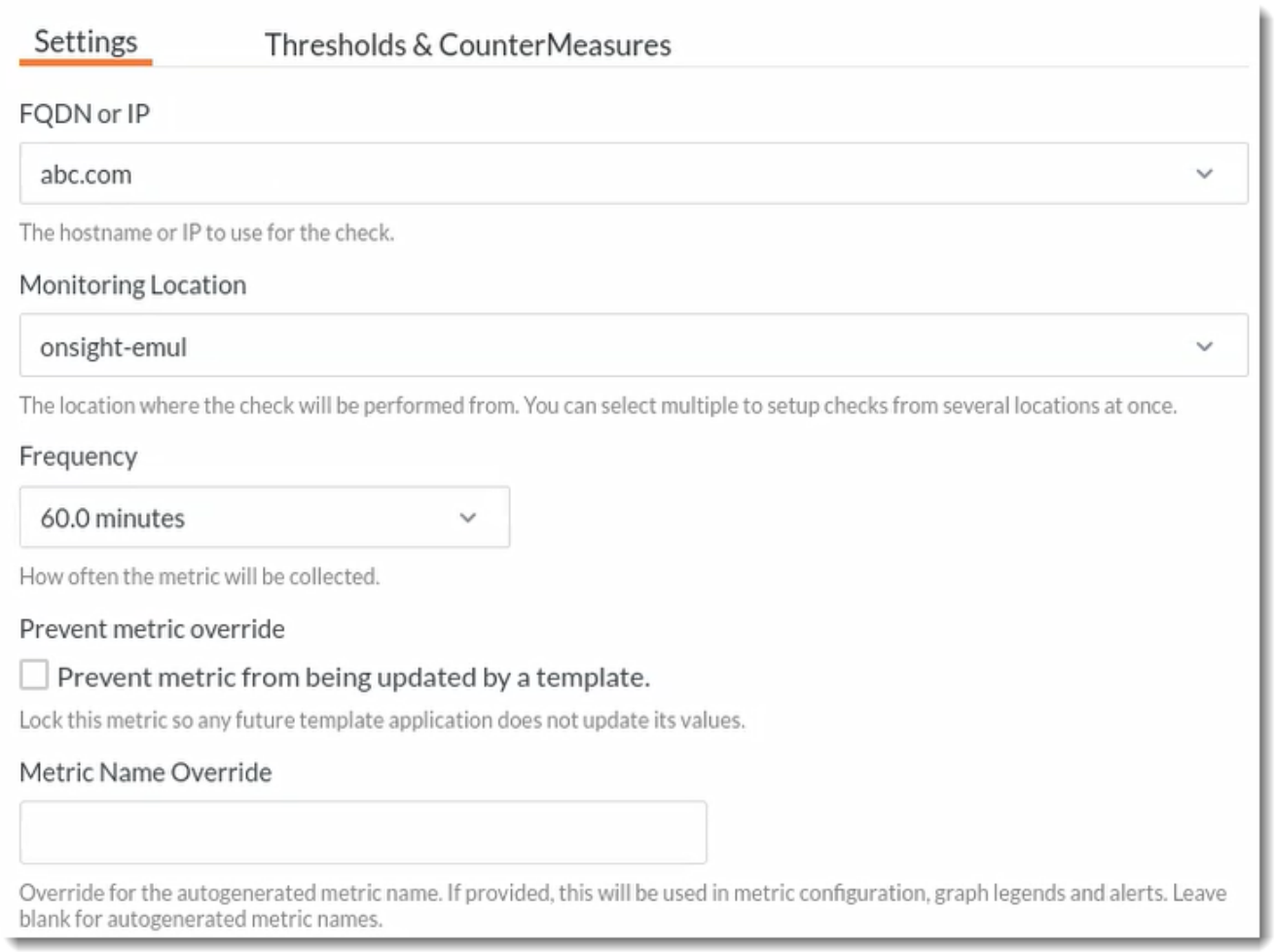


4. Under **SSL**, select **SSL Expiration**.

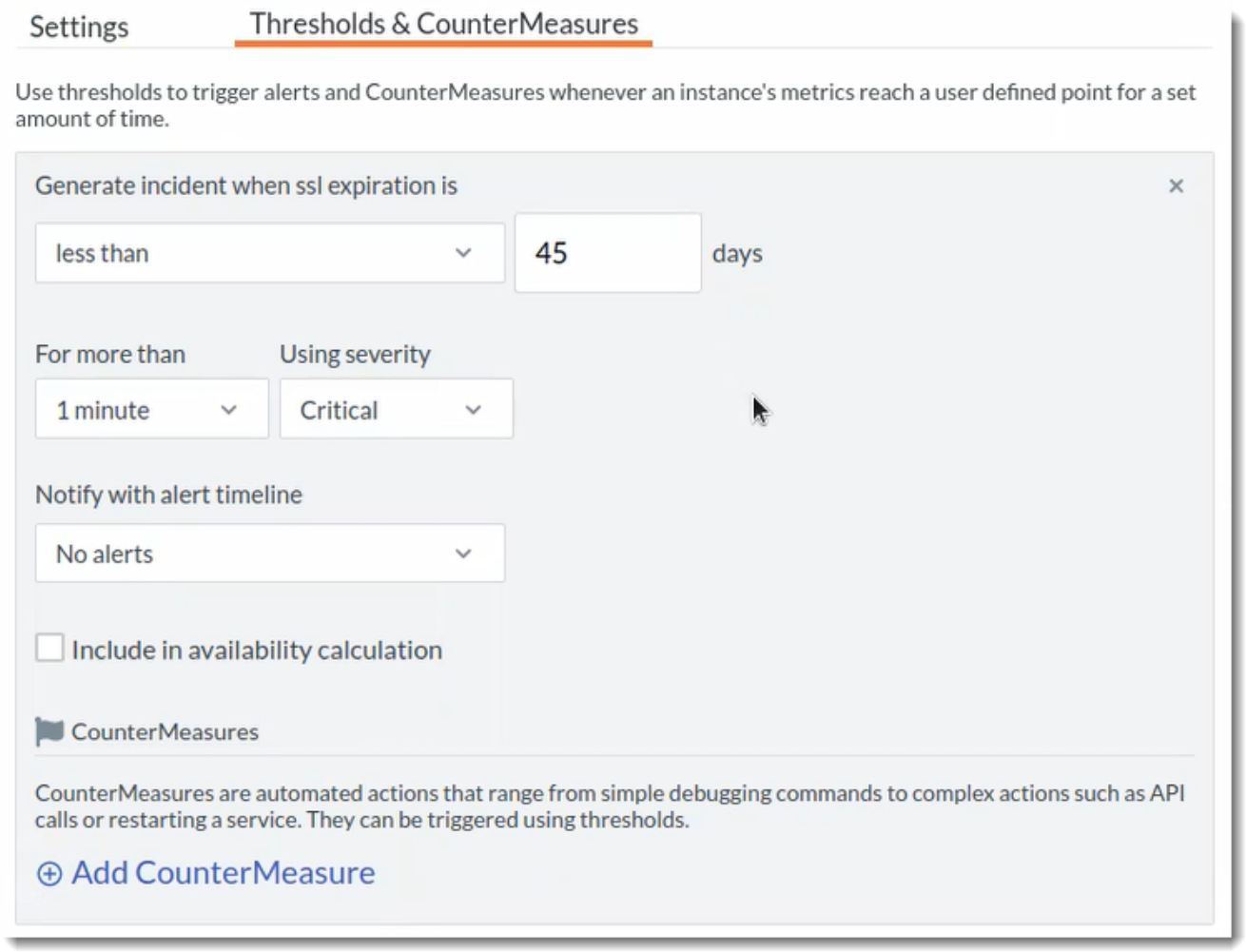


SSL expiration configuration

There are a number of other options to configure with SSL expiration including the IP or FQDN you will be checking, and from which location you would like the check to run.



The **Thresholds & CounterMeasures** tab allows you to configure when an incident for an expiring SSL certificate will be generated. In the following example, an alert will be triggered when there is less than 45 days before a certificate's expiration date.



You can configure the Alert Timeline in the **Thresholds & CounterMeasures** tab.

You can also enable the **Include in availability calculation** option if you want to include this alert to your server's availability calculation.

TCP network checks

Control Panel Configuration

Add a TCP/IP check by selecting TCP from the monitoring catalog.



There will only be one option, select Add. The configuration options will open, giving you an opportunity to name the check, and select the port to run the TCP check against, as well as from which location.

There are other tabs as well, to select the alert timeline, set any thresholds, and option to tag the check. Once you have the check configured to your liking, click **Save**.

If you have any questions, please contact our support team.

Synthetic monitoring

Synthetic checks can be run from any of our monitoring locations. You can also use the OnSight vCollector to run Synthetic Monitoring checks from within your private network.

Synthetic Monitoring check types

Basic synthetic checks

The following check types are available:

- HTTP/HTTPS port
- HTTP/HTTPS HEAD
- HTTP/HTTPS check
- NTP check

HTTP and HTTPS checks are simple tests to see if a destination, such as a port, website, content, or more, is responding correctly.

HEAD checks are used to ask only for information about the page and not the whole page itself. This can be useful if you do not want to use up your servers' bandwidth or use unnecessary resources.

See HTTP and HTTPS for more information.

Browser synthetic checks

Browser synthetic checks validate user interaction with your web applications using Selenium. The output of browser synthetic checks can help you validate that your application is enabling the outcome you expect for customers. The following details are provided:

- The status of each executed step
- The command and the target of each step
- The elapsed time to execute each step
- Screenshots of each step

Use Katalon Recorder extension for Chrome to simulate a user session.

See Browser synthetic (multistep) checks for more information.

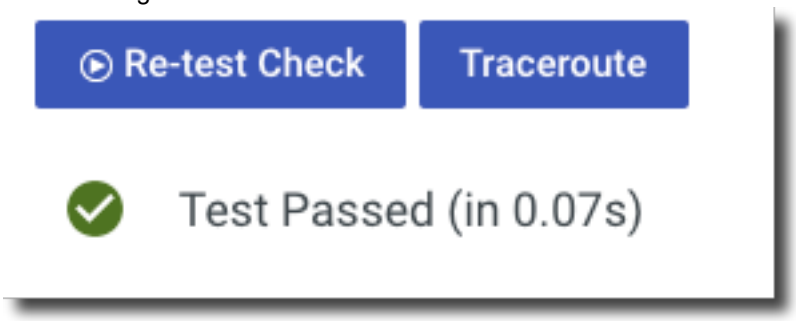
JavaScript synthetic checks

JavaScript synthetic checks are used for advanced tests that involve complex actions. To learn more about JS synthetic checks see JavaScript synthetic checks.

Test checks and Traceroute

Synthetic Monitoring in FortiMonitor also offers a way for you to check your synthetic checks directly from a metric's configuration drawer. To perform a Test Check, simply click the **Test Check** button and the test that you have configured will be performed in real-time.

The following screenshot shows the result of an HTTP check:



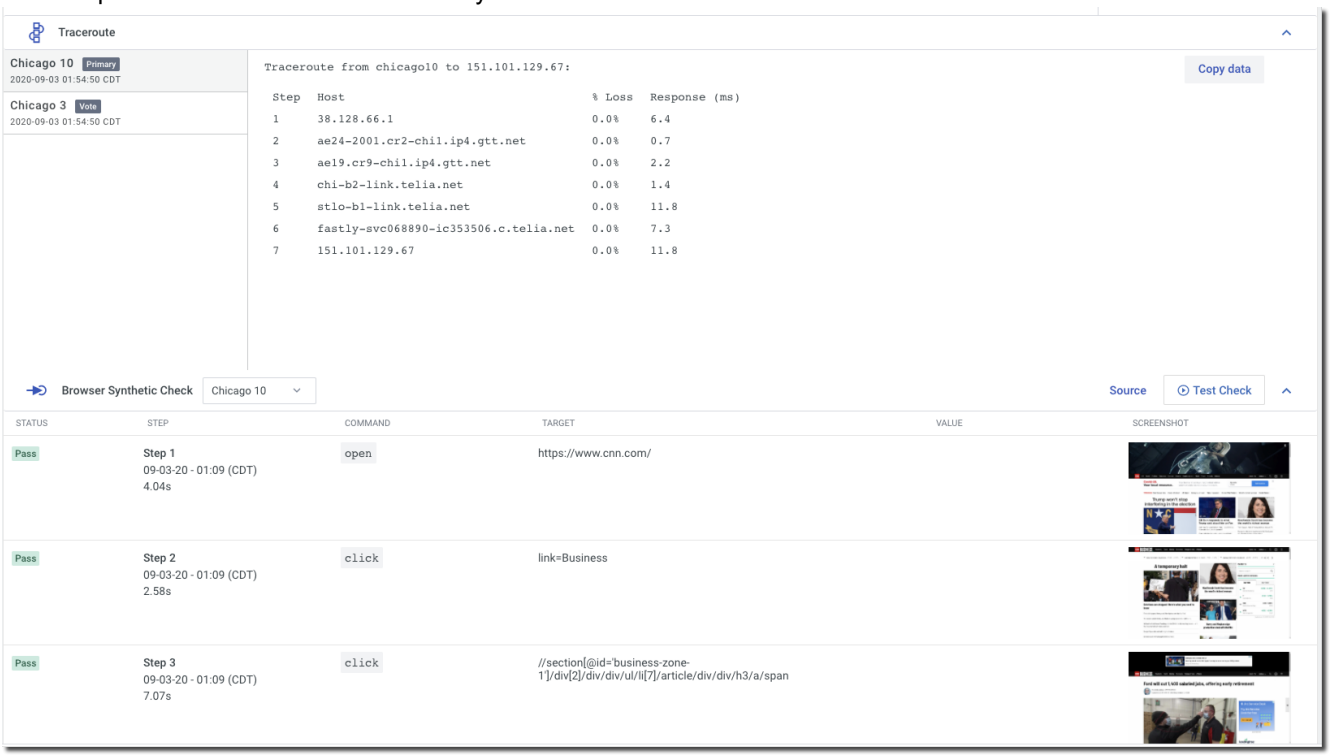
Traceroute tracks in real-time the pathway taken by a packet from source to destination. This can help you identify any delays or dropped packets when transmitting data. The following figure shows an example of a Traceroute output:



HOP	HOST	LOSS	AVG
1	146.71.78.254	0.0%	0.3 ms
2	199.116.115.246	0.0%	1.5 ms
3	te0-7-0-13.rcr21.b002281-5.ord03.atlas.cogentco.com	0.0%	1.7 ms
4	be2409.ccr41.ord03.atlas.cogentco.com	0.0%	2.0 ms
5	telia.ord03.atlas.cogentco.com	0.0%	3.2 ms
6	nyk-b3-link.telia.net	0.0%	26.3 ms
7	***	100.0%	0.0 ms
8	***	100.0%	0.0 ms
9	***	100.0%	0.0 ms
10	***	100.0%	0.0 ms
11	***	100.0%	0.0 ms
12	pkg02.panopta.com	0.0%	20.5 ms

Incidents

Active Synthetic Monitoring checks will create incidents for any test that fails. The incidents will be available in the **Incident Hub** or the **Instance Details page** of the instance where you configured the check. The following image shows an example of an incident from a browser synthetic check.



HTTP

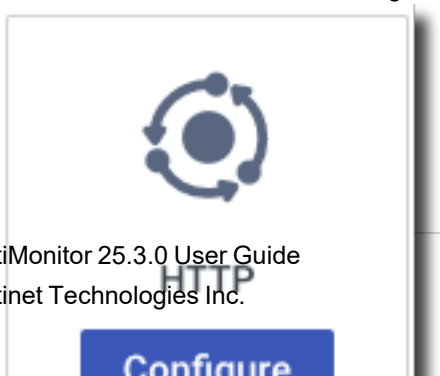
The HTTP check has multiple check types that fulfill different services.

- HTTP port
- HTTP HEAD
- HTTP check

Control Panel configuration

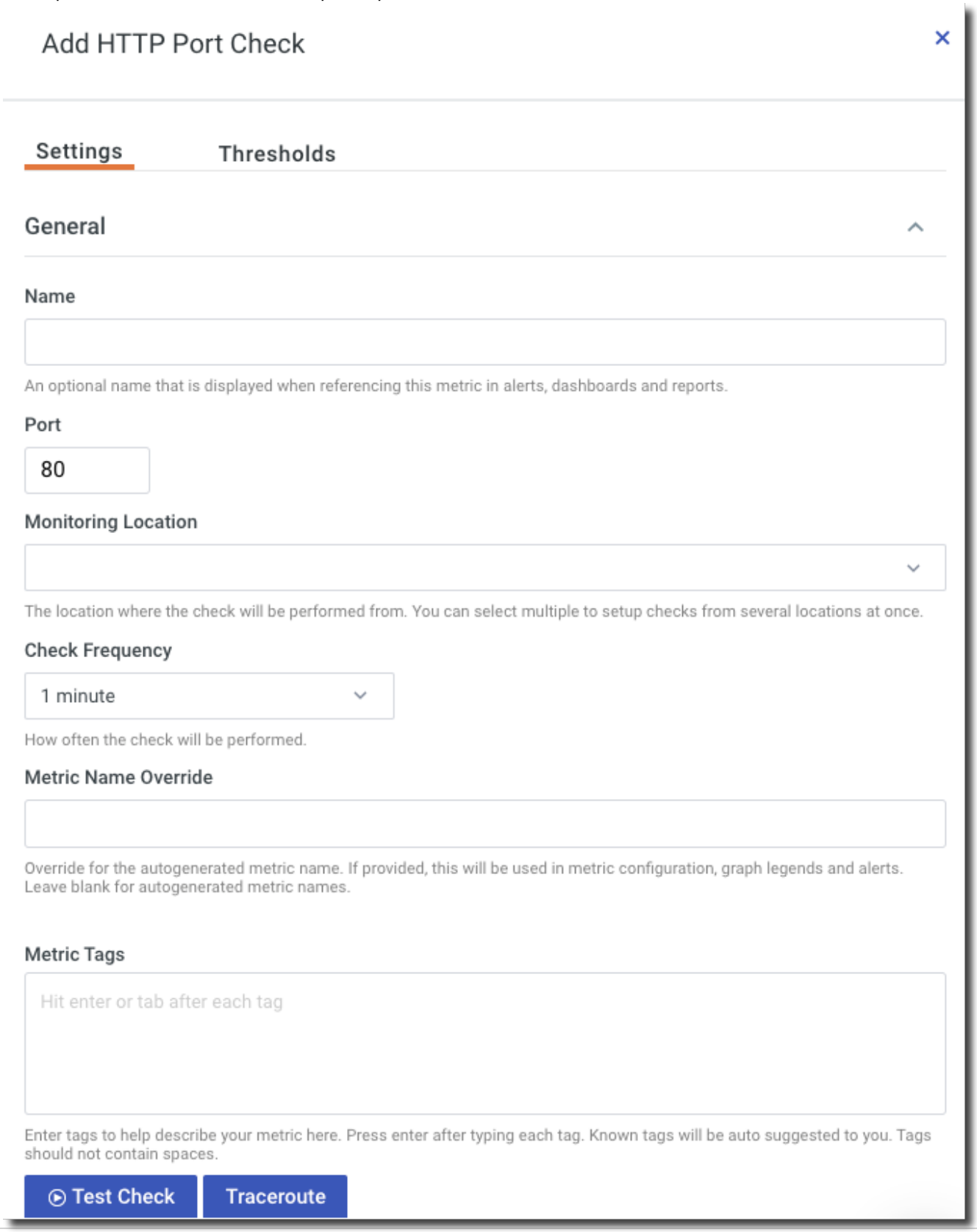
To get started, perform the following steps:

1. From the Control Panel, go to the **Instance Details** page of the instance where you want to add HTTP monitoring.
2. In the **Monitoring Config** tab, click **Add Monitoring**. The **Monitoring Catalog** will be displayed.
3. Select **HTTP** from the monitoring catalog.



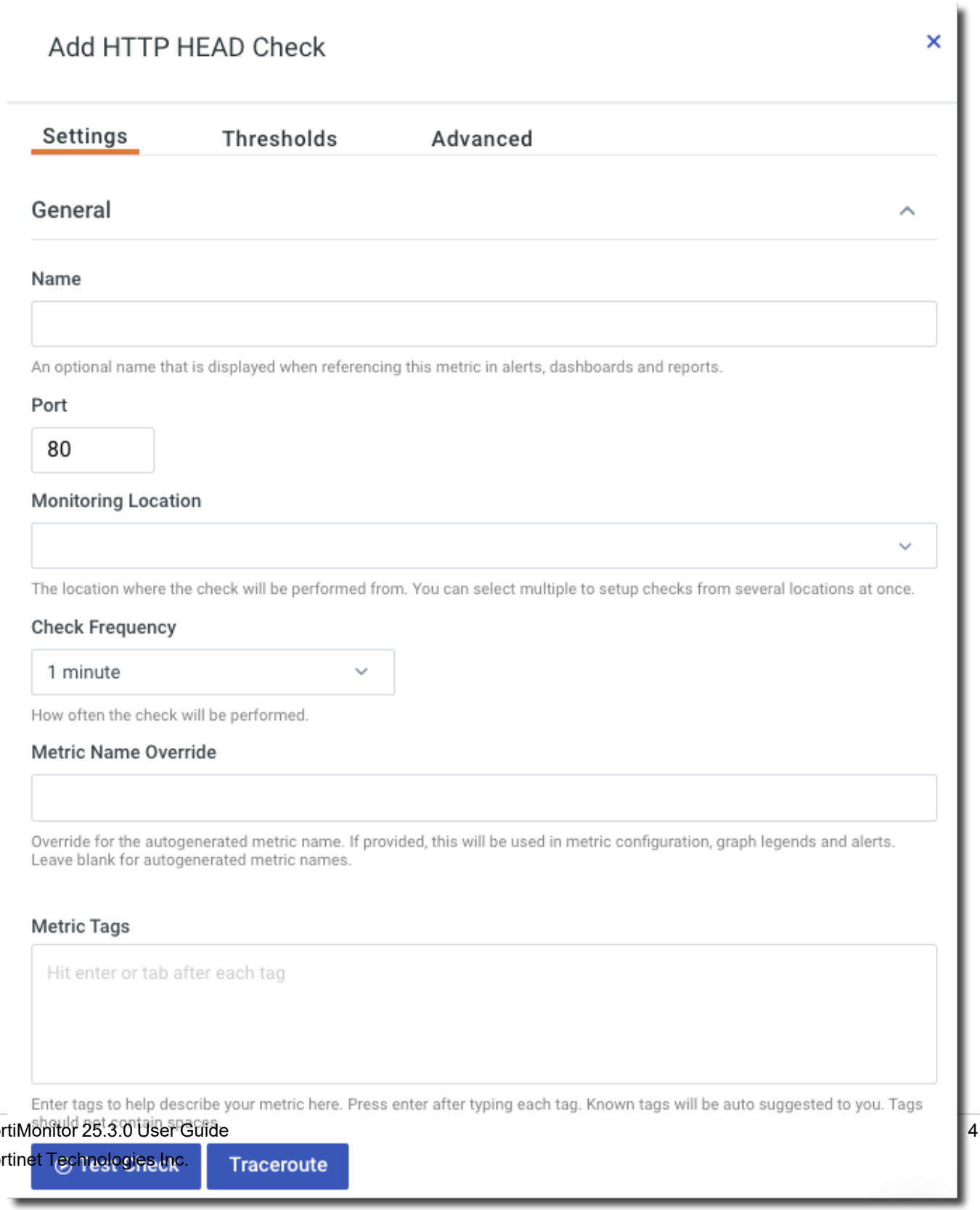
HTTP port configuration

HTTP port checks are used to test if a specific port is available.



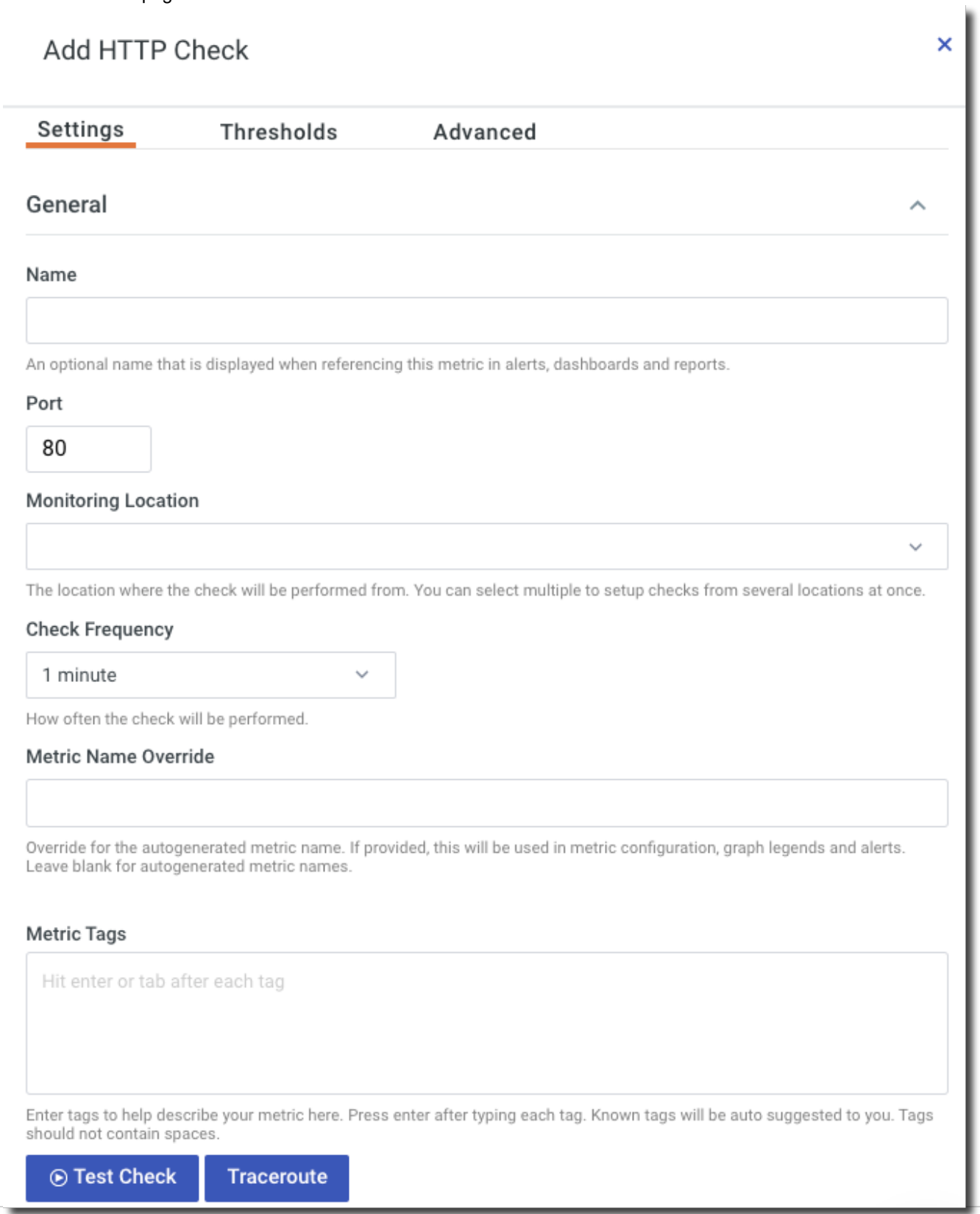
HTTP HEAD configuration

While the HTTP Port check tests if a port is reachable, the HTTP HEAD check tests if a HEAD request will return its appropriate response. A HEAD request is used to ask only for information about the page and not the whole page itself. This can be useful if you do not want to use up your servers' bandwidth or use unnecessary resources. If the HTTP HEAD check results in a 200 or a 300 status code, then the service is considered up, if it results in a 4xx or a 5xx, then the service is considered down.

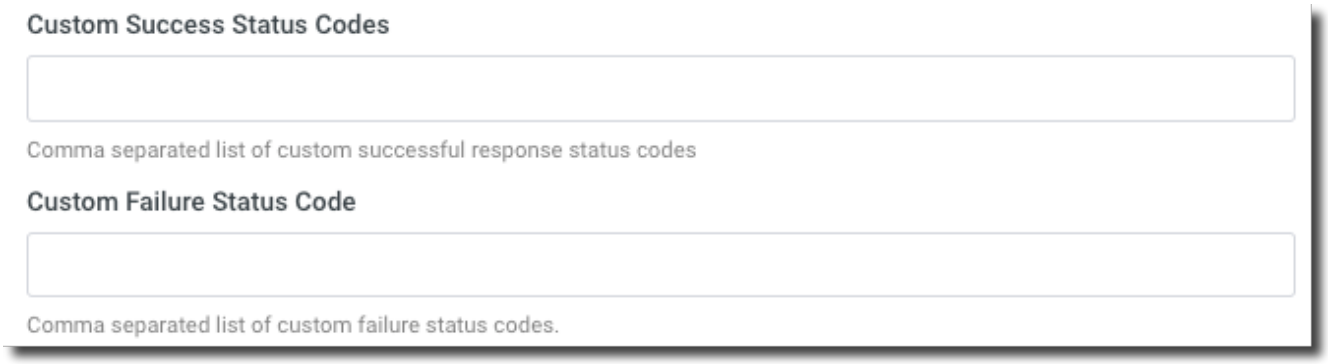


HTTP check configuration

The HTTP check configuration allows you to optionally specify the URI/path to the resource as well as check for specific content on a webpage.



If the HTTP check result in a 200, then the check is considered successful, if it results in a 4xx or a 5xx, then the check is considered a failure. You can also create custom success and failure status codes. In the **Advanced** tab, look for the **Custom Success Status Codes** or **Custom Failure Status Code** and enter a comma-separated list of your custom status codes.

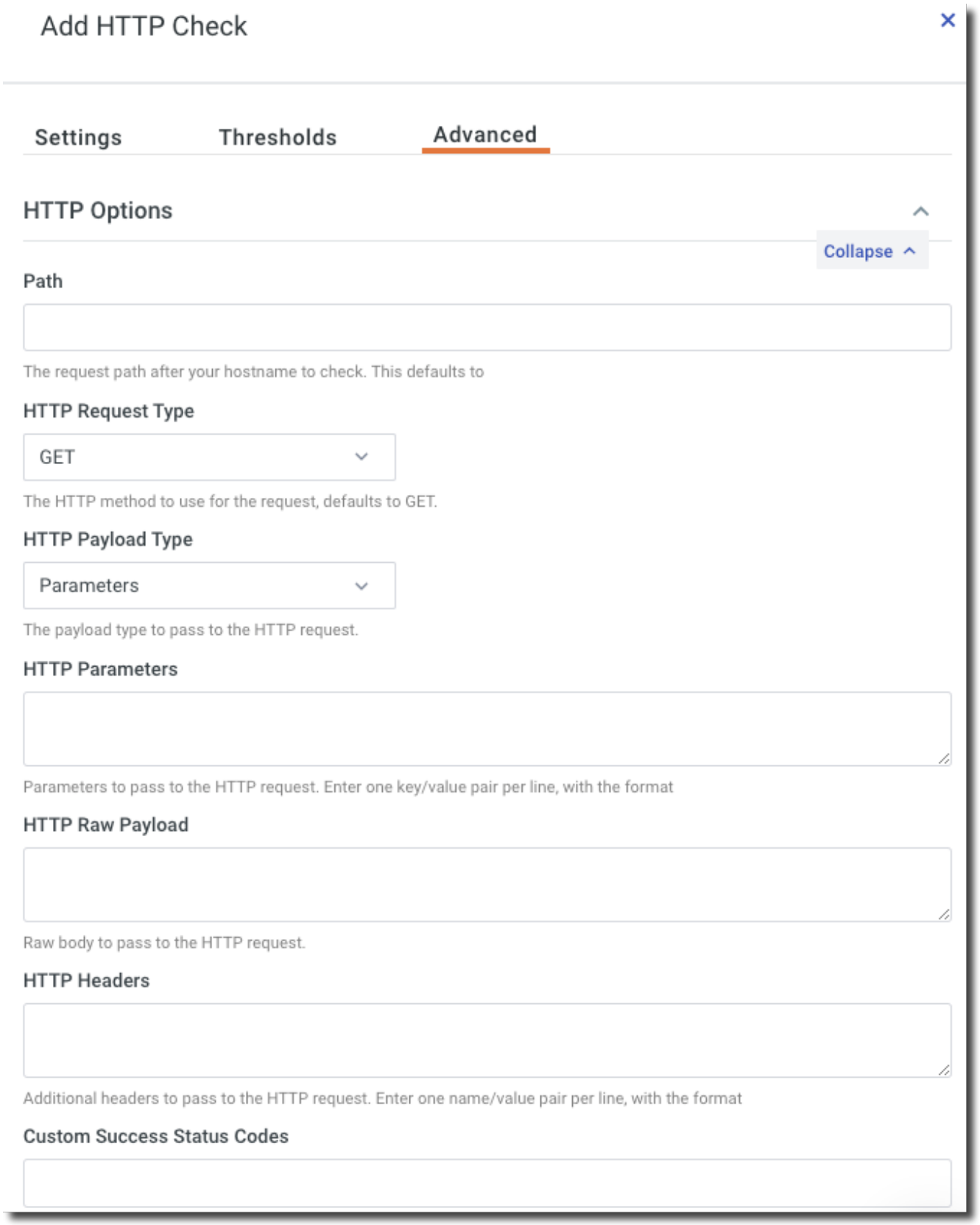


You can also optionally configure thresholds for this metric. See Configure incident alert and response time thresholds for more information.

Resources with one or more redirects may report higher load times than a browser experience.

Monitoring a specific endpoint

The **HTTP Options** module of the **Advanced** tab lets you add a specific path to check.


Add HTTP Check
Settings
Thresholds
Advanced
HTTP Options
Collapse
Path
The request path after your hostname to check. This defaults to
HTTP Request Type
GET
The HTTP method to use for the request, defaults to GET.
HTTP Payload Type
Parameters
The payload type to pass to the HTTP request.
HTTP Parameters
Parameters to pass to the HTTP request. Enter one key/value pair per line, with the format
HTTP Raw Payload
Raw body to pass to the HTTP request.
HTTP Headers
Additional headers to pass to the HTTP request. Enter one name/value pair per line, with the format
Custom Success Status Codes

For example, if you wanted to monitor http://docs.fortinet.com/fortimonitor/ you would create a server with a fully qualified domain name (FQDN) of **www.fortinet.com** and set the **Path** field to **/fortimonitor/**. This functionality works with both GET and POST requests. For GET requests, the path can contain additional parameters (i.e. **/fortimonitor/?source=web**). For POST requests, additional key/value pairs can be set in the **HTTP Parameters** field.

Content Matching

Content matches can either be straightforward string matches or, for more complicated matching can be full regular expressions. Our infrastructure uses the Python regular expression module; see the Python docs for details on the exact syntax supported.

Content matching allows you to do a number of in-depth checks such as looking for error messages that might get included on a page. You can also use this functionality combined with the ability to perform HTTP POST operations to perform a login transaction to a web application and make sure the login succeeds and returns the proper welcome message. This way, you can ensure both your application server and underlying database are working properly.

HTTPS

HTTPS checks operate like our HTTP checks but with the addition SSL/TLS support.

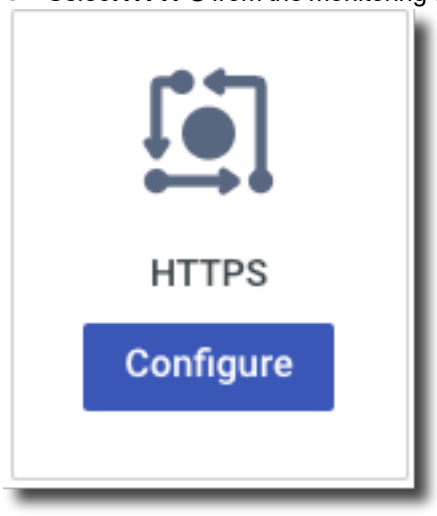
The following HTTPS checks are available:

- HTTPS port
- HTTPS HEAD
- HTTPS check
- HTTPS time

Control Panel configuration

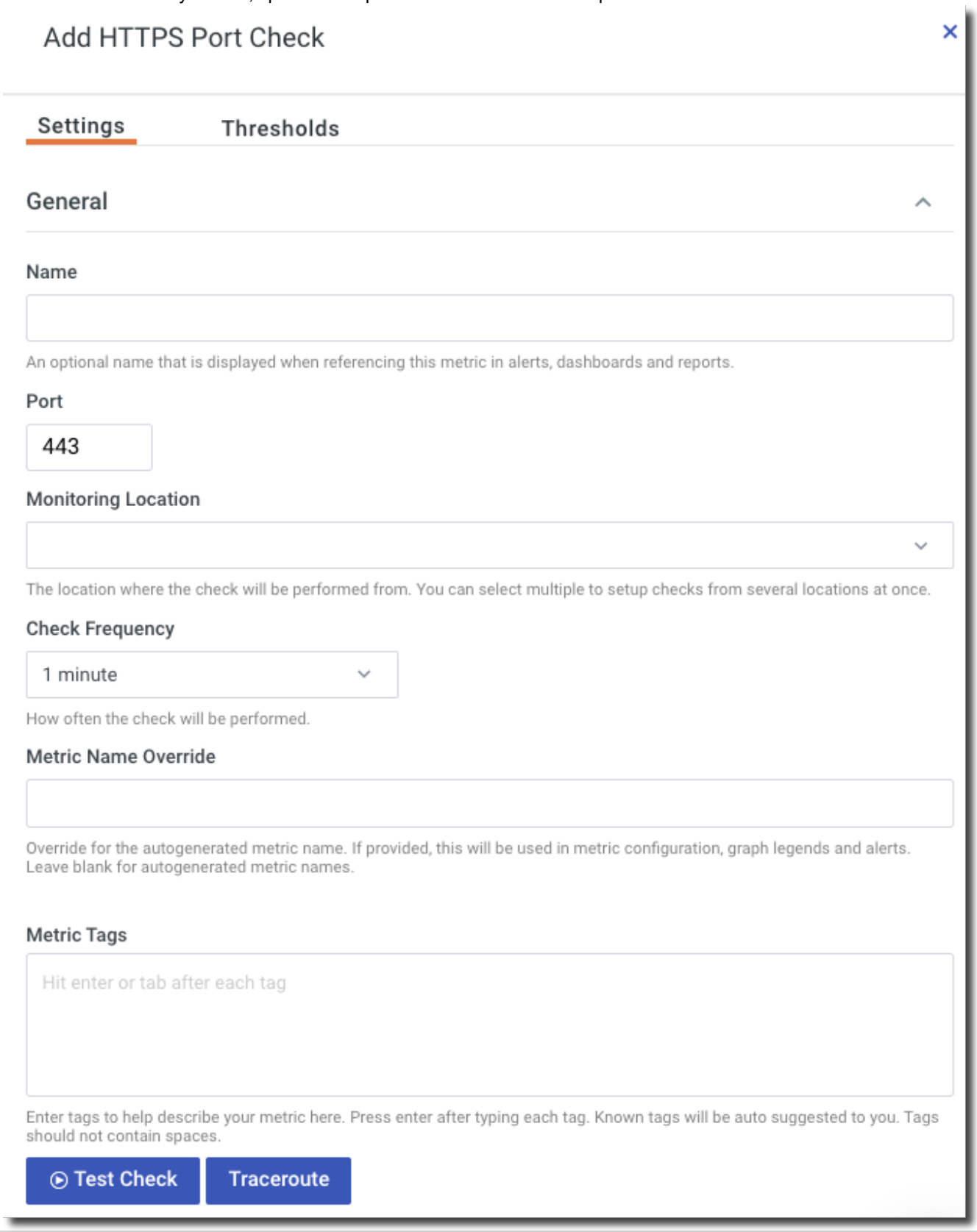
To get started, perform the following steps:

1. From the Control Panel, go to the **Instance Details** page of the instance where you want to add HTTPS monitoring.
2. In the **Monitoring Config** tab, click **Add Monitoring**. The **Monitoring Catalog** will be displayed.
3. Select **HTTPS** from the monitoring catalog.



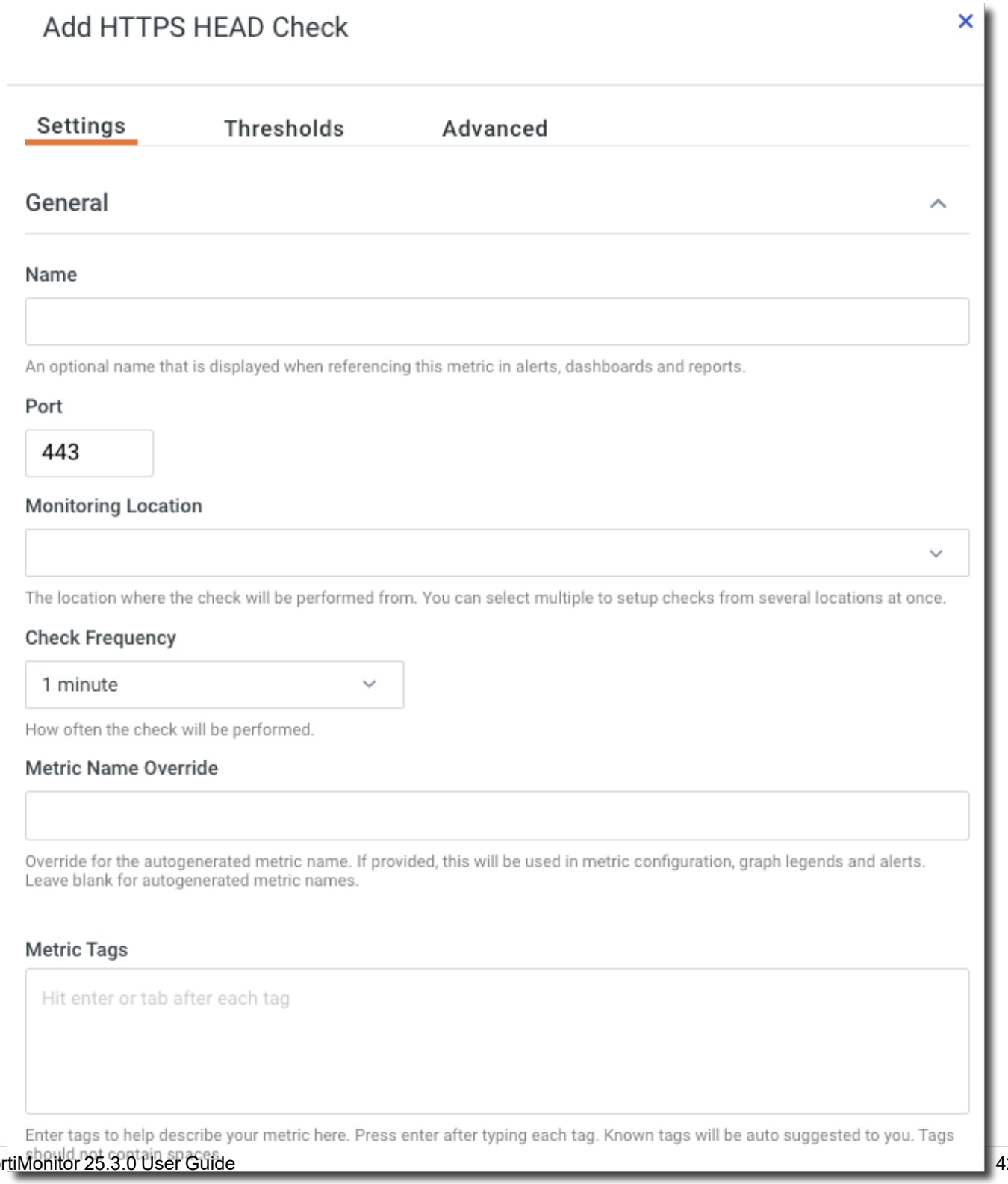
HTTPS port configuration

HTTPS Port checks by default, operate over port 443 and ensure that the port is available for connections.



HTTPS HEAD configuration

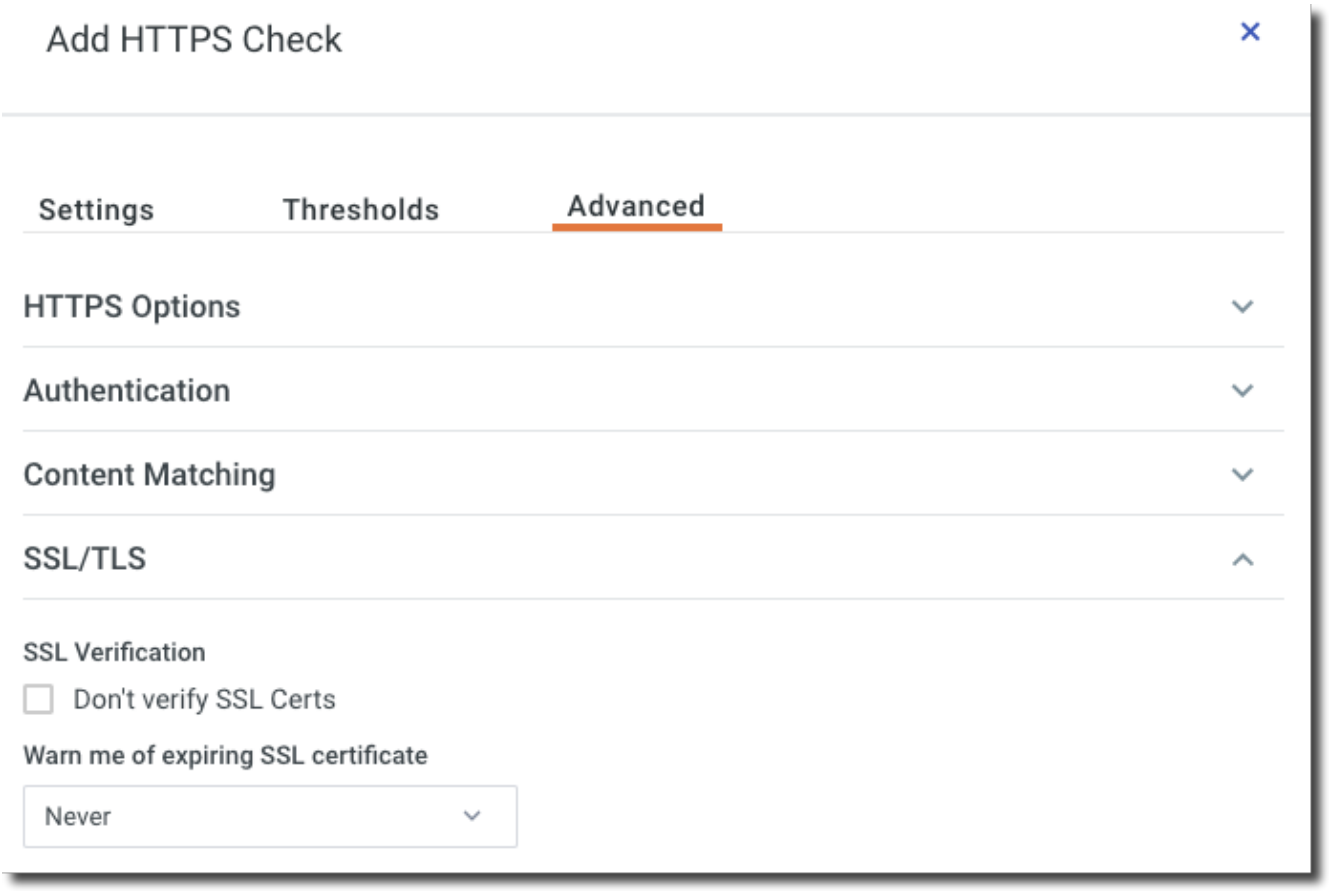
While the HTTPS Port check only tests if a port is reachable, the HTTPS HEAD check tests if a HEAD request will return its appropriate response. A HEAD request is used to ask only for information about the page and not the whole page itself. This can be useful if you do not want to use up your servers' bandwidth or use unnecessary resources. If the HTTPS HEAD check results in a 200 or a 300 status code, then the service is considered up, if it results in a 4xx or a 5xx, then the service is considered down.



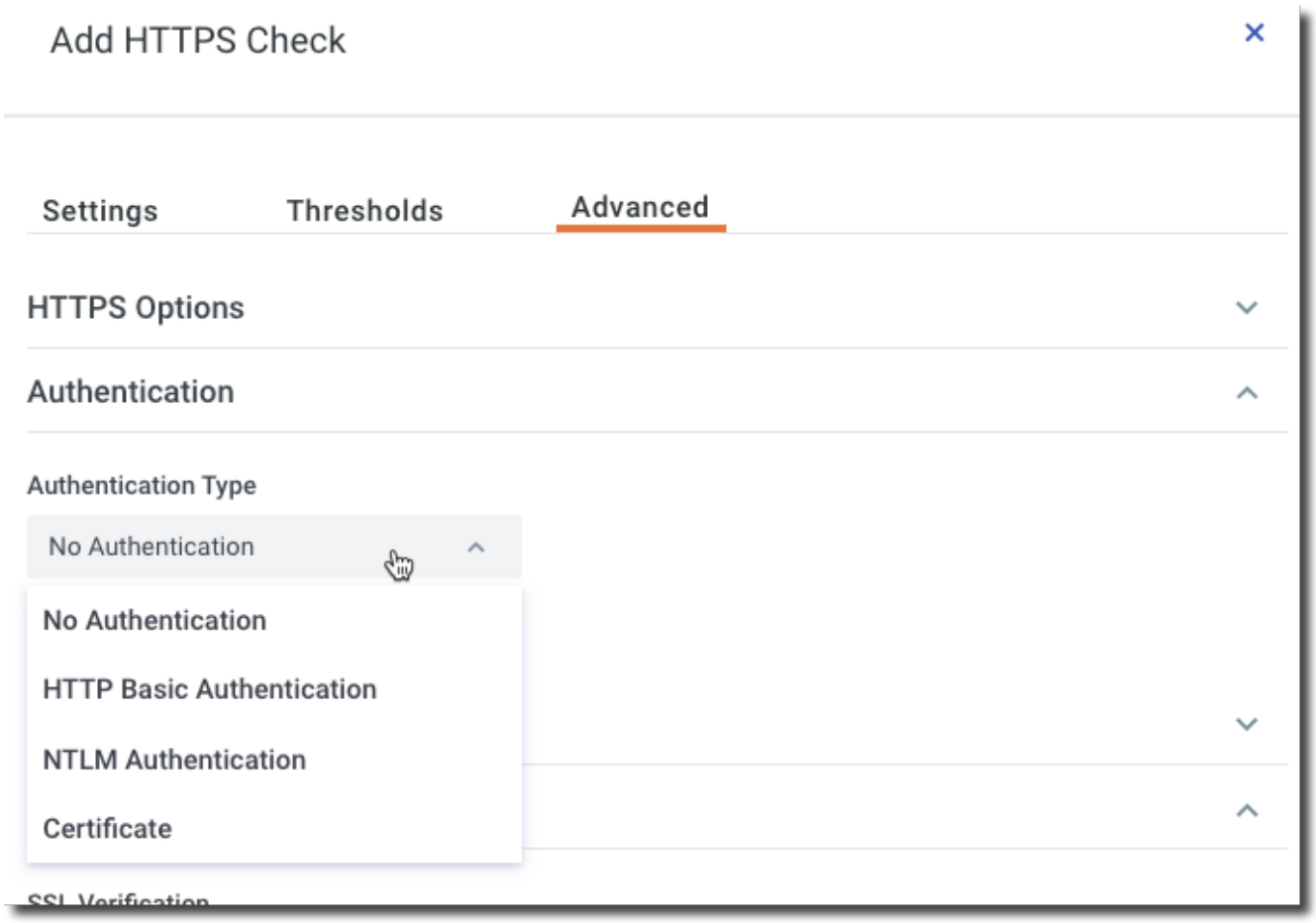
HTTPS check configuration

SSL validation is disabled for HTTPS checks provisioned by an OnSight vCollector.

Our HTTPS checks have all the same options as HTTP but with the addition of SSL/TLS specific options. Within the **SSL/TLS module**, you can enable/disable certificate strictness and schedule warnings for the expiration of SSL certificates.



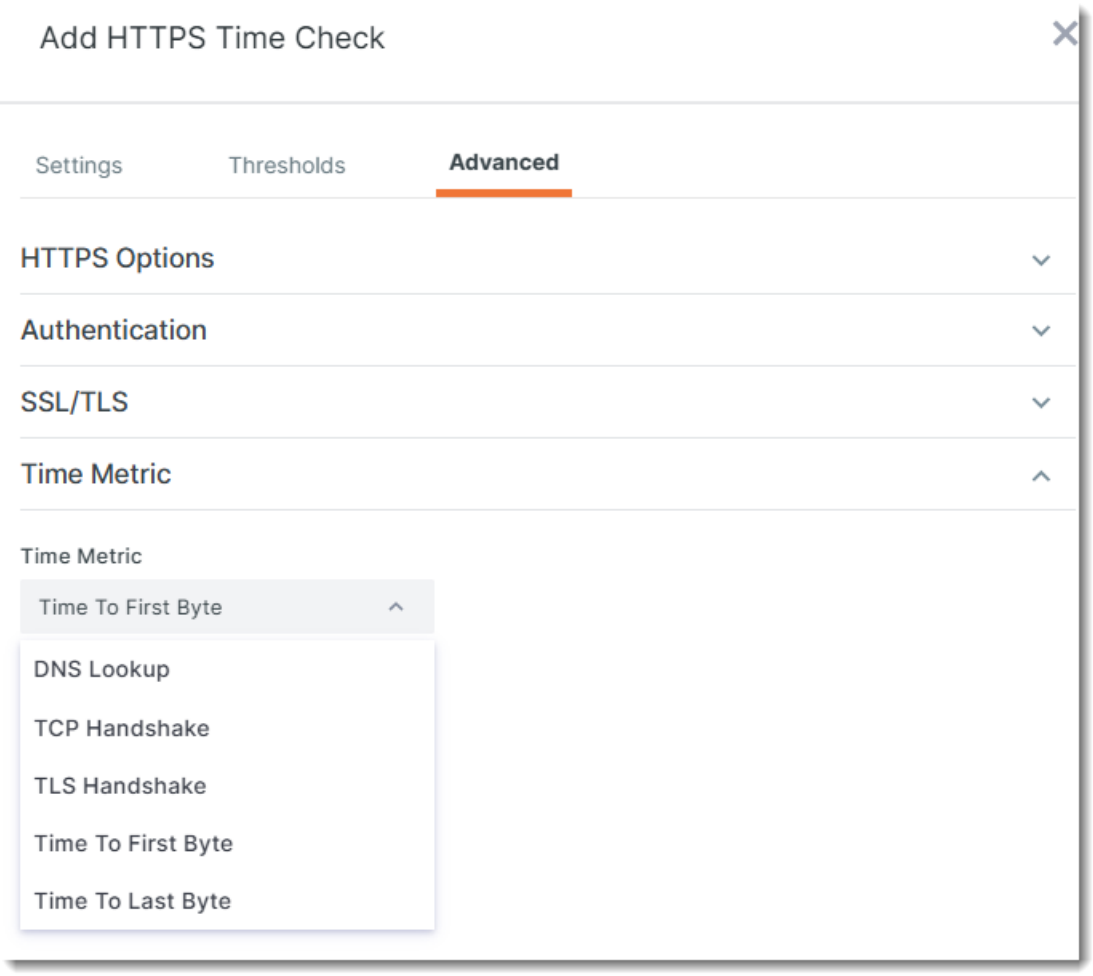
You can also add the client-side certificate and the certificate private key within the **Authentication** module.



HTTPS time configuration

HTTPS time checks allow you to measure the amount of time for a particular step in HTTPS requests. The following options are available:

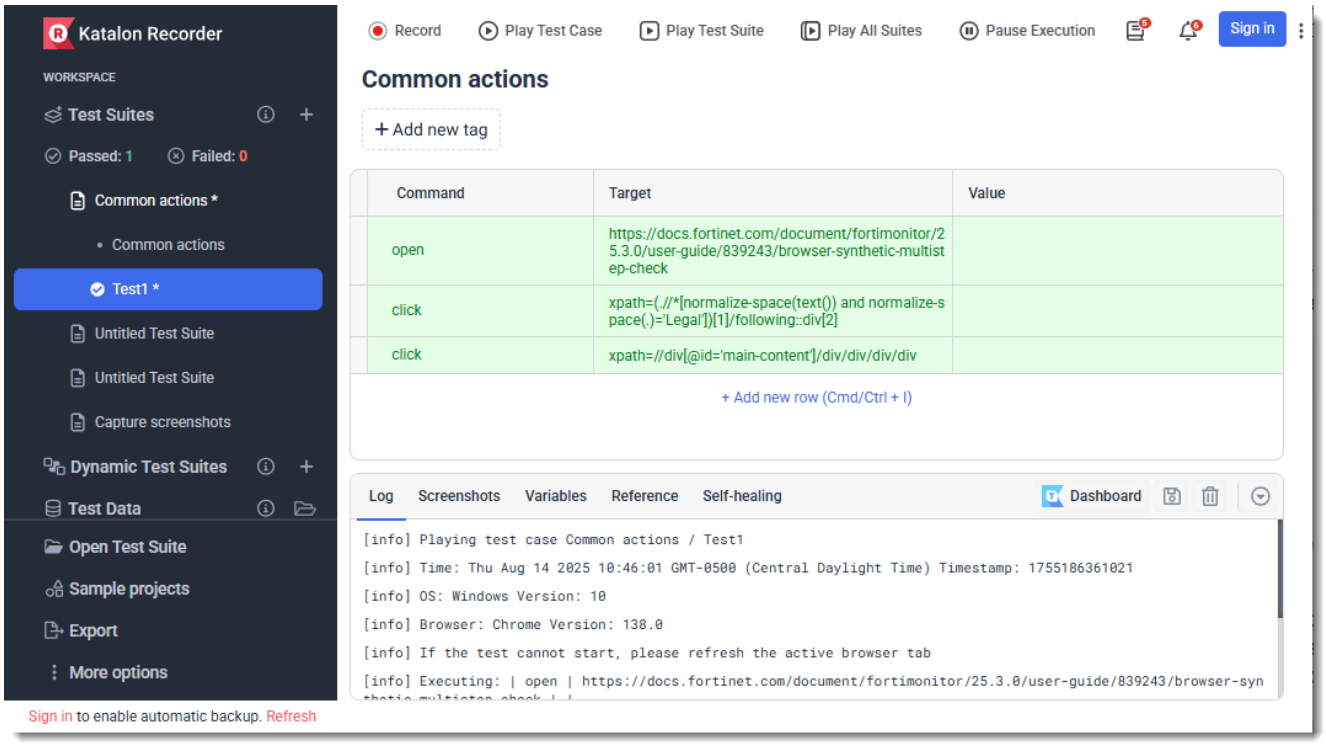
- DNS Lookup
- TCP Handshake
- TLS Handshake
- Time To First Byte (Default)
- Time To Last Byte



Browser synthetic (multistep) check

Browser synthetic monitoring allows you to simulate user interactions with your web applications, providing important insights into the application's availability and performance. Before you get started, download the Katalon Recorder extension for Chrome. Katalon records browser interactions and outputs them into a format that can be executed by FortiMonitor. Use this to either create simple scripts or assist in exploratory testing.

Below is what the Katalon IDE plugin looks like when opened:



Record browser interaction using Katalon

Once you have added the Katalon extension, record a browser interaction by performing the following steps:

1. Allow Katalon to start browsers in private mode.
2. Make sure to clear your cookies and browsing data.
3. Open a browser in private/ incognito mode.
4. Start the extension from your browser's extensions menu.
5. Click the record button to start capturing your clicks and key presses.
6. Katalon will record all of your browsing actions until you click **Stop**. To learn more Katalon Docs.
7. Once your test session is complete, export the test as XML by clicking **Export**.
8. Under **Format**, select **XML**.
9. Click **Copy to Clipboard** or save the file. This source code will be used to configure the check in the Control Panel.
10. Go to your Control Panel and start setting up the check. See the following section.

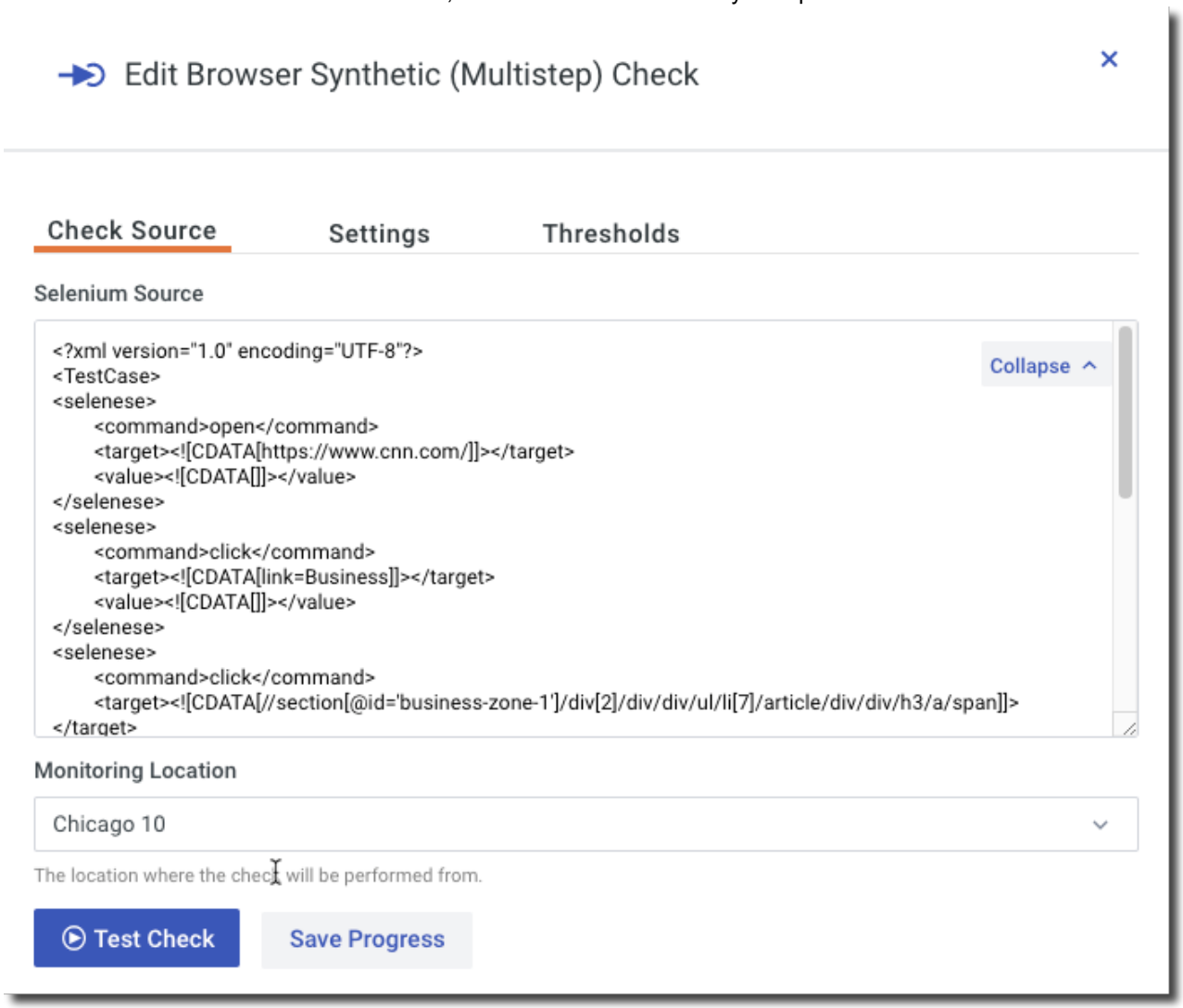
For a list of supported multistep commands, see Available multistep commands.

You may run into browser issues when recording in incognito mode. Allowing 3rd party cookies in your incognito session will help alleviate that.

Metric configuration

1. From the Control Panel, go to the **Instance Details** page of the instance where you want to add browser synthetic monitoring.

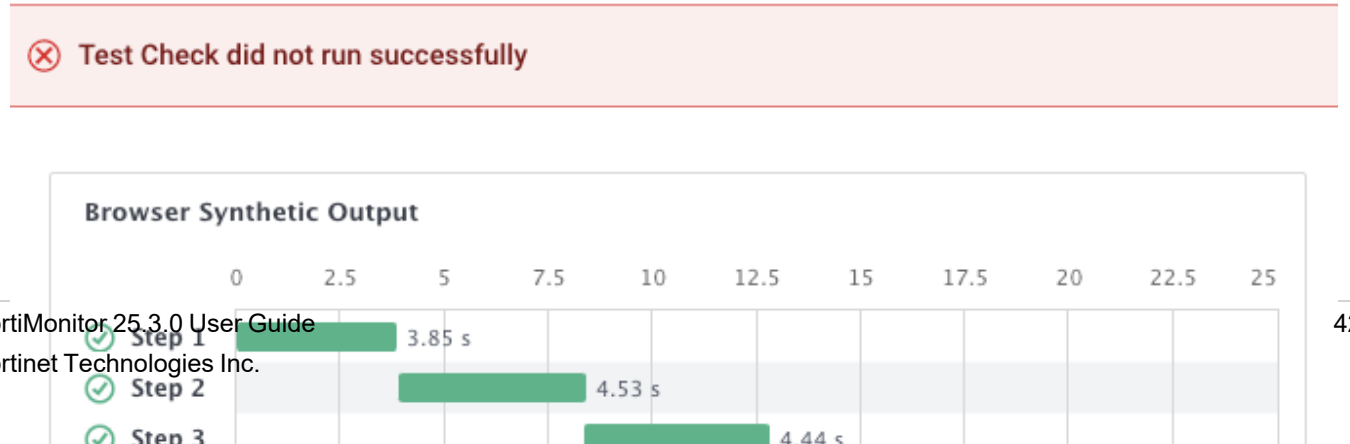
2. In the **Monitoring Config** tab, click **Add Monitoring**. The **Monitoring Catalog** will be displayed.
3. Select **Configure** under the **HTTP** or **HTTPS** cards. If you have not added any of these services, you will see an **Add** button instead of **Configure**.
4. Select **Browser Synthetic (Multistep)** check from the metric list.
   Under **Check Source** > **Selenium Source**, enter the source code that you exported from Katalon.



5.
6. Optionally, configure the metric's settings. See Configure incident alert and thresholds for more information.

Test your browser synthetic check

To test your browser synthetic check, click **Test Check**.



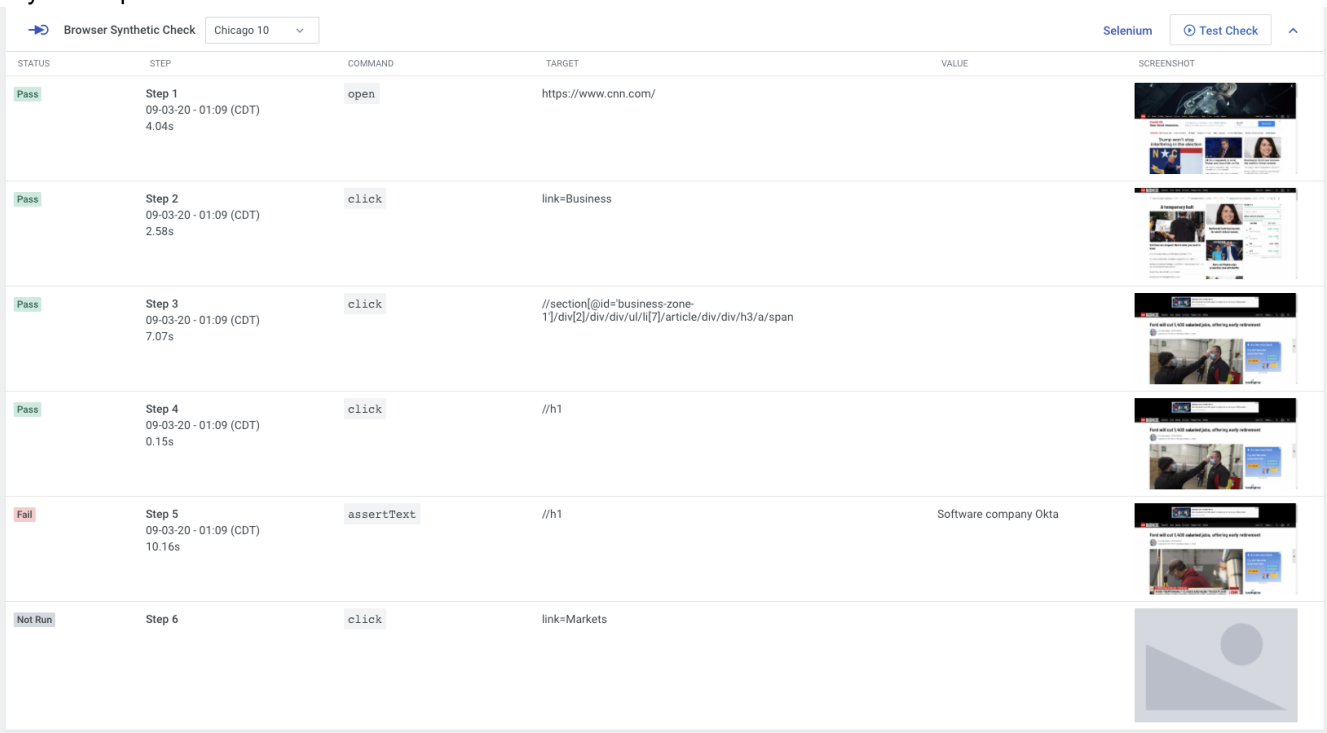
The following details are displayed in the metric's drawer:

- The status of each executed step
- The command and the target of each step
- The elapsed time to execute each step
- Screenshots of each step

The test output can also help you identify any bottlenecks that are affecting your web application's page load times, allowing you to make adjustments to improve its performance.

Incidents

Once you save a browser synthetic check and it starts running, an incident will be created in the Incident Hub if the check fails. Open the incident's details page then expand the **Browser Synthetic Check** section to show the test results and any failure point.



If you want to re-test the check, you can perform a real-time test by clicking **Test Check**.

Available multistep commands

Our multistep engine is continuously updated with new features and improvements. Many of these updates add new commands to bring our engine translation library closer to the full set of Selenium's commands. Our currently supported commands are listed below:

- addSelection
- altKeyDown
- altKeyUp
- assertAlert
- assertAlertNotPresent

- assertAlertPresent
- assertAllButtons
- assertAllFields
- assertAllLinks
- assertAllWindowIds
- assertAllWindowNames
- assertAllWindowTitles
- assertCookieNotPresent
- assertCookiePresent
- assertCssCount
- assertElementNotPresent
- assertElementPresent
- assertLocation
- assertNotCssCount
- assertText
- assertTextNotPresent
- assertTextPresent
- assertTitle
- check
- click
- clickAndWait
- close
- controlKeyDown
- controlKeyUp
- createCookie
- deleteAllVisibleCookies
- deleteCookie
- deselectPopUp
- doubleClick
- focus
- goBack
- hover
- open
- pause
- refresh
- removeScript
- runScript
- select
- selectFrame
- selectPopUp
- selectWindow
- sendKeys
- storeAlert

- storeAlertPresent
- storeAllButtons
- storeAllFields
- storeBodyText
- storeLocation
- submit
- timeout
- type
- typeKeys
- uncheck
- verifyText
- waitForElement
- waitForElementPresent
- waitForPageToLoad
- waitForText
- waitForTextNotPresent
- waitForTextPresent
- windowMaximize

JavaScript synthetic

JavaScript synthetic checks allow you to execute JavaScript scripts from our network of monitoring probes. Use JS synthetic checks for complex, multi-step actions that our Browser synthetics checks, which use Selenium, are unable to perform. For example:

- Branching, looping, or more advanced control flow
- Inspecting the response content in detail
- Testing API endpoints
- Creating sessions or generating tokens via API calls
- Testing against non-HTTP/s endpoints

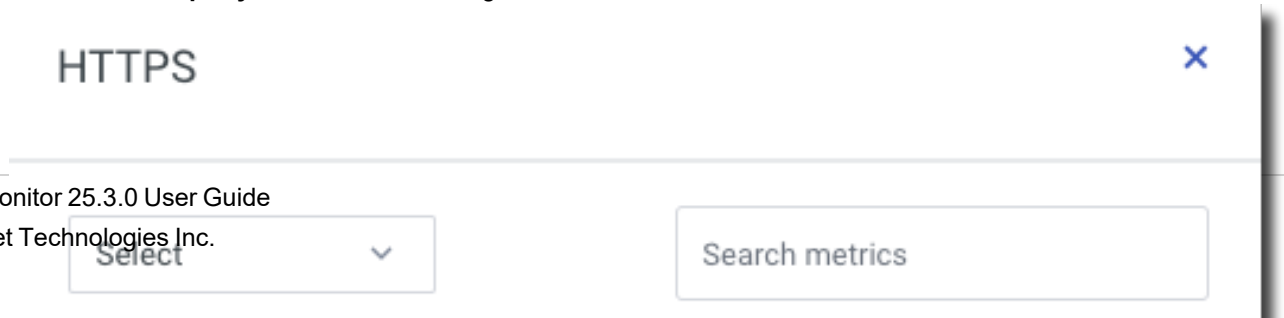
JavaScript synthetic checks can be performed using our public monitoring nodes. To perform checks against your private infrastructure, utilize the latest version of the OnSight vCollector.

JavaScript synthetic checks operate in a restricted JavaScript sandbox environment with support for a limited set of JS libraries. For more information, please contact our support team.

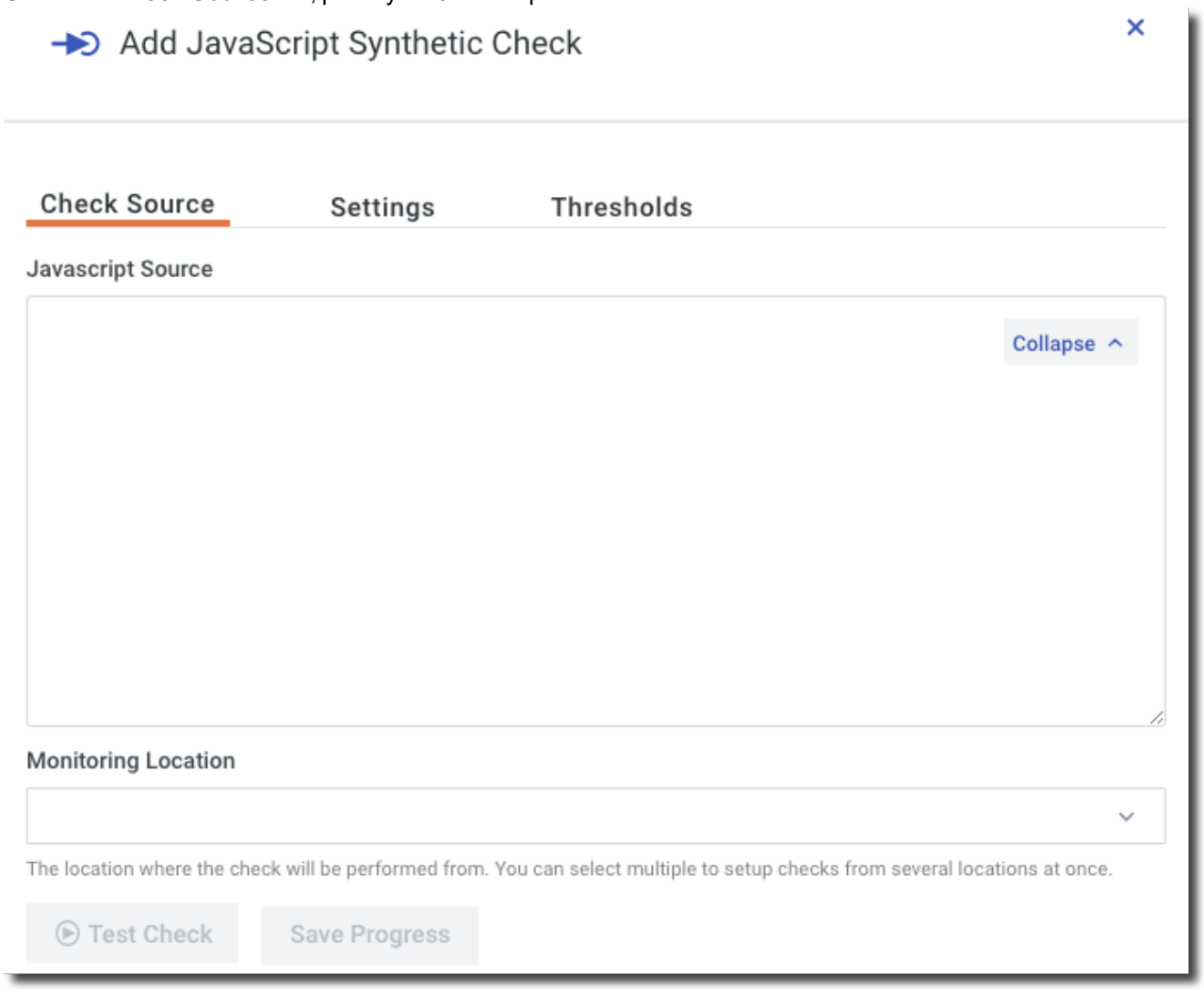
This article provides information on how to set up an environment for JavaScript synthetic checks. The article also includes a JavaScript test template, a sample test, and more information and resources to help you create your own JavaScript synthetic checks.

To create a JavaScript synthetic check:

Select **JavaScript Synthetic** when creating an HTTP or HTTPS metric.



Under the **Check Source** tab, paste your JavaScript test code.



2.
3. Optionally, configure the rest of the metric's settings. See Configure incident alert and thresholds for more information.
4. Select **Save**.

JavaScript checks will timeout if an individual step takes more than 30 seconds, or if the whole test exceeds 2 minutes.

Set up a JavaScript synthetic check environment

This article describes how to set up an environment for JavaScript synthetic checks in FortiMonitor.

Prerequisites

1. Install **Node.js** and **Node Package Manager** (**npm**).
2. Enter the following command to install the Selenium WebDriver package:

```
npm install selenium-webdriver
```

3. Install the Chrome browser driver for development and testing. Note that the browser driver must be the same version as the browser you are using (Chrome browser 80 must be paired with Chrome driver 80). The driver must be located somewhere in your system path so that it can be executed by the WebDriver.

4. Download the FortiMonitor synthetics module and save it to your local directory.

Test template

You can use the following template to get started:

```
const { init_panopta_synthetics } = require('./panopta-synthetics');

(async function example() {
    let driver = await init_panopta_synthetics('chrome');
    try {
        driver.ps_step('Opening initial page');

        await driver.get('https://google.com');
        const result = await driver.findElement(By.name('q'));

        await result.sendKeys('webdriver', Key.RETURN);
        driver.ps_pass_test();
    } catch (err) {
        driver.ps_fail_test(err);
    } finally {
        await driver.quit();
    }
})();
```

Use this template to run tests in Chrome. You can also use **no-browser** for tests that do not require a browser to perform tests.

In addition to the normal Selenium WebDriver function, the following additional functions are available in the driver object:

Function	Description
ps_step(name)	Marks the beginning of a step in the test. The name of the step will be displayed in the final output as well as a description of the overall test failure if it fails at this step. This function also begins the timer for measuring the execution time of the test step.
ps_annotate(message)	Adds additional logging information to the test step.
ps_pass_test()	Called to indicate that the test has successfully completed.
ps_fail_test(err)	Called to indicate that the test has failed. Takes an optional JavaScript exception object which, if provided, is used to build the description of why the test failed.

When running a browser-based test, additional functions expose more detailed browser timing metrics for the most recent request:

Method	Description
ps_get_dns_time()	The time needed to resolve the hostname of the request.
ps_get_connect_time()	The time needed to make the initial socket connection.
ps_get_connect_time_ssl()	The time needed to perform the SSL/TLS connection.
ps_get_request_time()	Total time to make the HTTP request.
ps_get_response_time()	Total time needed to download the HTTP response.
ps_get_ttfb()	Time from the initial request until the first byte of the response was returned.
ps_get_total_time()	Total time to process the most recent request.

Full test example

The following is an example JavaScript synthetic check code snippet. This simple test runs a Google query.

```
const { By, Key, until } = require('selenium-webdriver');
const { init_panopta_synthetics } = require('./panopta-synthetics');

(async function example() {
    let driver = await init_panopta_synthetics('chrome');
    try {
        driver.ps_step('Opening initial page');

        await driver.get('https://google.com');
        const result = await driver.findElement(By.name('q'));

        driver.ps_annotate('Running sample Google query');
        await result.sendKeys('panopta', Key.RETURN);

        driver.ps_pass_test();
    } catch (err) {
        driver.ps_fail_test(err);
    } finally {
        await driver.quit();
    }
})();
```

Test execution

Tests can be run using **node test.js --local**. This will start the browser and execute the test, displaying information in the terminal about each step that is performed. Upon completion, a JSON block that the FortiMonitor monitoring probe will interpret in production use will be displayed. If you would like to turn this off, add **--hide-results** to the end of the command.

Additional node modules

The test scripts are run from FortiMonitor's monitoring probes in a limited sandbox environment for security reasons. The following modules are supported:

- https://www.npmjs.com/package/ftp
- https://www.npmjs.com/package/axios
- https://www.npmjs.com/package/adm-zip

Additional modules can be for tests, but they need to be whitelisted in our environment in order to be imported. Please contact our support team to discuss any modules that your tests require.

Related documentation

- Selenium WebDriver - The basic documentation for the JavaScript version of the Selenium web driver covers the basic usage and configuration support for different browsers.
- Full Javascript API documentation - The most important components are:
  - WebDriver - the main object that represents the driver that Selenium is managing
  - WebElement - used for accessing information about specific DOM elements
  - By - the mechanism for locating elements on the page via CSS, XPath, element name, id, etc.
- This article describes various strategies on the best way to write repeatable JavaScript tests.

Configure incident alert and response time thresholds

Incident alerts and response time thresholds can be configured while adding or editing a network-based metric. This allows you to control when to get alerted, the Alert Timeline to use, and the severity to be assigned when an incident occurs.

For example:

1. You can configure a metric to create an incident when an **DNS port** check is unsuccessful for more than **5 minutes**.
2. Assign a severity of **Critical** to the incident.
3. Select the **Alert Timeline** to notify when the incident occurs.

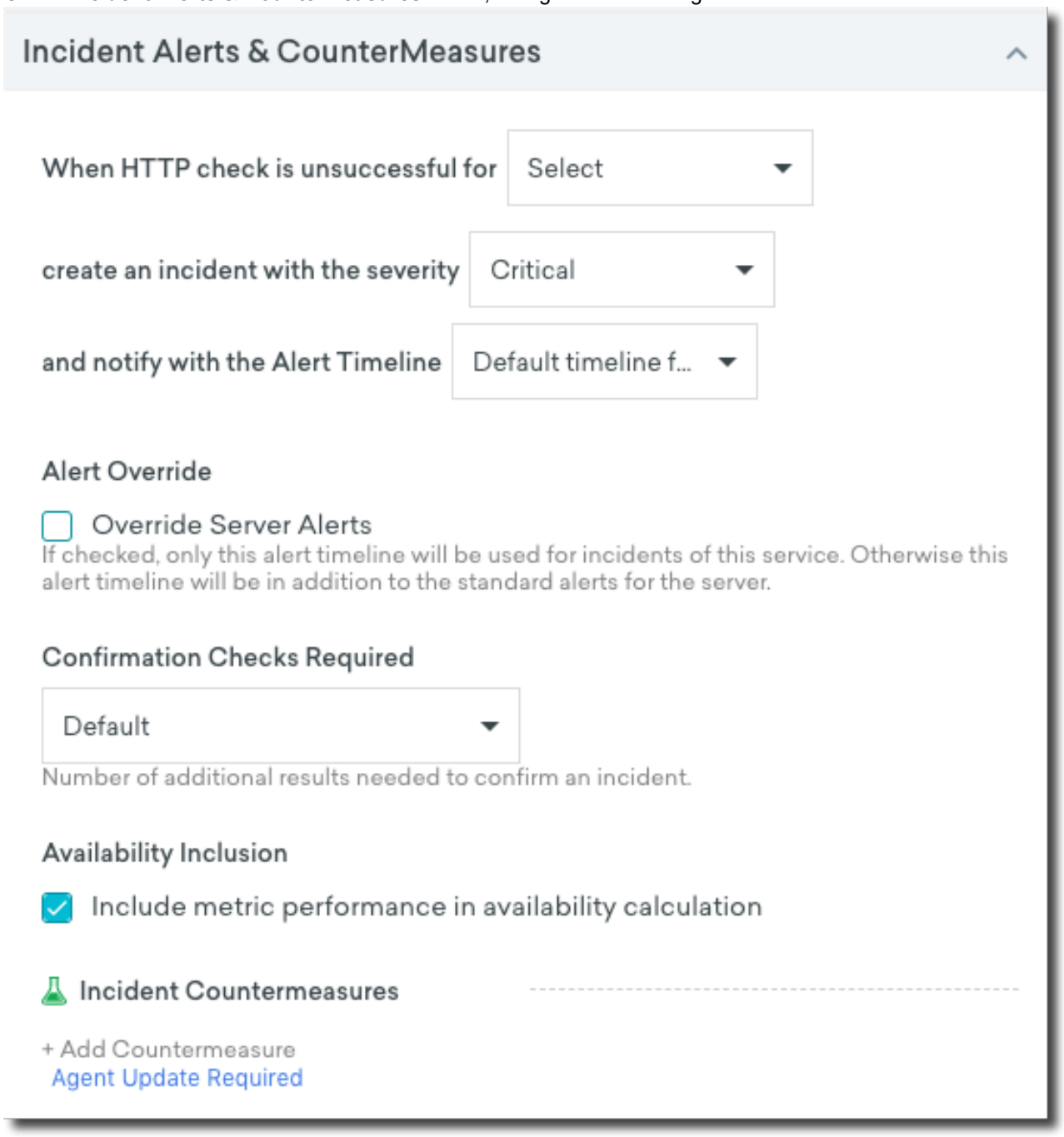
To configure alert thresholds for network service checks, see the following sections:

- Incident alerts thresholds
- Response time thresholds

Incident alerts thresholds

To configure incident alert thresholds, perform the following steps:

1. On the **Instance Details** page, choose a network-based metric to configure.
2. Select the **More** icon, then select **Edit.**
   On the **Incident Alerts & CounterMeasures** modal, configure the following fields:



3.

Field	Description

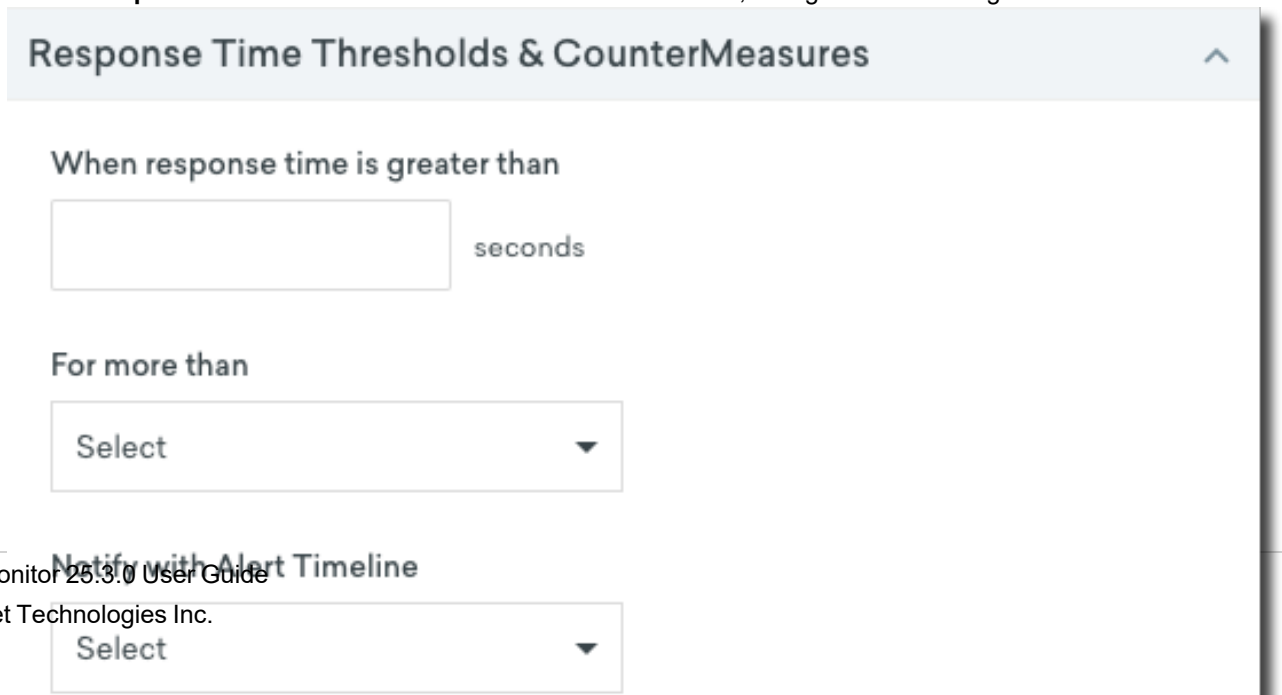
When HTTPS check is unsuccessful for	Select the period of time or (threshold period) before an incident is created after the first unsuccessful network check. **Values**: • 1 minute • 2 minutes • 5 minutes • 10 minutes • 15 minutes
create an incident with the severity	Select the severity to assign to the created incident. **Values**: • Warning • Critical
and notify with the Alert Timeline	Select which **Alert Timeline** to use when the configured threshold period is breached.
Override Server Alerts	If selected, only the specified Alert Timeline will be if an incident occurs for this service. Otherwise, the selected Alert Timeline will be used in addition to the standard alerts for the server.
Availability Inclusion	Select if you want to include this alert to your server's availability calculation.
Incident CounterMeasures	Select **+ Add CounterMeasure** to create a CounterMeasure to run when this threshold is breached. See CounterMeasures Overview. **Note**: The Agent must be updated to add a CounterMeasure.

4. To configure **Response Time Thresholds**, see the following section. Otherwise, select **Save**.

Response time thresholds

To configure incident alert thresholds, perform the following steps:

1. On the **Instance Details** page, choose a network-based metric to configure.
2. Select the **More** icon, then select **Edit.**
   On the **Response Time Thresholds & CounterMeasures** modal, configure the following fields:



Field	Description
When response time is greater than	If the selected value is greater than the response time for a specified amount of time, an incident is created. **Values**: • 1 minute • 2 minutes • 5 minutes • 10 minutes • 15 minutes
For more than	If the response time is greater than the value specified in the **When response time is greater than** field for more than the selected value, an incident is created. **Values**: • 1 minute • 2 minutes • 5 minutes • 10 minutes • 15 minutes • 20 minutes • 30 minutes • 45 minutes • 1 hour • 2 hours • 4 hours • 8 hours • 12 hours • 24 hours
Notify with Alert Timeline	Select which Alert Timeline to use when the configured response time threshold is breached.
With a severity of	Select the severity to assign to the created incident. **Values**: • Warning • Critical
Include metric performance in availability	Select if you want to include this metric's performance to your server's availability calculation.
Response time CounterMeasures	Select **+ Add CounterMeasure** to create a CounterMeasure to run when this threshold is breached. See CounterMeasures Overview. **Note**: The Agent must be updated to add a CounterMeasure.

4. Select **Save**.

Dynamic variables

If you have common patterns in your infrastructure monitoring, FortiMonitor's templating support can be a powerful option to streamline your configuration and maintain consistency.

One complication with templates is being able to capture dynamic elements that will change with each server. Within the options tab for each network service check, you can use dynamic variables to be filled in with specific details which will be dictated by the server's configuration. The variables will be replaced by the coinciding values upon application of the templates.

Standard Variables

There are standard variables relating to the server's properties which are listed below. Alternatively, you can use our server attribute system to introduce your own custom attributes. Simply use the attribute name in the macro name.

fqdn – This field would be replaced with the server's primary FQDN

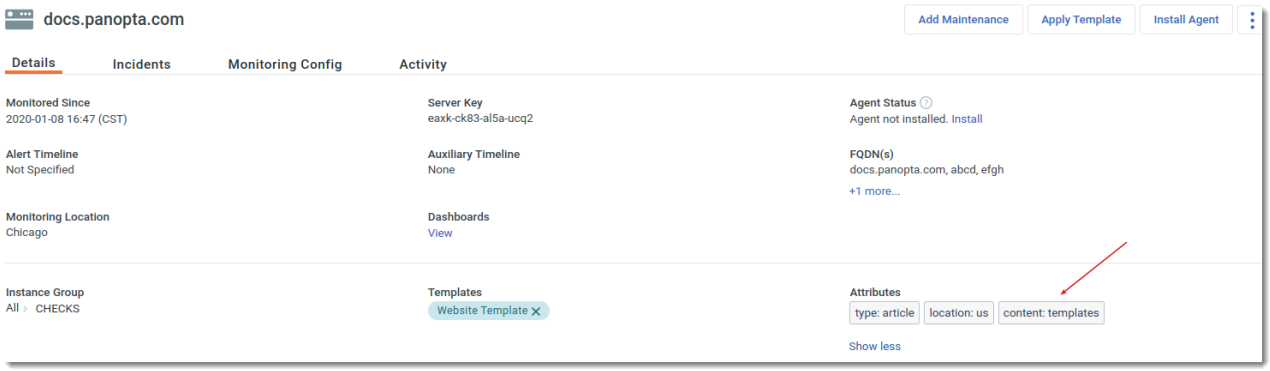
server_name – This field would be replaced with the Server Name

server_group – This field would be replaced with the Server Group that the server is located in

attribute_name – This field will be replaced with the value of the attribute referenced. So, if you had an attribute key named operating_system, and the value was Ubuntu, you could reference **operating_system** in your check options and it would be translated to Ubuntu.

Demonstration of Usage

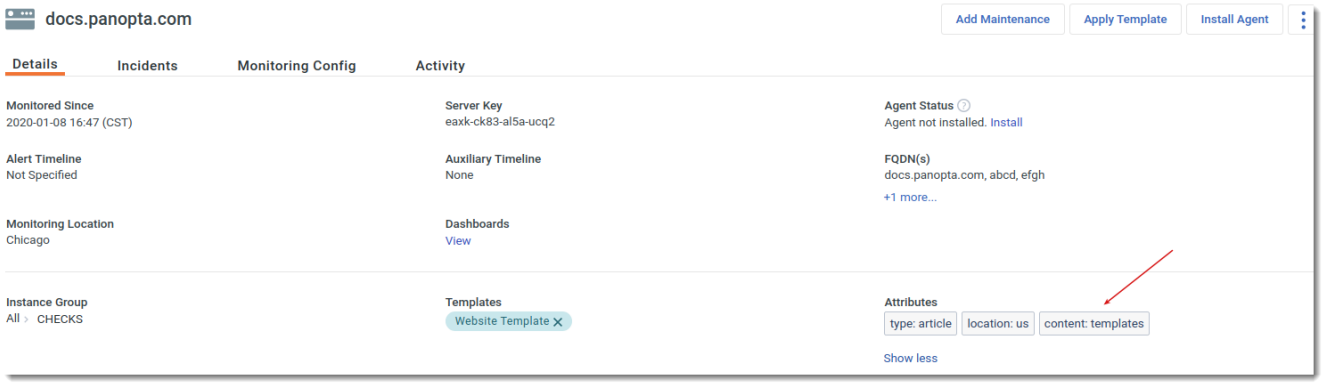
- Below you can see a screenshot of a website instance called "docs.panopta.com" within the Panopta Control panel. This instance has the following attributes: **type:article**, **location:us**, **content:templates**. We are going to use the **content:template** attribute to apply a metric.



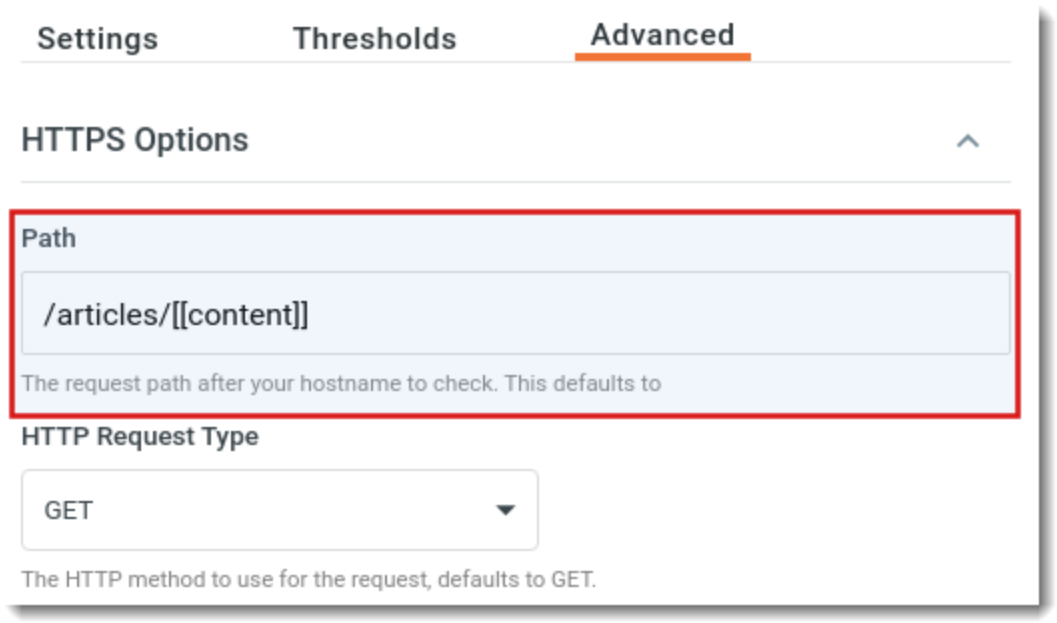
- In the following screenshot, there is a template called "Website Template". We will be applying this template to docs.panopta.com. In this template, we specified an HTTPS GET check to apply the following path for an HTTPS metric:

```
/articles/[[content]]
```

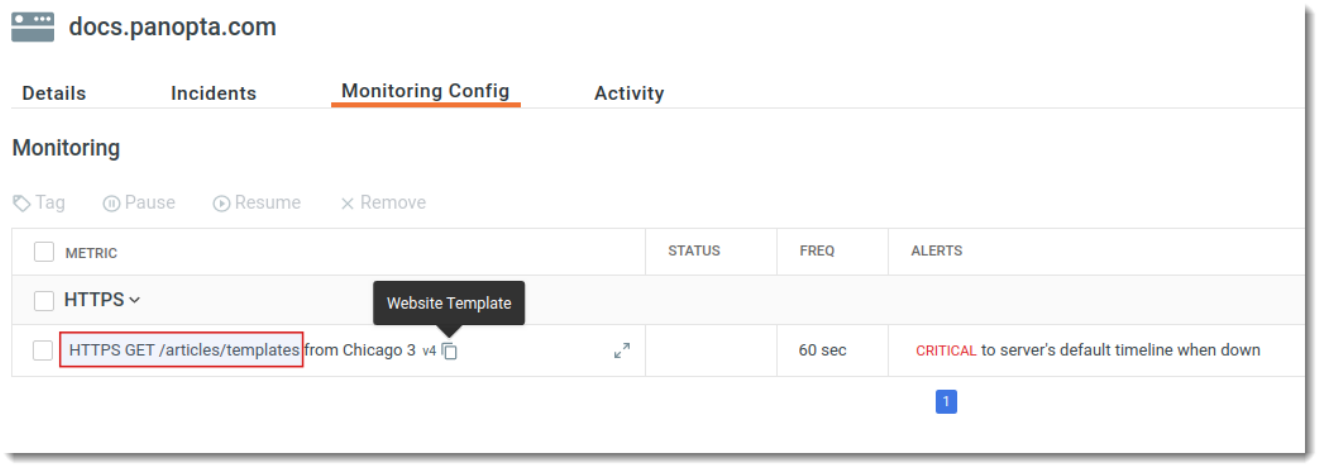
NOTE: **content** is the attribute key found on docs.panopta.com (refer to the above screenshot).



Below is a screenshot of the configuration made to set up this path. You can access this when editing your HTTPS GET metric.



After saving the template configuration, and applying it to the instance "docs.panopta.com", we can now see that there is a metric called *HTTPS GET /articles/templates from Chicago 3*. This took the **key** (*content*) from the instance, and applied the corresponding **value** (*template*) to this metric.



FortiMonitor Agent

The FortiMonitor Agent is a local monitoring utility that is deployed directly on your physical server, VM, or cloud server. It allows you to monitor your server's resources, processes, and services. The FortiMonitor Agent also supports several plugins to monitor applications like web servers, databases, caches, and so forth. It is also extensible, allowing you to customize it to meet your own needs.

The FortiMonitor Agent can be used to monitor applications running on your servers such as Apache, Cassandra, SQL Server, and more. The Agent is compatible with Windows, Linux, and Unix.

Benefits of using the FortiMonitor Agent

The main benefit of running the monitoring agent on servers and virtual machines is the correlation of data and root-cause detection. By checking servers and services from the collectors/probes, you're able to perform outcome detection and measure network/application latency. Running the monitoring agent allows you to correlate internal server health and application performance metrics to those outcomes. Some examples include:

- Application latency being related to increased connections on the related database node
- Database commits/checkpoints taking longer than usual due to high disk I/O on the node

This basically gives you the what and the why part of the equation which is important for proactive monitoring and root cause resolution.

Linux

The following Linux operating systems are supported:

- Red Hat Enterprise Linux (RHEL) 7.0 or later
- Ubuntu 18.04 or later
- Debian Linux 9.0 or later
- Amazon Linux
- AlmaLinux 8.x to 9.x
- Rocky Linux 8 and 9
- AIX
- FreeBSD/OSX
- Solaris

Note: The Agent only supports operating systems that are actively under standard support from their corresponding vendors.

Windows

The following Windows operating systems are supported:

- Microsoft Windows Server 2025
- Microsoft Windows Server 2022
- Microsoft Windows Server 2019
- Microsoft Windows Server 2016
- Microsoft Windows Server 2012 R2
- Microsoft Windows 10
- Microsoft Windows 11

Notes: The Agent only supports operating systems that are actively under standard support from their corresponding vendors.

The Windows Agent also supports auto-update which allows the Agent to periodically check and install new versions for itself. For more information, see Windows Agent Auto-update.

Install the FortiMonitor Agent

Install an Agent on a new server

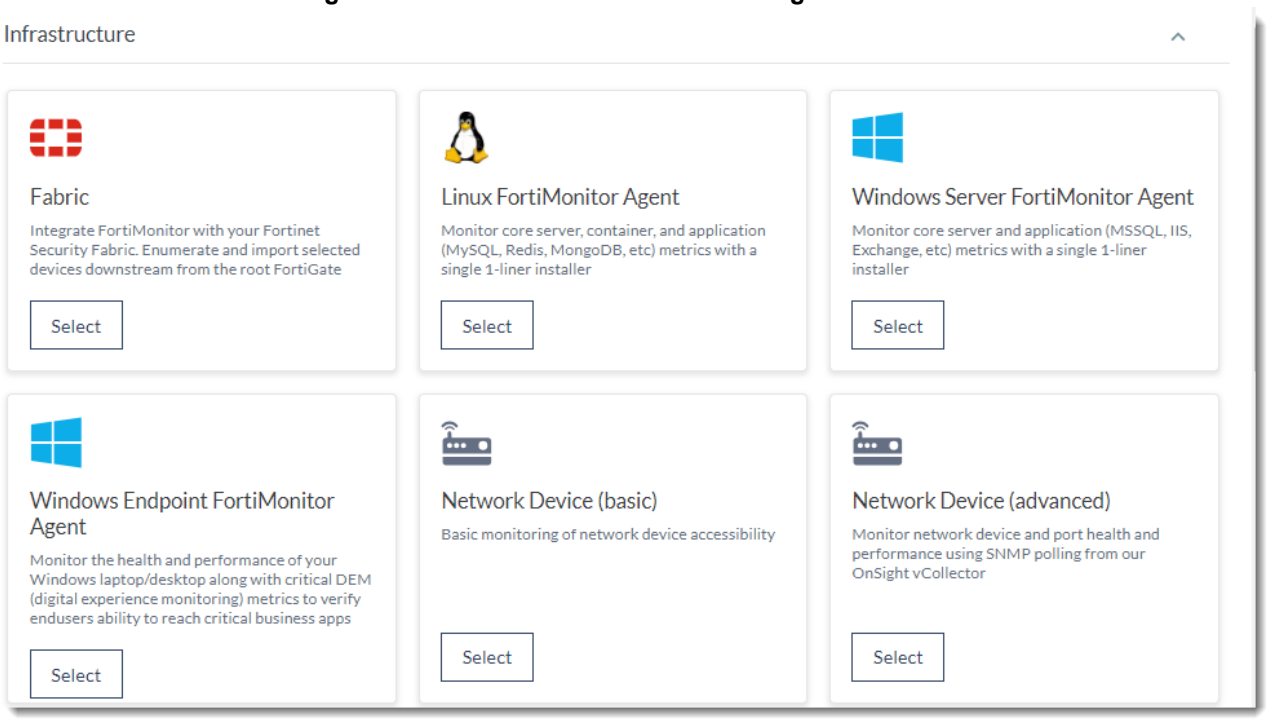
You can install the Agent on a server and it will automatically be added to your FortiMonitor account. This is a helpful option for installing the Agent on a number of servers at once to import into FortiMonitor.

Note: Python 3 is required for all Linux Agent installations. For Windows Agent installations, you need .NET 4.5 or higher. TLS 1.2 and 1.3. are supported on DotNet versions 4.5 or higher.

To install the Agent to a Windows or Linux server, perform the following steps:

1. From the navigation menu, click **Add**.

Select **Linux FortiMonitor Agent** or **Windows Server FortiMonitor Agent**.



2.
3. Select either **Quick Install Instructions** or **Manual Install Instructions**.
   Selecting Quick Install Instructions will generate a command that you can run manually on each of your servers or through your configuration management tool, like Chef or Puppet.
4. Follow the on-screen instructions.

Script reusability

This script is generic for your account - you can use it on any new server you'd like to add to FortiMonitor and it will obtain its own distinct key during the installation process.

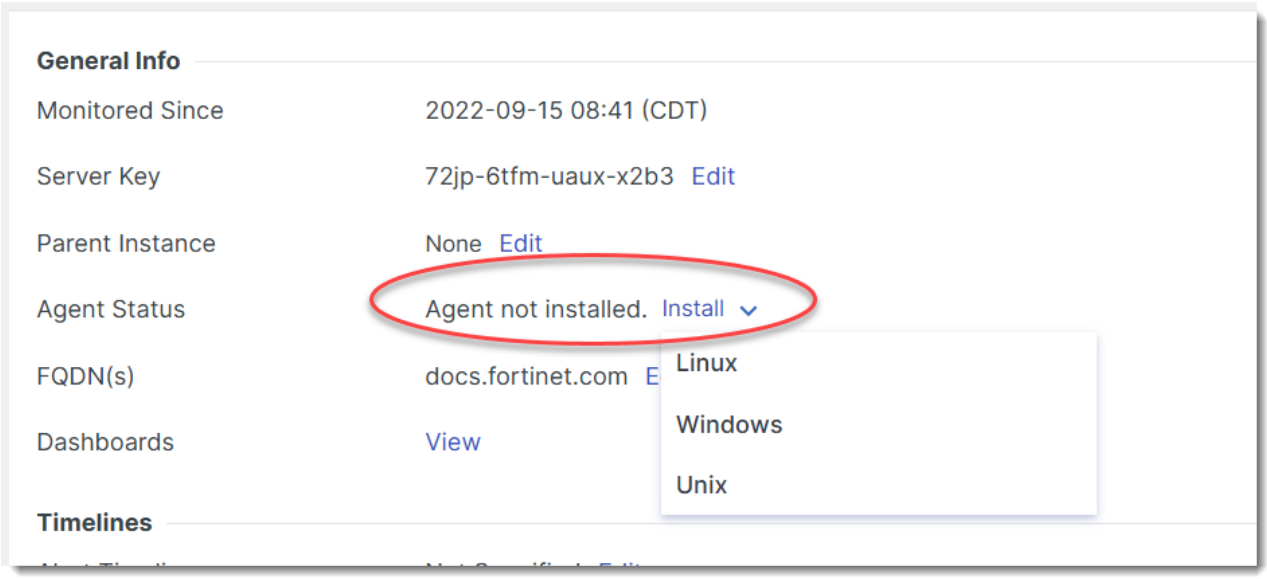
Python

Python is required on all Linux Agent installations for the script to work properly.

This command will install the Agent, obtain a server key, add the server to FortiMonitor, and begin reporting metrics.

Install the Agent on a server already in FortiMonitor

1. On the **Instance Details page**, go to the **Details** tab.
2. Select **Install** under the **General Info** section.



3. Clicking **Install Agent** will give you a few prompts. The first being to select the OS of the server on which you are installing the Agent.
4. After selecting the OS, you have the option to either do a **Quick Install** or a **Manual Install**. The **Quick Install** tab provides a quick script for you to run which will download the Agent from our repository and link the Agent to your account. It will start reporting metrics in minutes, which you will be able to see in the control panel. Selecting the **Manual Install** tab will provide several commands that you must run individually to download and install the Agent to your server.



This version of the script has a unique key for this one instance only. Do not use the same script to try to install other Agents. A key is generated for each server.

Installer parameters

FortiMonitor Linux Agent

The following parameters can be appended to the standalone installer command if you selected Linux and want to install the **fm-agent**:

Parameter	Argument Type	Description
-c	Customer key - Found in the Control Panel under My Account	**Required** --customer_key To obtain your customer key, select your avatar then select **My Account**.

-s	Server key - Unique server key	**Optional** --server-key Server key to use if one has already been assigned. Typically not needed except for a re-installation.
-a	URL	**Optional** --aggregator-url If you run a proxy or use a different aggregator. Not typically needed.
-g	Int	**Optional** --server_group Assign the instance to one or more instance groups.
-f	URL	**Optional** --manifest-url If you host a manifest file on your installation network, typically used when automating large installations.
-t	Int	**Optional** --templates Applies the defined template or templates to the instance.
-l	Comma-separated value, string	**Optional** --tags Applies the defined tags to the instance
-d	Takes no argument	**Optional** --disable-server-match Disable server matching. This will prevent instances from getting merged when using the Agent one-liner installer for multiple servers.
-x	Takes no argument	**Optional** --migrate Used to migrate your configuration from the panopta-agent. For more information, see Migrating a Panopta Agent configuration to FortiMonitor Agent.
-y	Takes no argument	**Optional** --skip Used when automating migration.
--dem	Takes no argument	**Optional** --dem Used to enable Digital Experience Monitoring (DEM). The following entry is added to the manifest file: `[dem]`

```
enabled = true
```

For example, enter the following command:

```
curl -s https://repo.fortimonitor.com/install/fm_agent_install.sh | bash /dev/stdin -c
     <customer-key> -l linux,database,prod -g 23242 -t 2121,32323
```

The parameters and values will be inserted into the manifest file and subsequently be applied to the instance.

```
[agent]
customer_key = 13232-32342-2323
tags = linux,database,prod
server_group = 23242
templates = 2121,32323
```

FortiMonitor Windows Agent

The following parameters can be appended to the standalone installer command if you selected the MSI installer option:

Parameter	Type	Description
customer_key		**Required** To obtain your customer key, select your avatar then select **My Account**.
server_group	Int	Assign the instance to one or more instance groups.
templates	Int	Applies the defined template or templates to the instance.
tags	Comma-separated value, string	Applies the defined tags to the instance
disable-server-match		Disable server matching. This will prevent instances from getting merged when using the Agent one-liner installer for multiple servers. A `true|false` argument is needed after the `–disable_server_match` flag. For example: `Invoke-WebRequest https://repo.fortimonitor.com/install/win/fortimonitor_agent_windows.ps1 -OutFile fortimonitor_agent_windows.ps1; .\fortimonitor_agent_windows.ps1 -customer_key xxxx-xxxx-xxxx-xxxx -countermeasures -disable_server_match true`
autoupdate		Enables the Windows Agent auto-update feature.
scheduled_ update HH:MM		Set a specific time for the Agent to update. This should be in military-time format; for example, 2:09 PM would be set as 14:09. The agent will query/attempt an update at this time. If this flag is not used, the Agent will select a random time within 24 hours of its startup to check for a download; this should be logged in the Agent log.
force_thread_ culture		Set the current thread culture to English.

For example, enter the following command:

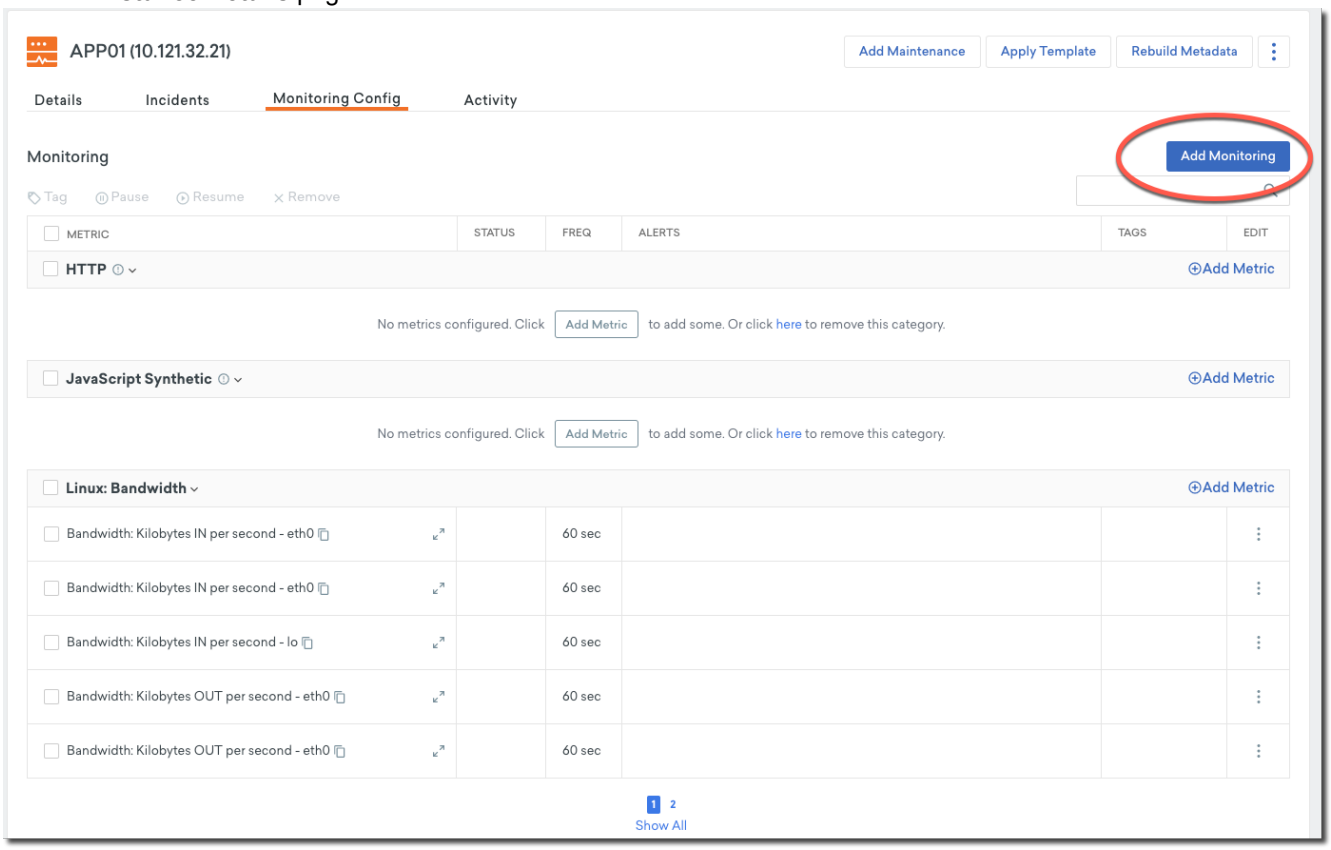
```
Invoke-WebRequest https://repo.fortimonitor.com/install/win/fortimonitor_agent_
windows.ps1 -OutFile fortimonitor_agent_windows.ps1; .\fortimonitor_agent_windows.ps1 -
customer_key xxxx-xxxx-xxxx-xxxx-server_key aaaa-bbbb-xxxx-yyyy -tags
windows,database,prod -server_group 23222 -templates 2121,32323 -countermeasures
```

The parameters and values will be inserted into the manifest file and subsequently be applied to the instance.

```
[agent]
customer_key = 1122-1212-1111-2222
tags = windows,database,prod
server_group = 23222
templates = 2121,32323
```

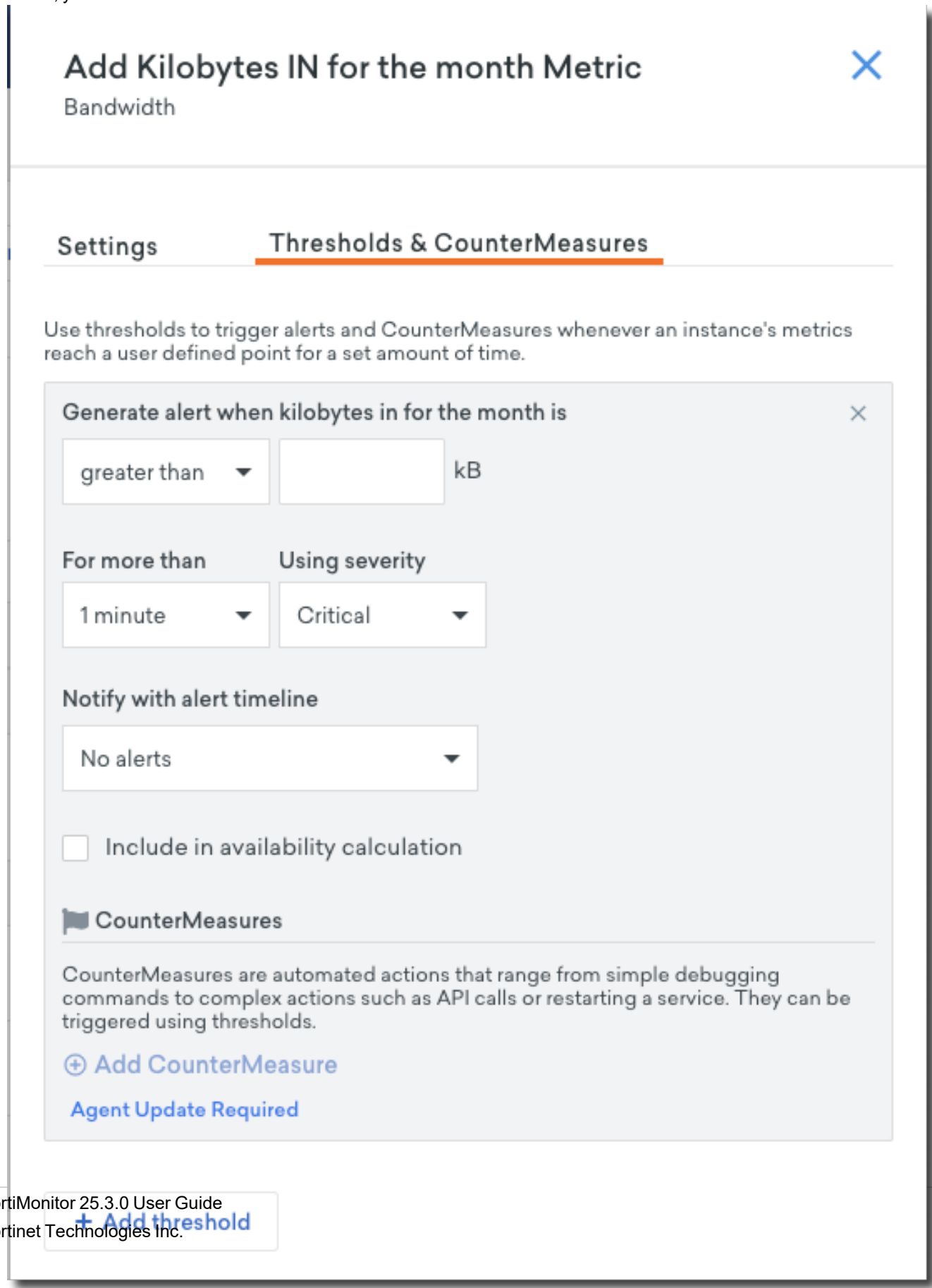
Add a monitoring agent resource

To add a monitoring Agent resource to your server, select the Add Monitoring from the **Monitoring Config** tab of your server's **Instance Details** page.



Each plugin may have its own configuration options which are required. Those options are highlighted in each plugins page which you can find here.

In the configuration drawer of each plugin, you should see a **Thresholds & CounterMeasures** tab. By clicking on this tab, you can configure notifications for this resource however you like. If you would like multiple thresholds on a single check, you can do so with the **+ Add Threshold** button.



It is not unusual to choose not to be alerted for an Agent resource, but it is important to know how such alerts work if you decide you would like them. Every agent resource returns a number and the alert will be based on that number. You can set the alert threshold such that if that number is greater than, less than, equal, or not equal to a number you specify for an amount of time you also specify, then that will result in an alert. Since every resource returns their own numbers it is important to understand which resource you are collecting and what kinds of numbers you should expect. Whenever you have the resource configured as you want, click **Add Metric**. After the next time the agent syncs with our servers (which happens every minute) this check will be up and running. Each of these resources is described in detail in Application plugins.

Install the FortiMonitor Agent without root access

Important Security Note

We highly recommend creating a new user for the FortiMonitor Agent so that the permissions are scoped to only require what is needed for the Agent, rather than a user which has some sort of elevated permissions. Please consult with FortiMonitor Support for best practices on this.

To install the FortiMonitor Agent without root access

User Creation (to be done as root)

```
adduser -m -s /bin/bash fm-agent;
passwd fm-agent

# redhat/centos only;
yum install python3-devel gcc;
# debian/ubuntu only;
apt-get install python3-devel gcc;

pip3 install virtualenv
```

Allow Cron

On some systems, you may have an **/etc/cron.allow** file that whitelists which users can run crontab. If this exists on your system, add the new **fm-agent** user to that file. If it does not exist, skip this step.

Agent Installation

Login to the server as user fm-agent and then run the following:

```
wget https://storage.googleapis.com/agent-repo/tar-stable/fm-agent-latest.tar.gz && tar -
     xvzf fm-agent-latest.tar.gz
cd ~
mkdir etc
vim etc/fm-agent-manifest
```

Within that text editor session, add the below line and replace with your own customer key .

```
Copy;
customer_key = xxxx-xxxx-xxxx-xxxx;
```

If needed, you can specify any of the other agent install options that are supported in our Agent Manifest File. Then run the following:

```
cd ~/fm-agent-2022.22.3/ #replace version number;
python install.py --no-root
```

This should install the agent and register the server within your control panel account within the incoming directory.

Install the FortiMonitor Agent on a Docker host

Docker containers can be monitored similar to any existing physical server or VM. By installing the FortiMonitor Agent on your Docker host, each container can be monitored and represented in FortiMonitor individually. Applications within containers can also be monitored with a single host agent, assuming they expose their metrics in an accessible way. You may also run the FortiMonitor Agent in containers to obtain metrics not exposed outside of the container.

Installation

To install the FortiMonitor Agent on your Docker host, see Install the FortiMonitor Agent. Once the Agent is installed, run the following command to add the user the agent runs as to the Docker user group. This enables the FortiMonitor Agent to connect to the Docker socket and obtain Docker metrics and metadata.

```
sudo usermod -a -G docker fm-agent
```

If you run the Agent as a different user, add that user to the docker group.

The FortiMonitor Agent will begin discovering containers on the Docker host and adding them to FortiMonitor.

Application Monitoring

Currently, the following set of applications running in Docker containers are supported:

- Nginx
- Mongodb
- Apache
- MySQL
- Redis
- RabbitMQ

To monitor these applications, add a mapping block to your config file that lets FortiMonitor know which application is using which image. This allows you to specify if an application is using a custom image you have built.

Here's an example with each of the above mentioned applications. The left side is the FortiMonitor name for the application; the right side is the name of your container image.

```
[agent]
customer_key = abcde-1232-ploc03-xcxvp
enable_docker = true

[docker_image_mapping]
mongo = mongo
rabbitmq = rabbitmq
mysql = mysql
redis = redis
nginx = nginx
apache = httpd
```

You will still need to perform the required configuration for each application (if needed). See Application plugins for more information.

Control Panel

Instance Tree

If you open the instance tree on the left side of the control panel, you will see that your containers are nested under each Docker host. Clicking any of the containers will take you to its details page.

Infrastructure Map and Dashboards

Containers will appear in the **Infrastructure Maps** and **Dashboards** similar to any other monitored instance.

Update the FortiMonitor Agent

To update the FortiMonitor Agent, select your server's OS from the list below then run the command.

Redhat or CentOS

Run the following command:

```
yum install fm-agent
```

Debian or Ubuntu

Run the following commands:

1. Get the key by running the following command:
   `wget -O - https://repo.fortimonitor.com/fortimonitor.pub | sudo apt-key add -`
2. Then run the following:
   `apt-get update && apt-get install fm-agent`

Linux or Unix tar installer

Run the following command:

```
curl -s https://repo.fortimonitor.com/install/fm_agent_install.sh | bash /dev/stdin --
     customer-key CUSTOMER_KEY_HERE
```

To obtain your customer key, select your avatar then select **My Account**.

Windows

If you originally installed the FortiMonitor Agent using the one-liner quick install, perform the following steps:

1. Retrieve the quick install command from within your FortiMonitor account by following the steps to Install the FortiMonitor Agent.

2. Run the command in an administrative PowerShell window.

If you originally used MSI to install the agent, download and install the latest MSI.

Additionally, you can also enable Windows Agent auto-update.

MacOS

Perform the following steps:

1. Open a terminal window then run the following command. Make sure you have administrator privileges (sudo).
   ```
   curl -# https://repo.fortimonitor.com/darwin-stable/fm_agent_install.sh | sudo bash
   /dev/stdin -c <CUSTOMER_KEY> -e -d -u
   ```
   Where:
   <CUSTOMER_KEY> is your customer key.
   -e sets the digital experience section in the manifest file.
   -d sets the disable server match key in the manifest file
   -u sets the auto-update flag in the manifest file.
2. Wait for the installation to complete.
3. Go back to FortiMonitor then click **Finish Installation**.

Uninstalling the FortiMonitor Agent

Removing the FortiMonitor Agent from your server should use the same mechanism which was used to install but with the platform-specific differences. Each platform is covered below.

Redhat/Centos

To remove the **fm-agent**, run the following command:

```
yum remove fm-agent
```

Debian/Ubuntu

To remove the **fm-agent**, run the following command:

```
apt remove fm-agent && apt purge fm-agent
```

Linux/Unix Tar Installer

To remove the **fm-agent**, run the following command:

1. Download the uninstall script by running:
   ```
   curl -O https://repo.fortimonitor.com/install/linux/uninstall_agent_zip.py
   ```
2. Run the following script as root:
   ```
   python3 uninstall_agent_zip.py
   ```

Windows

If you originally used the MSI to install the Agent:

Go to **Add Remove Programs** and uninstall the agent

If you originally used PowerShell to install the Agent, use WMI through PowerShell to uninstall it.

To remove the **Windows FortiMonitor Agent**:

```
$product = get-wmiobject Win32_product | where-object {$_.Name -eq'FortiMonitor Agent' }
$product.uninstall()
```

MacOS Endpoint Agent

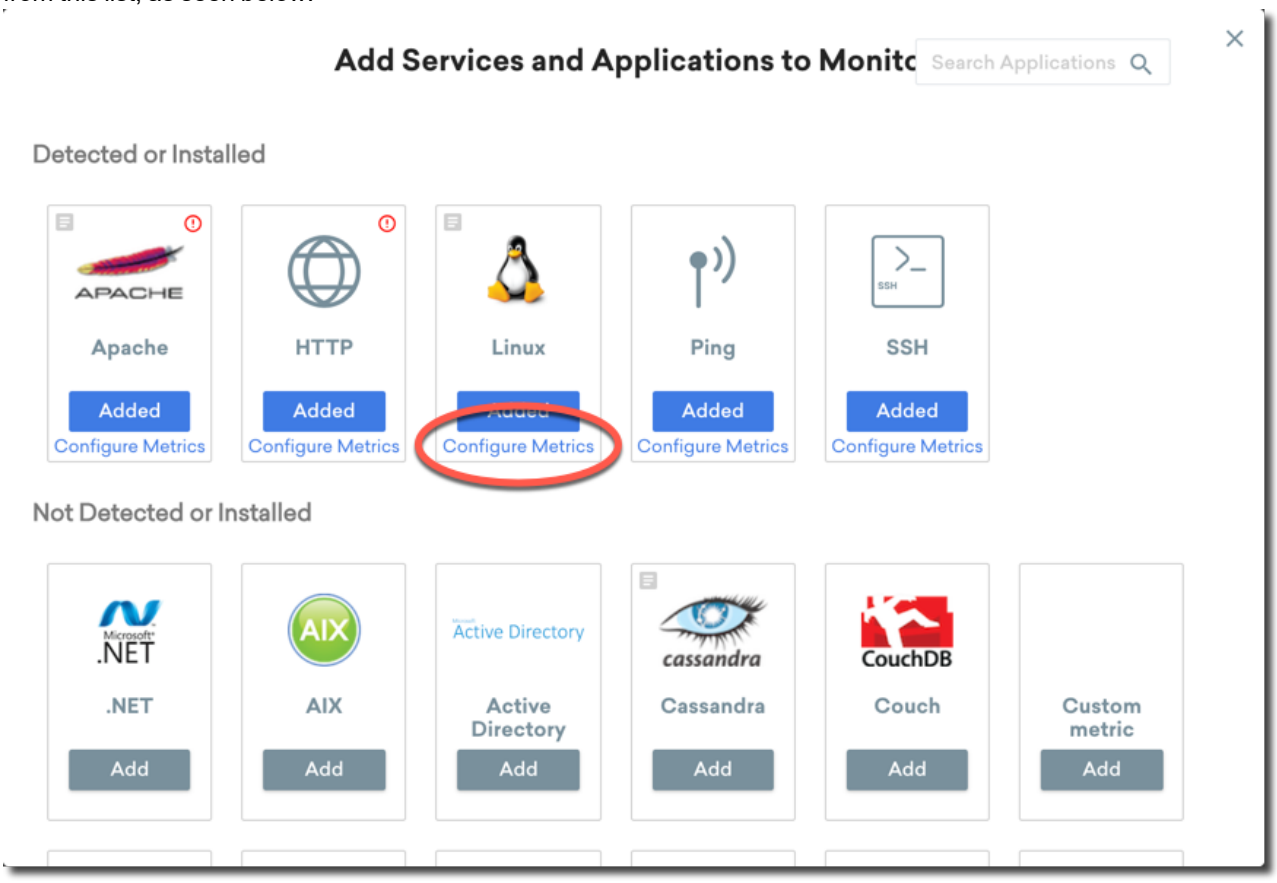
To uninstall the agent, open a terminal window then run the following command:

```
sudo bash /usr/local/fortimonitor/uninstall.sh
```

Add FortiMonitor Agent checks

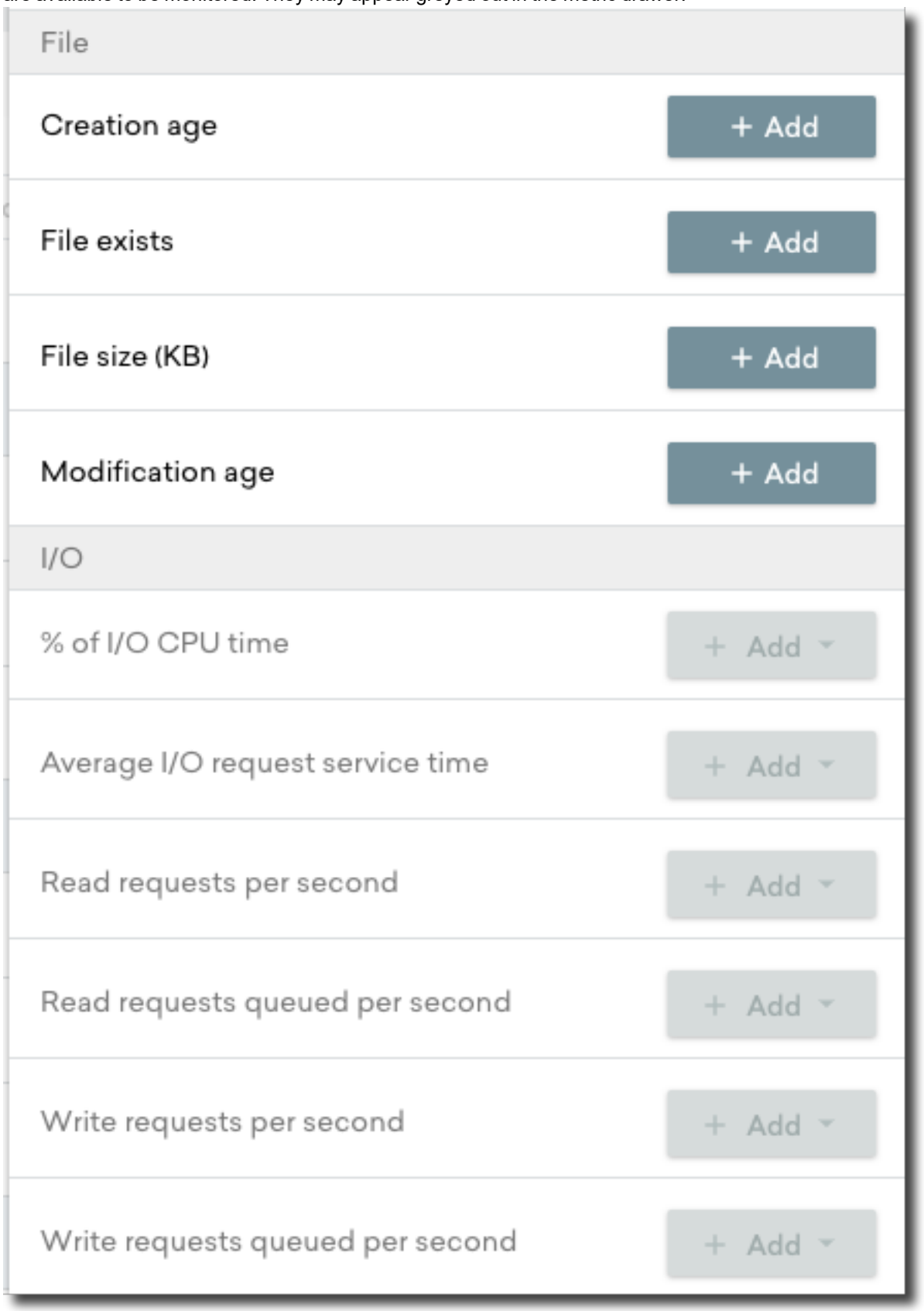
To add a monitoring Agent resource to your server:

Go the **Monitoring Config** tab of the **Instance Details** page and select **Add Monitoring**.
A catalog with all available services and applications will be displayed. You can select any category you require from this list, as seen below:



1.
2. Select a plugin then click **Configure Metrics.** A drawer will slide out from the right to show you the metric options available for the plugin.
   At this point, it is important to point out that some Linux agent resources require additional configuration before they

are available to be monitored. They may appear greyed out in the metric drawer.



Each plugin has its own configuration options. Those options are discussed individually in Application plugins.

3. Choose a metric then select **Add Metric**.
   In the following step, you have the option to add a metric manually or use **Quick Add**. Selecting the metrics under **Quick Add** adds a metric with default parameters. You can edit the parameters anytime.
4. Select **+ Metric** or any metric under **Quick Add**.
5. Configure the following fields:
   - **Interface** - Select the interface of the server where you want to add the metric.
   - **Frequency** - Select how often the metric will be collected.
   - **Metric Name Override** - Enter a metric name to override the autogenerated name. If provided, this will be used in metric configuration, graph legends, and alerts. Leave blank to use auto-generated metric names.
6. Expand **Thresholds & CounterMeasures** then select **+ Add threshold**.
7. Configure the following fields:
   - **Generate alert when kilobytes in for the month is** - Set the alert threshold such that if that number is greater than, less than, equal, or not equal to a number you specify for an amount of time that you also specify, an alert will trigger.
     For more than - Specify a time.
   - **Using severity** - Select a severity (Critical, Warning).
   - **Notify with alert timeline** - Select the Alert Timeline.
     Include in availability calculation - Select if you want to include this alert to your server's availability calculation.
8. See CounterMeasures if you want to configure **CounterMeasures**.
9. (Optional) Select **Add threshold** to add multiple alerts based on severity for a single check.
10. Expand **Tags** then add the tags that you want to use. To add multiple tags, hit Enter.
11. When you have the resource configured, select **Save**. The next time the agent syncs with FortiMonitor's servers (which happens every 2-3 minutes), this check will be up and running.

Since every resource returns their own numbers, it is important to understand which resource you are collecting and what kinds of numbers you should expect. Each of these resources is described in detail on our support page and if you would like to learn more about any one of them you can see the list in Application plugins.

FortiMonitor Agent Security

Network Communication + Encryption

The FortiMonitor Agent performs all of its monitoring and reporting without opening any inbound ports from the outside. When the Agent needs to report monitoring data to FortiMonitor, an HTTPS call is initiated from the Agent to one of FortiMonitor's secure servers, the data is exchanged, and then the connection is closed. Furthermore, all traffic exchanged between the Agent and FortiMonitor is encrypted with SSL/TLS to keep your monitoring data safe.

Linux Agent: Supports TLS 1.2 and 1.3 if OpenSSL version is 1.0.1 or higher.

Windows Agent: Supports TLS 1.2 and 1.3 on DotNet versions 4.5 or higher.

The agent communicates with secure FortiMonitor endpoints which are listed below. If you are whitelisting access, please allow the below addresses to ensure proper agent function.

Port	Destination

443/tcp	rx.us01.fortimonitor.com (resolves to 34.102.162.51) aggregator2.panopta.com (resolves to 35.190.4.8) aggregator2-secondary.panopta.com (regional failover, resolves to 35.186.218.62)
80/tcp, 443/tcp	packages.panopta.com (resolves to either 162.243.0.67 or 165.227.86.161) repo.fortimonitor.com (resolves to 34.111.115.83)

Data and Metrics

The FortiMonitor Agent only collects data and metrics which are specific to the health and performance of the systems. No customer information, PII data or log files are gathered and sent to the cloud. The FortiMonitor Agent uses a plugin-based model for metric collection and each plugin uses the appropriate system command, utility, or API to pull only the data required to evaluate the necessary thresholds for alerting and event management. On Windows, the plugins leverage the standard Perfmon counters which the OS maintains to measure the various resource levels and health metrics.

Open Source

On Linux, the Agent itself consists of a set of Python scripts. Feel free to take a look at the code in **/usr/lib/fm-agent** after installing and let us know if you have any questions or concerns.

Non Administrative User

The agent is run on the server as a non-administrative user. On Linux, the installation creates a user account with limited privileges and does not have any shell access. All actions which require higher permissions need to be explicitly elevated by an admin. On Windows, the agent runs the service as the **LOCAL SYSTEM** user and is primarily leveraging PerfMon counters to facilitate the monitoring.

Network proxy configuration

In instances where an Agent is deployed in an environment that requires going through an upstream proxy to get public internet access, additional configuration may be needed to allow the Agent to sync with FortiMonitor.

If the proxy server you are using does not require credentials, there is no additional configuration needed. The Agents will automatically pick up the proxy settings from your environment.

For proxies that require credentials, you need to add the credentials to the Agent's manifest file or configuration file. To do so, perform the steps detailed below.

Windows Agent

1. Open the Agent configuration file (**C:\Program Files (x86)\FortiMonitor\Agent.config**).
2. Add an **agent_proxy** block under the **config** node, for example:

```
<agent_proxy>
```

```
<add key="http" value="http://uname:password@216.128.151.185:3128" />
</agent_proxy>
```

3. Restart the **FortiMonitor Agent** service.

Note: You can restart the service by running the PowerShell command **restart-service FortiMonitorAgent** or by going to **Windows Services** and restarting the **FortiMonitor Agent** service.

Linux Agent

1. Open the Agent configuration file (**/etc/fm-agent/fm_agent.cfg**).
2. Add an **agent_proxy** block, for example:

```
[agent_proxy]
http = http://uname:password@216.128.151.185:3128
```

MacOS Agent

1. Open the Agent configuration file (**/usr/local/fortimonitor/agent/config/fm-agent-manifest**).
2. Add an **agent_proxy** block, for example:

```
[agent_proxy]
http = http://uname:password@216.128.151.185:3128
```

Windows Agent Auto-update

Enabling the **Windows Agent Auto-update** feature allows the Agent to periodically check and install new versions for itself. For large deployments, the feature will nullify the need to update each of your Windows servers manually. Enabling the feature once will ensure that all of your Windows Agents are up to date.

Installation issues are handled by the Agent gracefully. If an error occurs, a log file will be created with the error details. The Agent will also rollback, if needed, to its last working version.

Note: There will be a gap of a couple of minutes in data collection while the Agent restarts after the update.

- If you are enabling the feature as part of a new Windows server onboarding, see New Windows instances.
- For Windows servers already in FortiMonitor, see Enable Auto-update in existing Agent installs.

New Windows instances

Oneliner

To enable the Auto-update feature during Agent installation, perform the following steps:

Note: If you are installing the Agent for the first time, see Install the FortiMonitor Agent.

1. From the navigation menu, click **Add**.
2. Select **Windows**.

3. Under the **Quick Install Instructions** tab, copy the generated command.
4. Add the **-autoupdate** flag to the command then run it on a PowerShell console. For example:

```
Invoke-WebRequest https://repo.fortimonitor.com/install/win/fortimonitor_agent_
windows.ps1 -OutFile fortimonitor_agent_windows.ps1; .\fortimonitor_agent_windows.ps1 -
customer_key xxxx-xxxx-xxxx-xxxx -countermeasures -autoupdate
```

After installation, the server will be added to FortiMonitor with its own server key and begin reporting metrics. The Agent will also periodically check for new versions and will update automatically.

Manifest file

The Agent manifest file is only used during the initial installation of the Agent. Subsequent changes must be made using the **FortiMonitor Agent** configuration file. This file can be found in the following directories:

Windows: **\Program Files (x86)\FortimonitorAgent\Agent.config**

For more information, see Use the FortiMonitor Agent manifest file.

To enable the Auto-update feature using the manifest file, perform the following steps:

1. Create or verify the presence of the manifest file in **C:\FortiMonitorAgent.manifest**. At the minimum it should contain the following payload:

```
[agent]
customer_key = xxxx-xxxx-xxxx-xxxx
```

where **customer_key** is your customer key. To obtain your customer key, click your avatar then select **My Account**.

2. Add the following values to the manifest file:

```
auto_update = true
```

The Agent will select a random time within 24 hours of its start up to check for a download; this should be logged in the Agent log.

After installation, the server will be added to FortiMonitor with its own server key and begin reporting metrics. The Agent will check for updates as scheduled and will update accordingly.

3. (Optional) To select a specific time for the agent to update (in local time), use the `- scheduled_update HH:MM` flag. This should be in military-time format; for example, 2:09 PM would be set as 14:09. The agent should query/attempt an update at this time.

Enable Auto-update in existing Agent installs

The Agent configuration file is used in place of the manifest file to enable the Auto-update feature on instances that are already in FortiMonitor. To enable the feature, perform the following steps.

1. Make sure you have the latest version of the FortiMonitor Agent. To update the Agent, see this article.
2. Open the Agent config file **C:\Program Files (x86)\FortimonitorAgent\Agent.cfg**.
3. Add the following values to the **<service>** section of the Agent config file:
   ```
   <add key="AutoUpdate" value="true"/>
   <add key="ScheduledUpdate" value="HH:MM"/>
   ```
   where **HH:MM** is the time at which the update is to be run. For example, **21:05**.

The second value is optional. If it is not entered, the Agent will pick a schedule during start up.

4. Restart the **FortiMonitor Agent** service.

Note: You can restart the service by running the PowerShell command **restart-service FortiMonitorAgent** or by going to Windows Services and restarting the **FortiMonitor Agent** service.

Logging

Log file location

The log files are in the following directory:

C:\Program Files (x86)\FortiMonitorAgent\logs

Update schedule

To see when the Agent will check for updates, filter the Agent log file by **UpdateVersion**.

For example:

```
[2020-11-19 11:22:35,897] [Aggregator.Agent.Workers.UpdateVersionWorker] [INFO] [76] -
     Next timer fire expected 11/20/2020 11:22 AM
```

Installation logs

If the installation fails, an installation log file will be created in the directory where you ran the MSI file for the Agent. Look for the **FortiMonitorAgentSetup.log** file to check for errors.

Add plugins to a blacklist (FortiMonitor Agent)

You have the ability to blacklist certain plugins from ever being run by the agent. This can be useful if you don't want to see warnings about certain plugins requiring more configuration which you don't intend on enabling. A common example of this is Apache's extended status module. To blacklist a plugin, follow the instructions below:

Linux

To create a plugin blacklist for a Linux Agent, you will need to add a line to the Agent configuration file.

- **/etc/fm-agent/fm_agent.cfg**

Under the **[AgentConfig]** section of this file, add the following:

```
plugin_blacklist = [plugin textkey list]
```

Replacing the **[plugin textkey list]** with a comma-separated list of plugin textkeys. To find out what plugin textkeys to use, you can run the following command:

- `ls /usr/lib/fm-agent/plugins/`

This will list out all of your Agent's plugins. The plugin textkey is just the name of the plugin's file without the .py ending.

Now the Agent will no longer check on the configuration status of the services related to the plugins that you have placed on your blacklist.

Windows

To create a plugin blacklist for a Windows Agent, you will need to add a line to the Agent configuration file (**C:\Program Files (x86)\FortiMonitorAgent\Agent.conf**). Within the tags of this file, add the following:

```
<add key="PluginBlacklist" value="[resource category list]" />
```

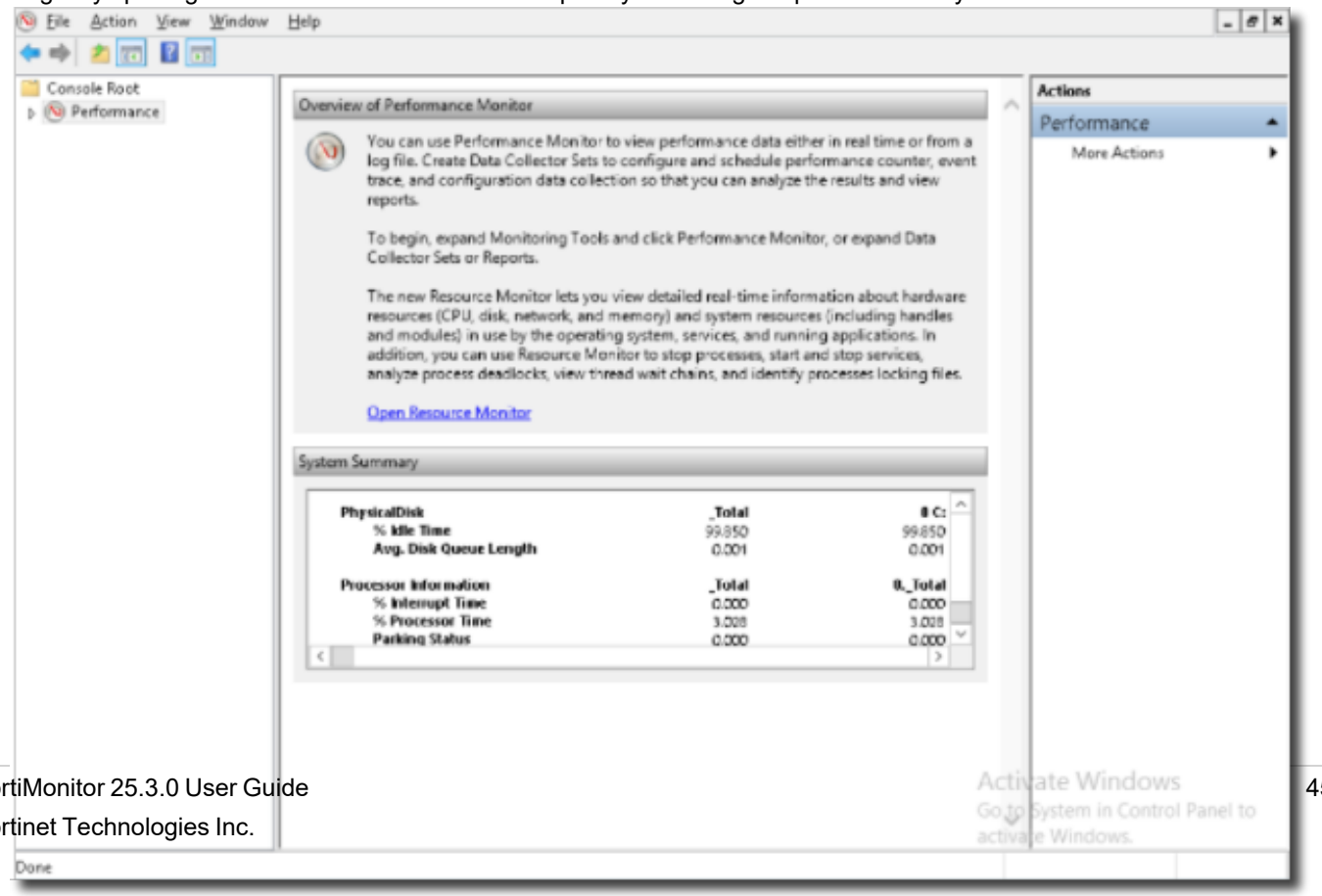
Replacing the **[resource category list]** with a comma-separated list of resource categories. To find out what resource categories to use, you can reference the list found within the tags of the configuration file that you are editing.

Once you have done this, restart the panopta-agent service. For instructions on how to that see this page for reference. Now the Agent will no longer check on the configuration status of the services related to the plugins that you have placed on your blacklist.

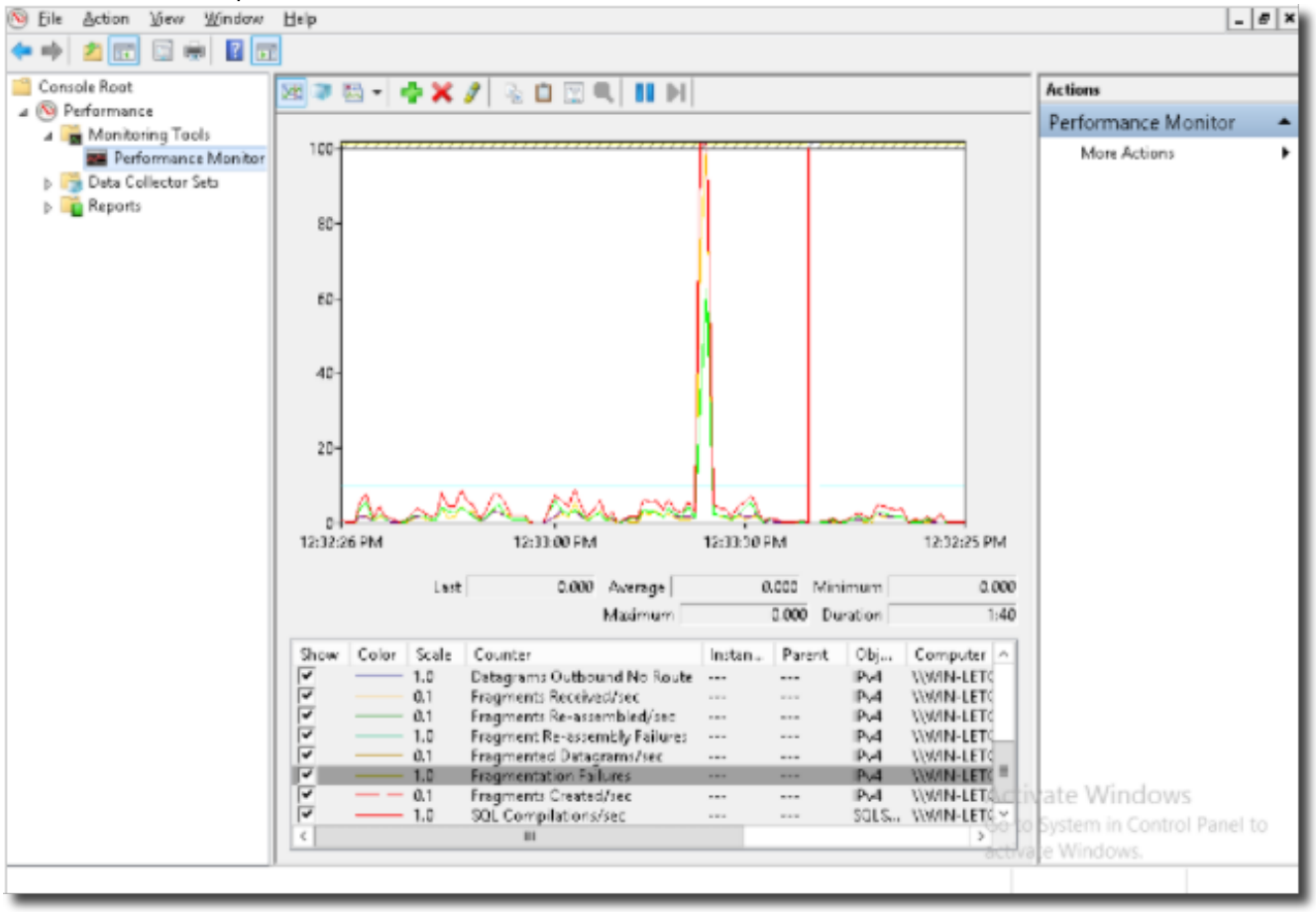
Add metrics using Windows Performance Monitor

The Windows Agent uses the PerfMon framework to gain access to system-level metrics on a Windows device. PerfMon exposes thousands of different metrics to help provide visibility on status and performance on the device. The Windows Agent comes configured with a standard set of metrics that serve as a good starting point for most use cases. However, if you need to monitor metrics not provided by the standard set, you can do so by following some very simple configuration steps.

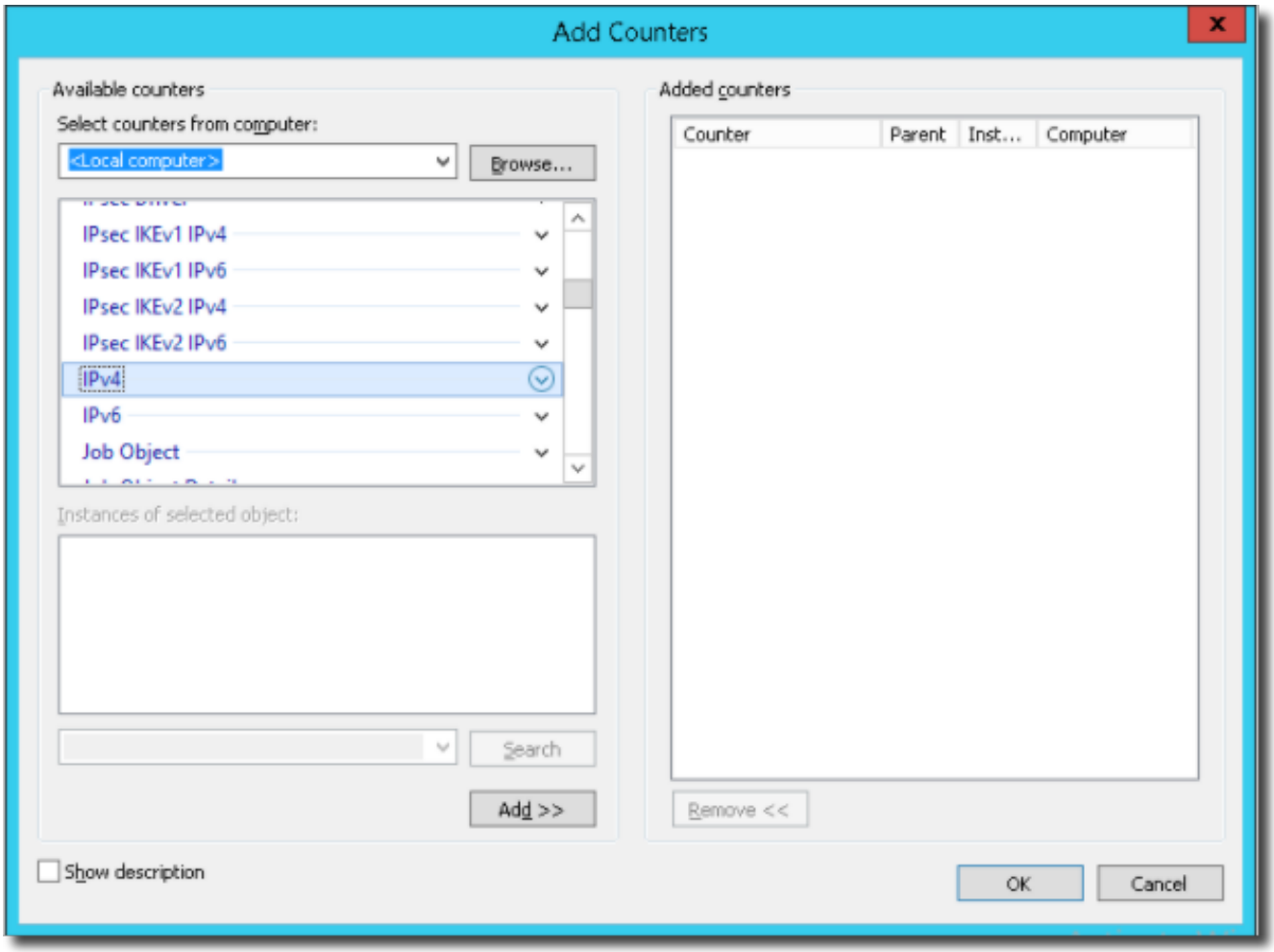
Begin by opening the the Performance Monitor snap-in by searching for “perfmon” from your start search menu.



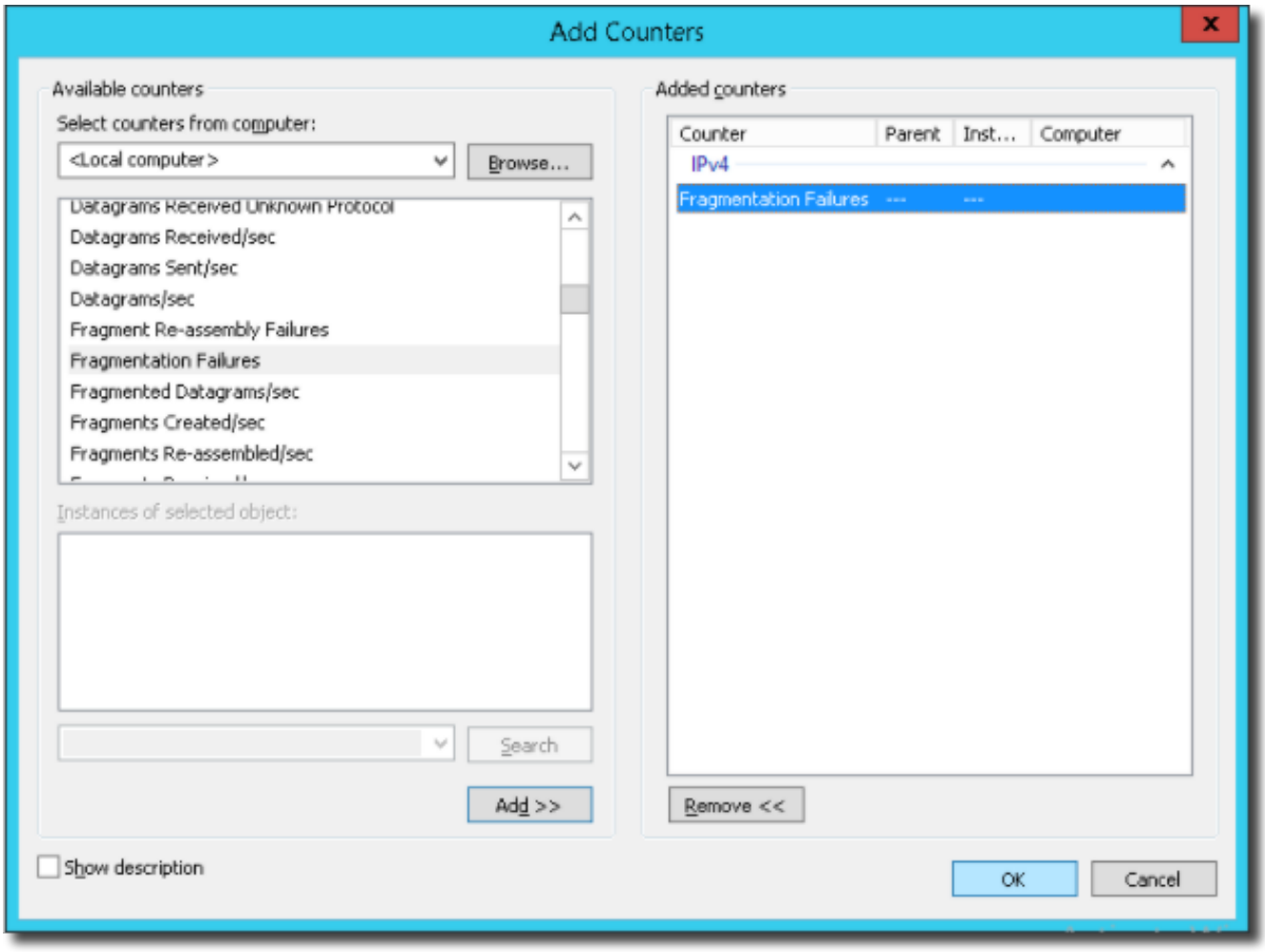
Now drill down to the performance monitor itself as seen below:



Press the green “+” button to see a list of all the performance metrics you can monitor with this program.



Here you will need to find the metric that you would like to monitor. For this example, we will select the “Fragmentation Failures” from the “IPv4” drop-down. Take careful note of the name of the metric and the category it is in.



Activating your metrics

Once your metrics are confirmed to be running in Perfmon, you'll need to tell the FortiMonitor agent to monitor them. There are two ways to do this, with Option #1 being the easiest and newest.

Option 1: Activating your metrics via the Control Panel

Under settings, go to `Perfmon Metrics`. You'll see a table that may be empty if you've never configured a custom Perfmon metric or one that will have rows if you have.

For each Perfmon metrics you'd like to add, you must enter at least the Perfmon Counter name and the Metric name. For example, to add monitoring for Requests Queued in ASP .NET, you would add the following record:

- Perfmon Counter
- Metric
- Unit
- ASP .NET

- Requests Queued
- requests

You can find all of these from the Perfmon tool itself as described in the first portion of this page.

After you hit save, you'll need to restart each Windows agent that you'd like to have picked up these metrics. As well, you'll need to add the monitoring for these metrics on their respective instances within the FortiMonitor control panel. You can learn more about that here.

Option 2: Activating your metrics via the agent config file

Now that you have found the exact syntax of the metric we would like to monitor, it is time to add this metric to your agent config file. Open up the C:\Program File\PanoptaAgent\Agent.config file in a basic text editor. Under **<plugin name="PerfMon">** you can add a new field for your chosen metric. As seen below we added the "Fragmentation Failures" metric under the "IPv4" category as we discussed above.

```
</service>
<plugins>
  <plugin name="PerfMon">
    <resource category="IPv4" label="Fragmentation Failures" unit="failures" />
    <resource category="APP_POOL_WAS" label="Current Application Pool State" unit="Current State" />
    <resource category="ASP.NET" label="Requests Queued" />
    <resource category="ASP.NET" label="Applications Running" />
```

Both the "Category" and the "Label" field must exactly match the metric you selected in the Performance Monitor. You can fill out the "unit" field as you wish, it will show up in the FortiMonitor control panel alongside your metric for reference.

You can add as many metrics in this fashion as you wish. To complete these configuration changes, you will need to restart your FortiMonitor Agent service (using the services program). After a few minutes, your machine will sync with FortiMonitor's servers and you should be able to add this new resource like you would add any other. If you do not know how to add an Agent resource then please see this article.

If you plan on adding this set of new metrics to more than one of your Windows Agents, please contact support! We can add a base PerfMon template to your account to expedite this process and make adding these checks easy! Once we set up the template for you, you can run a metadata refresh and the new metrics will be available to add on all your instances/servers.

Troubleshooting:
If these are added incorrectly you may see some issues in the agent log that look like:

```
Performance Counter \\IPv4\\Fragmentation Failures does not exist

Category does not exist.
```

Prepare a FortiMonitor Agent for server imaging/cloning

In some situations, you may want to install an Agent to a server and use that as a base image. If that is the case, you will need to follow the following steps to ensure that the agent is cloned on the server properly. Failure to do so may result to having multiple servers with the same server key. If there are multiple servers with the same server key, that will create sync issues from your hosts to our platform.

For Linux Servers

- Stop cron from running.
  RHEL: `service crond stop`
  Ubuntu: `systemctl stop cron`
- Confirm that cron is inactive.
  RHEL: `service crond status`
  Ubuntu: `systemctl status cron`

You should see output similar to the below snippet to indicate that cron is inactive.

```
# service crond status
Redirecting to /bin/systemctl status crond.service
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Mon 2021-06-28 16:37:17 EDT; 5s ago
```

- Then you need to remove some files the agent creates when it is installed and run.
  - Delete the **fm-agent.db** file.
    `rm /var/lib/fm-agent/fm-agent.db`
  - Delete the **fm_agent.cfg** file.
    `rm /etc/fm-agent/fm_agent.cfg`
- Begin creating the server image.



If you restart your server before creating an image, you will need to perform these steps over again unless cron is disabled upon reboot.

For Windows Servers

- Create the **FortiMonitorAgent.manifest** file in the root of the C: drive. For more information, see Use the Agent manifest file.
- Download the latest dotNet4.0 or dotNet4.5 server agent.
- Run the following command from a PowerShell Window as an administrative user:

```
Start-Process -FilePath msiexec -Wait -ArgumentList "/q INSTALLATION=`"Install`" /i
fortimonitor-agent-prod-NET4.5.latest.msi MANIFESTFILE=`"FortimonitorAgent.manifest`"
IMAGEINSTALL=`"true`""
```



The MANIFESTFILE parameter is the path to the manifest file created above. If this parameter is not present, the installer will look for the manifest in C:\ by default.

- Begin creating the server image.

Rebuild the FortiMonitor Agent metadata

Upon installation and once an hour thereafter, the FortiMonitor Agent performs metadata discovery. On all of our platforms, it will do a check for applications it has in its inventory to see if they've been added. It will also check for new peripherals such as mounted drives and network interfaces. You can also kick-off a metadata scan manually from any server in the control panel.

Manual Metadata Rebuild

To manually rebuild the metadata for a server:

1. Navigate to the details page of any server with the agent installed.
2. In the right side of the page, in the Agent Status block, click **Rebuild Metadata**.

For Linux Agents, you can also rebuild the metadata by entering the following command:

- For the **fm-agent**: `python3 /usr/bin/fm-agent/fm_agent.py --rebuild-metadata`

The agent will report the results of the discovery in the next 2-3 minutes. This results in a couple of things:

1. Any newly discovered applications will be shown as **Discovered** in the quick-add section of the **Monitoring Configuration** table.
2. If you're utilizing templates with recurring application, any new metadata discovered will cause your template to check and see if it has any monitoring to apply

IP matching

The FortiMonitor Agent has default IP matching which fires during the initial handshake. This IP matching can be helpful or the following scenarios:

- If a user may have uninstalled the Agent and is looking to reinstall. Rather than creating a new entry/object, we would want to attach the Agent to the same server as before to avoid duplication and keep all historical data. This works by the Agent sending the IP addresses of the instance to our cloud, and then we check to see if those IPs exist in your account. If they do, we automatically attach the Agent back to that particular server.
- For users moving on from basic instances (without the Agent) to advanced instances. In this case, instead of the Agent setting up a new instance (duplicating the existing instance), the IP matching would step in and assign the Agent to the server with the matching IP.

However, this can backfire in cases with multiple networks and overlapping IP ranges. If you happen to have the same IP in separate networks monitored by FortiMonitor, you can disable IP matching. The option is called **disable_server_match** and is further documented here.

Start or end Maintenance from the FortiMonitor Linux Agent

You can start or end maintenance on the server where the Linux Agent is installed using the server's command-line interface. Starting maintenance will end any active incidents on the server and prevent new ones from being created while the maintenance period is active.

To start or end the maintenance, run the following command with the appropriate options:

```
python /usr/bin/fm-agent/maintenance.py
```

Option	Description
• h, --help	Lists all of the supported options and their description.
• s, --start	Start maintenance. This option must always be used with **-d** or **--duration**.
• d DURATION, --duration=DURATION	The duration of the maintenance in minutes.
• e, --end	End any active maintenance on the server.
• t METRIC_TAGS, --metric-tags=METRIC_TAGS	An optional comma-separated list of tags to pass to the maintenance.

For example, to start a maintenance period of 120 minutes, run the following command:

```
python3 /usr/bin/fm-agent/maintenance.py --start --duration=120
```

An error will occur if you start a maintenance period without including a duration.

To end any active maintenance on the server, run the following command:

```
python /usr/bin/fm-agent/maintenance.py --end
```

Log messages can be found in **/var/log/fm-agent/maintenance.log**.

Migrating a Panopta Agent configuration to FortiMonitor Agent

The following sections describe how to migrate your existing Panopta Agent configuration to a new FortiMonitor Agent installation.

Linux Panopta Agent migration

Note: Python 3 is required for all Linux Agent installations. Ensure that Python 3 is installed before proceeding with the migration. See FortiMonitor Agent Security for encryption and communication ports/ protocols.

To migrate your Panopta Linux Agent configuration, run the following command:

```
curl -s https://repo.fortimonitor.com/install/linux/fm_agent_install.sh | bash /dev/stdin
     -c <customer-key> -x
```

Where -x is the argument for migration.

When the migrate flag (-x) is used to install the new FortiMonitor Linux Agent on a server that already has the Panopta Agent installed, the following occurs:

1. The migrate flag alerts the install script to create and configure a new FortiMonitor manifest file (**/etc/fm-agent-manifest**) based on the existing Panopta Agent configuration file (**/etc/panopta-agent/**).
   - If the **panopta_agent.cfg** file does not exist, a very unlikely scenario, a new clean manifest file will be generated. The Agent will still merge to the existing instance, however, the FortiMonitor configuration file (**/etc/fm-agent/fm_agent.cfg**) will only contain default properties. Any additional properties or code blocks that were previously configured will not be present and the **aggregator_url** property will default to **rx.us01.fortimonitor.com**.

- If the **aggregator_url** property in the existing **panopta_agent.cfg** file is configured to the default Panopta value (**aggregator2.panopta.com:443**), it will be changed to the new default (**rx.us01.fortimonitor.com**). All other existing values, including proxy settings will be migrated over.

2. The FortiMonitor Agent installation process is started.
   - If the connectivity test to **rx.us01.fortimonitor.com** fails, the installation process stops, and the current Panopta Agent installation remains.
3. A backup of the following is created in this directory **/tmp/panopta-agent-migration-bkp**.
   - Panopta Agent configuration file: **/etc/panopta-agent/panopta_agent.cfg**
   - Panopta CounterMeasures: **/usr/lib/panopta-agent/countermeasures/plugins**
   - Panopta custom plugins: **/usr/share/panopta-agent/**
4. The custom plugins from **/usr/share/panopta-agent/** are migrated to **/usr/share/fm-agent/**.
5. The custom CounterMeasures from **/usr/lib/panopta-agent/countermeasures/plugins** are migrated to **/usr/share/fm-agent/countermeasures**.
6. The Panopta Agent is uninstalled.

Migrate using a local Linux Agent package

For Linux servers that are restricted from accessing the internet, the Panopta Agent can be migrated using a local Agent package.

1. Download the latest Agent package. Contact support for the download information.
2. Copy the package to the Agent server.
3. Run the following command:
   ```
   python3 linux_fm_agent_install.py --migrate --customer-key <customer-key> --local
   <path to local rpm/deb pkg>
   ```
4. Verify that all of the custom metrics, plugins, and CounterMeasures are migrated to the new FortiMonitor Agent.

Windows Panopta Agent migration

To migrate your Panopta Windows Agent configuration:

1. Open PowerShell in administrator mode and run the following command to enable tls1.1 and 1.2 for the current session:
   ```
   [System.Net.ServicePointManager]::SecurityProtocol =
   [System.Net.SecurityProtocolType]::Tls11 -bor
   [System.Net.SecurityProtocolType]::Tls12;
   ```
2. Run the following command:
   ```
   Invoke-WebRequest https://repo.fortimonitor.com/install/win/fm-upgrade.ps1 -OutFile fm-
        upgrade.ps1
   .\fm-upgrade.ps1
   ```

The migration script only takes one **optional** argument, **-autoupdate**. If this is set, it modifies the Agent configuration file to enable autoupdate.

When the **fm-upgrade.ps1** script is used on a server that already has the Panopta Agent installed, the following occurs:

1. A backup of the following sources is created in the Windows User Temp Folder (**C:\Users\ {user}\AppData\Local\Temp\ panoptabkp_{random_characters}**).
   - Panopta Agent configuration file: **C:\Program Files (x86)\PanoptaAgent\Agent.config**
   - Panopta Agent CounterMeasures: **C:\Program Files (x86)\PanoptaAgent\cm_ps_plugins**
   - Panopta Agent Custom Plugins: **C:\Program Files (x86)\PanoptaAgent\ps_plugins**
2. The Panopta Agent is uninstalled.
3. The FortiMonitor Agent installation process is started.
4. The **AggregatorUrl** property of the Agent configuration file is checked. If it is pointing to **aggregator2.panopta.com** and the connectivity check to the new aggregator **rx.us01.fortimonitor.com:443** is successful, the **AggregatorUrl** property will be updated to **rx.us01.fortimonitor.com:443**. Otherwise, the **AggregatorUrl** will remain pointing to **aggregator2.panopta.com**.
5. The backup is copied to the following directories:
   - **Panopta Agent configuration file** to **C:\Program Files (x86)\FortiMonitor\Agent.config**
   - **Panopta Agent CounterMeasures** to **C:\Program Files (x86)\ FortiMonitor \cm_ps_plugins**
   - **Panopta Agent custom plugins** to **C:\Program Files (x86)\ FortiMonitor \ps_plugins**

Migrate using a local Windows Agent package

For Windows servers that are restricted from accessing the internet, the Panopta Agent can be migrated using a local Agent package.

1. Download the latest Agent package. Contact support for the download information.
2. Copy the package to the Agent server.
3. Run the following command:
   `Invoke-WebRequest https://repo.fortimonitor.com/install/win/fm-upgrade.ps1 -OutFile fm-upgrade.ps1 .\fm-upgrade.ps1 -local <FILENAME>`
   Where `<FILENAME>` is the path to the downloaded MSI file. If you are specifying a full path, use quotes.
4. Verify that all of the custom metrics, plugins, and CounterMeasures are migrated to the new FortiMonitor Agent.

Troubleshooting the FortiMonitor Agent

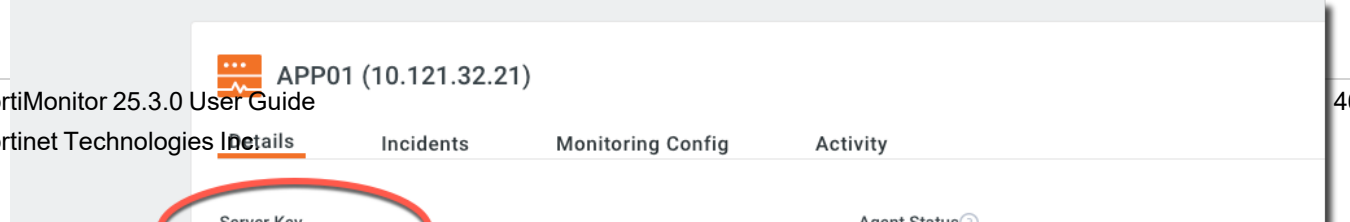
This article will cover a number of problems that customers run into with their Agent monitoring.

Installation

The first thing to check is that the server key for your Agent matches the Server key for that server in your Panopta control panel. On a Linux server you can always find your server key by typing the following command into your command window:

```
python3 /usr/bin/fm-agent/fm_agent.py
```

On a Windows server, you can look in the **Agent.config** file. To find the server key for your server according to your FortiMonitor account, go to that server's **Instance Details** page on your FortiMonitor account. It will be listed in the **Details** tab. If these do not match then you need to edit your server in the Panopta control panel and add the correct server key from your command window.



Agent communication and connectivity

A common problem is the Agent not being able to communicate with the FortiMonitor monitoring cloud. By default, the Agent attempts to communicate with **https://rx.us01.fortimonitor.com** on port 443, which is fully encrypted with TLS. If you're looking to test the connectivity, you can attempt to reach the hello endpoint located at **https://rx.us01.fortimonitor.com/v2/hello** or using **curl/wget** or a browser on a Windows server. You will see:

```
{"hello": "world". "timestamp": }
```

If the server does not have an outbound connection to the internet, you can alternatively proxy through an OnSight collector.

System time

Often times, you may see unexplainable gaps in graphs for metrics when the Agent appears to be reporting consistently. This happens if the system time on the server is drifting from the correct time and differs from our central infrastructure. Make sure to check the time settings on your machine and use our NTP (Linux) monitoring to alarm on this.

Adding new resources

Another frequently encountered issue is that after installing a new component to the Agent the new resource is not available to add on the control panel for that server. You can solve this problem by clicking on the three dot drop-down on your server's page in the control panel and selecting **Rebuild Agent Metadata**. This will force the metadata to regenerate, and this should let our servers know that this Agent is capable of collecting data on this new resource. This will take about a minute to take affect, but once it does the resource you were looking for should now be available.

If all this fails then it is time to talk to us, send us your latest Agent logs and we will find out what is wrong for you. To find this log in Linux you can type in the following command into the command window:

```
less /var/log/fm-agent/fm-agent.log
```

To find this log in Windows, enter the following command into the PowerShell command window:

```
Invoke-Item c:\FortimonitorAgent\logs\agent.log
```

Duplicate server keys

In some cases, you may encounter two different agents receiving back the same server key upon initial handshake. This causes the metrics for two different instances to be combined into one instance in your control panel. Panopta will attempt to give a server the same key if it appears to be a previous server using the agent. Read about IP Matching to learn more.

OS-specific troubleshooting

For Agent issues related to Linux, click here.

For Agent issues related to Windows, click here.

FortiMonitor Linux Agent

Unable to download the FortiMonitor Agent

If you are using Redhat or Centos and you can not download the Agent from our repositories, perform the following:

1. Run the following command to clear your repositories cache:

```
yum clean all --disablerepo="*" --enablerepo=fortimonitor.repo
```

2. Try the installation again.

3. If it does not work, run the following command and send the output to FortiMonitor support:

```
yum --disablerepo=""*"" --enablerepo=""fortimonitor.repo"" list available
```

Agent not running

If you notice the agent is not running after the initial install, the agent's cron entry may be getting blocked by the system.

1. Check **/var/log/syslog** to see if some other server-level security it blocking the **fm-agent** cron from running. Servers using PAM or AppArmor often suffer from this and require the **fm-agent** user to be added to **access.conf**.
2. You may also have to add the **fm-agent** user to **cron.allow** file.
3. For users running CageFS on their machines, you would need to disable **cagefs** for the **fm-agent** user in order for the Agent to run properly. You can do this using the following command:
   ```
   /usr/sbin/cagefsctl --disable fm-agent
   ```

Agent not running due to locked account

The FortiMonitor agent installation creates a limited user account in order to scope all execution properly. By default, that user has no shell login allowed and has password set to never expire. That being said, some security tools/agents will force passwords to expire. Once the password is expired, the agent may not continue to run on it's schedule. In order to resolve this, you need to set the password to never expire.

1. Check password expiration by running the following command:
   ```
   sudo chage -l fortimonitor-agent
   ```

You will see an output similar to the following:

```
Last password change : Mar 13, 2018
Password expires : Jun 11, 2018
Password inactive : Jul 11, 2018
Account expires : never
Minimum number of days between password change : 7
Maximum number of days between password change : 90
Number of days of warning before password expires : 7
```

If lines 2 and 3 indicate an expired password, change the password to never expire by running the following command:

```
sudo chage -I -1 -m 0 -M 99999 -E -1 fm-agent
```

Agent not running due to permission problems in log files

In some cases, when the Agent is originally run from root to debug a situation, permissions on log files can be set improperly. To resolve the issue:

1. Set the ownership on the Agent log files back to **fortimonitor-agent**.

Error with a partial Linux installation

If you run into an error with a partial installation:

1. Uninstall the Agent.
2. Reinstall the Agent.

If you get an error with the uninstallation, you can force remove the Agent by running the following commands:

Redhat

```
rpm -e --noscripts fm-agent
yum clean all
```

Debian

```
apt-get purge fm-agent
dpkg --purge fm-agent
```

After this, reinstall the agent and wait for the sync to happen. If the agent does not begin to sync, look in the contents of **fm-agent** in **/etc** and see if the aggregator URL is there. If not, please add the URL to the file. The aggregator URL (if not using the OnSight as a proxy) is **https://rx.us01.fortimonitor.com**.

Agent not seeing network attached storage/partitions

If your Agent isn't able to see your network-attached storage or partitions, it's likely a permissions issue for the Agent User.

1. You can confirm this by first checking if the Agent User can see the drives.
2. Test to see if root can see the drives.
   If root sees them, and the Agent User does not, you'll need to give the Agent User permission to access those drives.
3. Once that's complete, you need to force a metadata rebuild on the Agent from the control panel.

Non-root users are unable to run cronjobs

Cron is only working for root users. When a cron job is run, it fails with the following error in **/var/log/messages**:

```
Mar 19 15:06:01 p123456s crond[27xxx]: Permission denied
Mar 19 15:06:01  p123456s crond[27xxx]: CRON (panopta-agent) ERROR: failed to open PAM security
session: Success
Mar 19 15:06:01  p123456s crond[27xxx]: CRON (panopta-agent) ERROR: cannot set security conte
```

This solution only applies if you wish to use a custom **access*.conf** file to list which users are allowed to use cron.

1. Ensure that **/etc/pam.d/crond** is referencing the correct access*.conf file:

```
$ cat /etc/pam.d/crond
#
# The PAM configuration file for the cron daemon
#
#
auth         sufficient    pam_env.so
auth         required      pam_rootok.so
auth         include       system-auth
account      sufficient    pam_succeed_if.so uid = 0 use_uid quiet
account      sufficient    pam_listfile.so onerr=succeed item=user sense=allow
file=/etc/security/access-cron.conf
account      include       system-auth
session      required      pam_loginuid.so
session      include       system-auth
```

2. Ensure that the user running cron (in this case **cron_user**) is listed in **/etc/cron.allow**:

```
# cat /etc/cron.allow
root
user1
cron_user
```

3. Ensure the user is allowed to log in via **/etc/security/access.conf**:

```
+ : root : ALL
+ : cron_user : ALL
+ : @u1_dbadmin : ALL
+ : @u2_sysadmin : ALL
- : ALL : ALL
```

4. A custom list of users that can run cronjobs is at **/etc/security/access-cron.conf**. However, PAM does not reference **/etc/security/access-cron.conf** when authenticatingfor crond. The default for **pam_access.so** is to reference **/etc/security/access.conf** when access file is not used.

Installation issues on RHEL 5.x servers

When installing the FortiMonitor Linux Agent on RHEL 5.x servers, you may run into the following errors:

```
Error checking SSL compatibility - cannot import SSL libraries!
```

or

```
[Errno -3] Error performing checksum
```

This means that you will need to update the sha256 ciphers to calculate the checksums for newer RedHat repositories.

You can do this by the following method:

Use yum to install `python-hashlib` on the machine and then try to install the FortiMonitor agent via the repo again.

```
yum install python-hashlib
```

How to Uninstall the Agent

To uninstall the Agent, run the following commands:

Debian

```
apt-get remove fm-agent
```

Redhat

```
yum remove fm-agent
```

Still having trouble?

Please send the FortiMonitor Agent logs to our support team so we can further assist you. You can find the logs in:

Log file location

```
/var/log/fortimonitor-agent/
```

Troubleshooting the FortiMonitor Linux Agent Heartbeat Incident

An Agent heartbeat incident occurs when the FortiMonitor Agent is unable to sync to our platform. The following troubleshooting steps should be performed on Linux servers experiencing heartbeat incidents.

Determine if the server running the Agent is able to reach our platform

- From the Linux terminal, run the following command:

```
https://rx.us01.fortimonitor.com/v2/hello
```

- You should receive a response similar to the following:

```
{"hello": "world", "timestamp": 1612463423}
```

If you do not receive a response similar to what is shown in the example above, this indicates that this server is unable to reach our platform and is unable to sync. If this server cannot reach our platform, the agent will experience an agent heartbeat incident. Please check your network configuration to determine why the server cannot reach our platform.

Verify that the cron service is running on the server

The Agent uses cron to run. If cron is not running on the server, then the agent will not be able to sync. This will result in an agent heartbeat incident.

The command to check if cron is running on Ubuntu: `service cron status`
The command to check if cron is running on RHEL: `service crond status`

Confirm that the Agent Server Key matches the Server Key of the affected FortiMonitor instance.

- To check the server key in Linux, run the following command:

```
$ grep -i server_key /etc/fm-agent/fm_agent.cfg
```

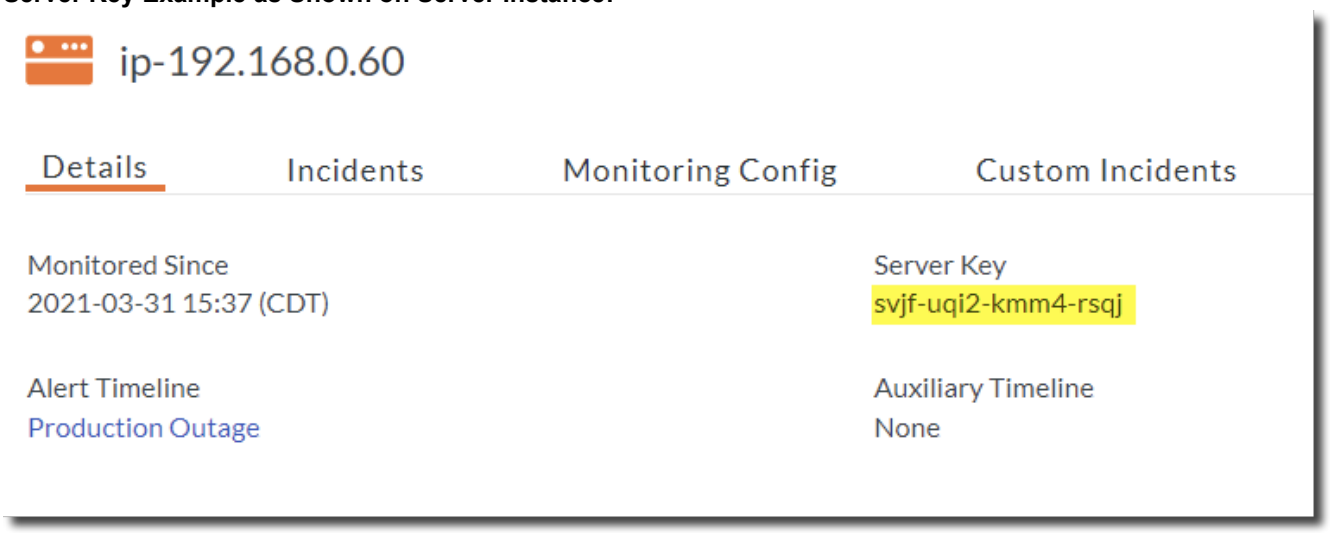
You should receive output similar to the following:

```
server_key = svjf-uqi2-kmm4-rsqj
```

Once you have the agent server key, confirm that this key matches the key that you are seeing in the FortiMonitor Control Panel. If the server keys do not match, confirm you are looking at the right server.

If you are looking at the correct server, then you will need to change the server key on the instance to match the key that is found in the **fm_agent.cfg** file.

Server Key Example as Shown on Server Instance:



Confirm that the User Agent is not locked out and the password is unexpired

- This article covers the process of checking the agent password expiration.

If the steps above do not identify the cause of the agent heartbeat problem, please send us a copy of your agent logs. Our support team will look at them and follow up with you.

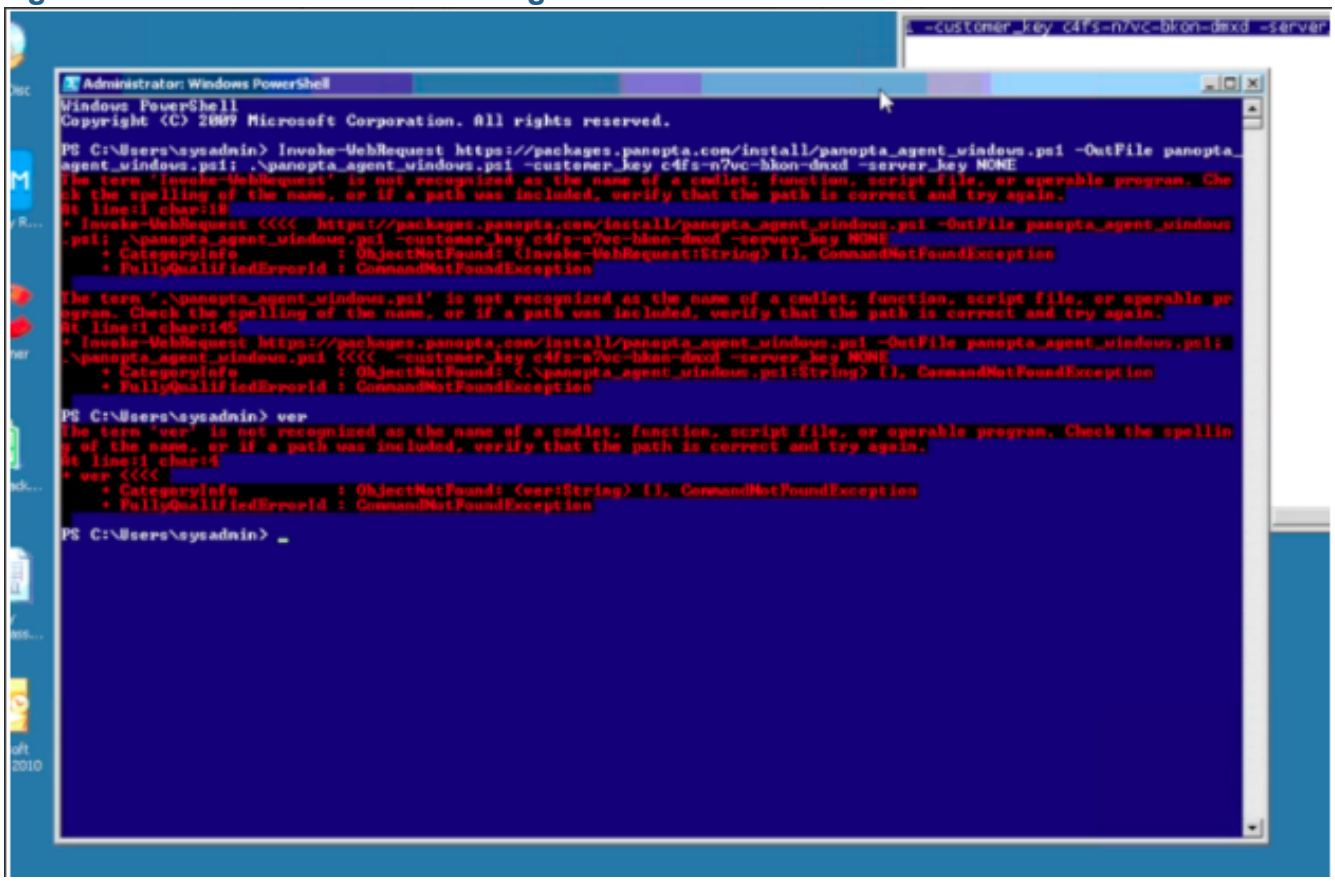
fm-agent log location: **/var/log/fm-agent/**

FortiMonitor Windows Agent

Quick Tips

- Confirm that the proper .NET version is installed for the Agent you are using (.NET 4.0 or 4.5).
- Check **Add/Remove programs** (for MSI Agent) to confirm if installation was completed.
- Verify if the **Windows Agent service** is running/started.

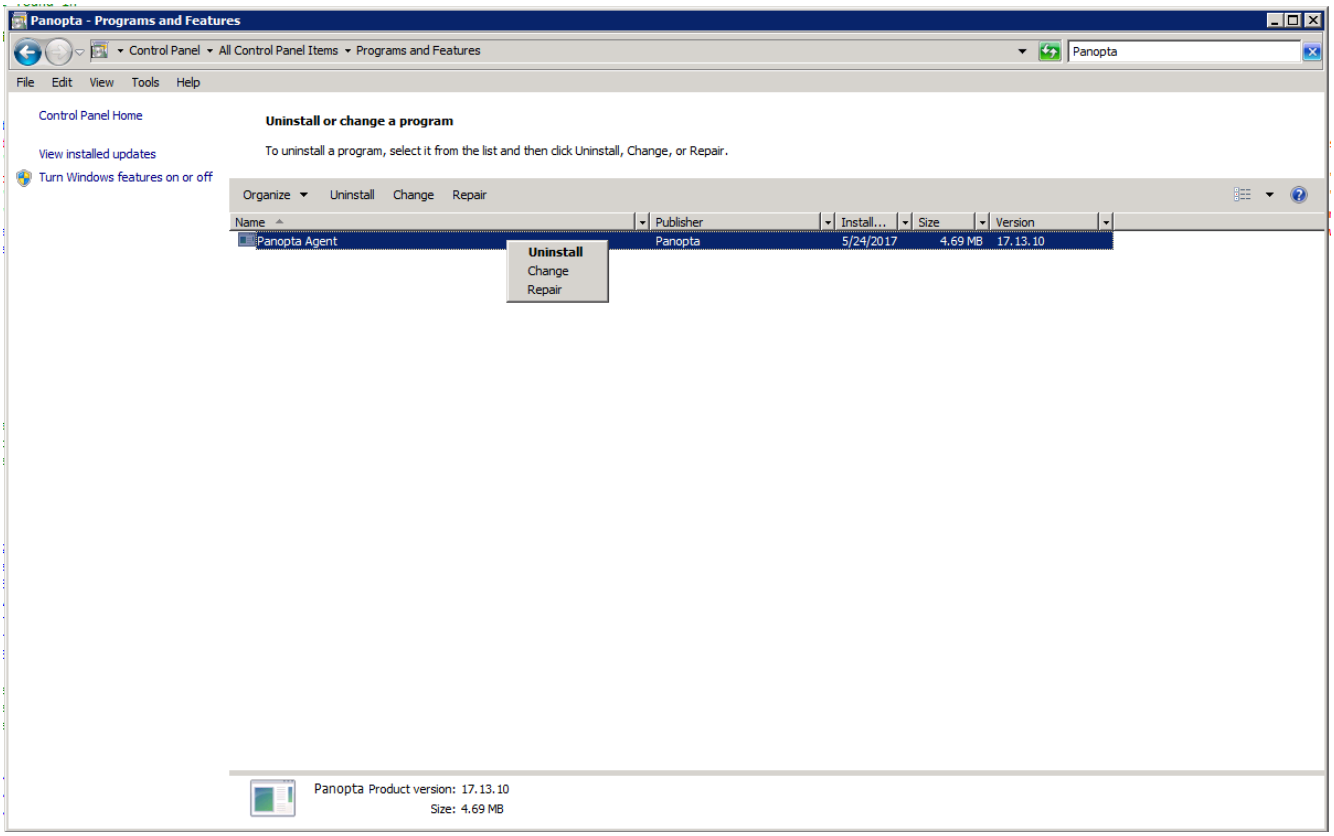
Agent one line installer not working



If this occurs, you should either update your version of PowerShell, or follow the manual instructions for installing the agent.

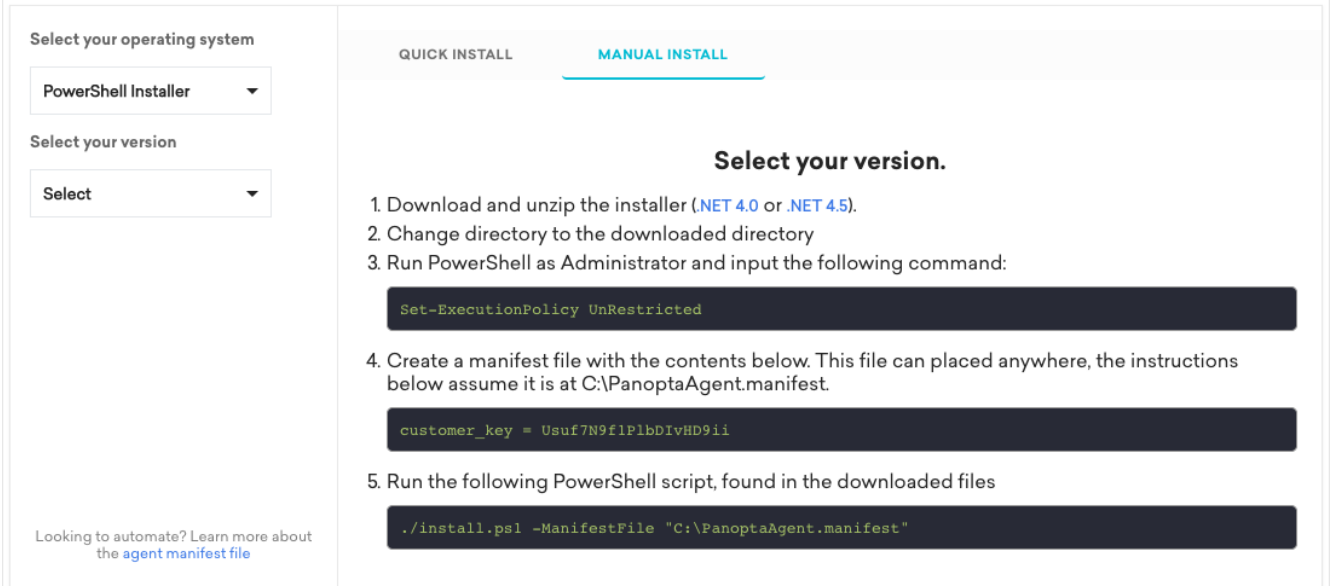
How to uninstall the Agent

You can uninstall the Agent by removing it from the programs menu, you can follow detailed instructions here.



32bit/64bit Agent Installations

For installing the 86x version of the Agent, you must follow the manual install instructions for PowerShell installation instead of using the one-liner.



Error installing due to unsupported protocol (TLS 1.0)

If the windows agent fails to install, you can typically find the reason in the install log which is located in the root C drive. If you are seeing the following error:

```
AGENT: Check to Aggregator API failed with message: The requested security protocol is not supported.
```

This means that server is attempting to communicate with the FortiMonitor cloud over the deprecated (and insecure) TLS 1.0 protocol. This can be very common on windows 2008 R2 servers which do not have Service Pack 2 and only have .NET 4 installed. In order to solve the problem, you must either apply SP2 or install .NET framework 4.5 or higher which dynamically selects the right TLS protocol. Once you have installed that, restart the windows agent install and it should succeed.

More information on this topic can also be read on the following Microsoft help article: https://support.microsoft.com/en-us/help/3154517/support-for-tls-system-default-versions-included-in-the-net-framework

Certain metrics don't appear as available in the metric pane

You may find that certain core system/application metrics are not showing as "available" on your windows server. This can occur if the underlying Perfmon counter has become corrupted on the system. This problem can be resolved by running the following command in an administrative PowerShell command window:

```
lodctr /R
```

Once this has completed, restart the windows agent and allow a few minutes for the agent to report up the available metrics.

More information can be found on the **lodctr** command here: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/lodctr

Additional support

Please send the Agent logs and Windows Event Application Log to our support team so we can further assist you. You can find the Agent logs in the following directory:

C:\Program Files (x86)\FortimonitorAgent\Logs .

In some cases, to obtain additional diagnostic information, it may be required to put the Agent into **DEBUG** mode. This triggers the Agent to add more verbose logging of its internal operations. To do this, perform the following steps:

1. From the Windows Agent server, open **C:\Program Files (x86)\FortimonitorAgent\bin\logging.config**.
2. Locate the line that shows:
   `<level value="INFO" />`
3. Edit the line to enable **DEBUG** logging. The line should now look like:
   `<level value="DEBUG" />`
4. Save the changes.
5. Go to `C:\Program Files (x86)\FortiMonitorAgent\logs` and remove the current **agent.log** , **discovery.log**, and **metric.log** files.
6. From the **Windows Services Control Panel**, restart the **FortiMonitorAgent** service to apply the changes to the logging level.

7. Rebuild the metadata for the Windows Server and allow 5-10 minutes for the data to collect.
8. Send the Agent logs and Windows Event Application Log to our support team.

Troubleshooting FortiMonitor Windows Agent Heartbeat Incident

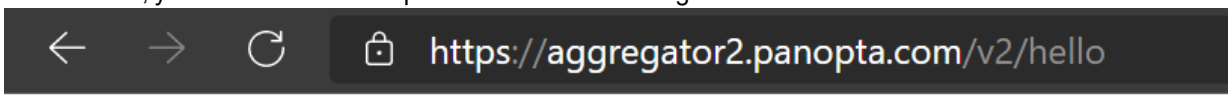
An Agent heartbeat incident occurs when the Agent is unable to sync to our platform. The following troubleshooting steps should be performed on Windows servers experiencing heartbeat incidents.

Determine if the Server Running the Agent is Able To Reach Our Platform

- Method 1: Checking through Windows PowerShell (Useful for Windows Server Core)
  - In a Windows PowerShell command line, run the following command:
    `invoke-webrequest https://rx.us01.fortimonitor.com/v2/hello`
  - You should receive a response similar to the following:

```
StatusCode        : 200
StatusDescription : OK
Content           : {"hello": "world", "timestamp": 1617805078}
RawContent        : HTTP/1.1 200 OK
                    X-Frame-Options: deny
                    Alt-Svc: clear
                    Content-Length: 43
                    Content-Type: application/json
                    Date: Wed, 07 Apr 2021 14:17:58 GMT
                    Via: 1.1 google

                    {"hello": "world", "timestamp": 1617...
Forms             : {}
Headers           : {[X-Frame-Options, deny], [Alt-Svc, clear], [Content-Length, 43],
[Content-Type,
                    application/json]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : System.__ComObject
RawContentLength  : 43
```

- Method 2: Checking through Web Browser
  - In a web browser on the target server, enter the following web address into the browser
    `https://rx.us01.fortimonitor.com/v2/hello`
    If successful, you should receive output similar to the following in the web browser
    

  - {"hello": "world", "timestamp": 1617811272}

If you do not receive a 200 status code similar to what is shown in the method 1 example above, or if method 2 fails, this indicates that this server is unable to reach our platform and is unable to sync. If this server cannot reach our platform, the agent will experience an agent heartbeat incident. Please check your network configuration to determine why the server cannot reach our platform.

Outbound traffic requirements can be found in this article.

Verify That the FortiMonitor Agent Service Is Running

- Method 1: Checking through PowerShell (Useful for Windows Server Core)
  - In a Windows PowerShell command line, run the following command:
    `Get-Service FortimonitorAgent`
  - You should receive output to show that the service is “Running”.

    ```
    Status   Name               DisplayName
    ------   ----               -----------
    Running  PanoptaAgent       FortiMonitor Agent Service
    ```

  - If the **FortimonitorAgent** service status is shown as **Stopped**, you can run the below command to start it.
    `Start-Service FortimonitorAgent`
- Method 2: Checking through Windows Service Application
  - Open the Windows services application. You can find this by searching for **Services** in the **Windows Start Menu**.
  - In the Windows services application, confirm that the **FortimonitorAgent** service status is running. If it is not, start the application.

The FortiMonitor Agent Windows service must be running in order to sync. If the service is stopped, an agent heartbeat incident will occur.

Confirm the Agent Server Key Matches the Server Key of the Affected FortiMonitor Instance

Every agent instance has a server key. The server key of the device you are troubleshooting, should match the server key of the instance shown in the FortiMonitor Control Panel.

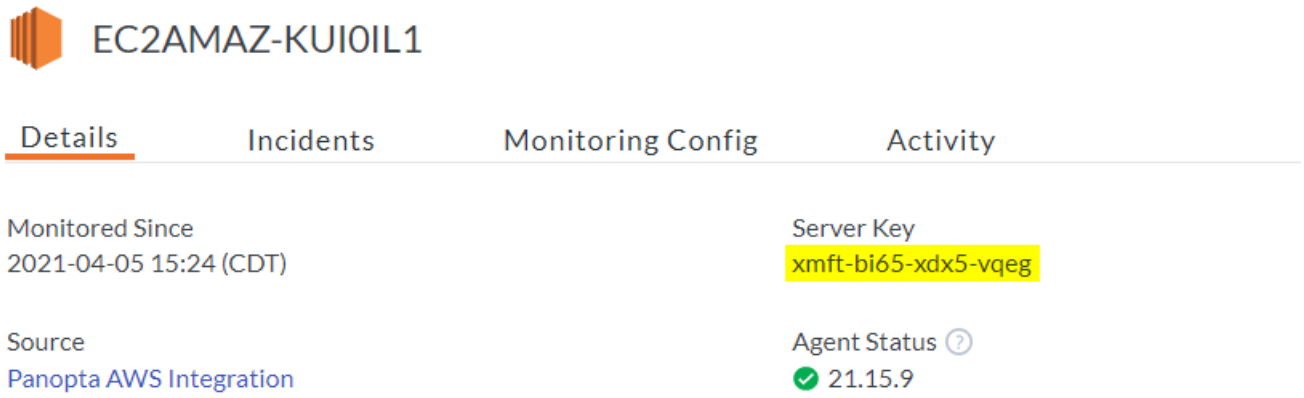
- Method 1: Getting Server Key through Windows PowerShell
  - The agent server key can always be found in the agent.config file for Windows. To get the server key through PowerShell, you can run the following command:
    `Select-String -Path "C:\Program Files (x86)\FortimonitorAgent\Agent.config" -Pattern "ServerKey"`
  - After running the above command you should receive output showing the server key. In the below example, **xmft-bi65-xdx5-vqeg** is the server key.

    ```
    PS C:\Users\Administrator> Select-String -Path "C:\Program Files
    (x86)\FortimonitorAgent\Agent.config" -Pattern "ServerKey"C:\Program Files
    (x86)\FortimonitorAgent\Agent.config:5:    <add key="ServerKey" value="xmft-bi65-
    xdx5-vqeg" />
    ```

Once you have the agent server key, confirm that this key matches the key that you are seeing in the FortiMonitor Control Panel. If the server keys do not match, confirm you are looking at the right server.

If you are looking at the correct server, then you will need to change the server key on the instance to match the key that is found in the **agent.config** file.

Server Key Example as Shown on Server Instance:



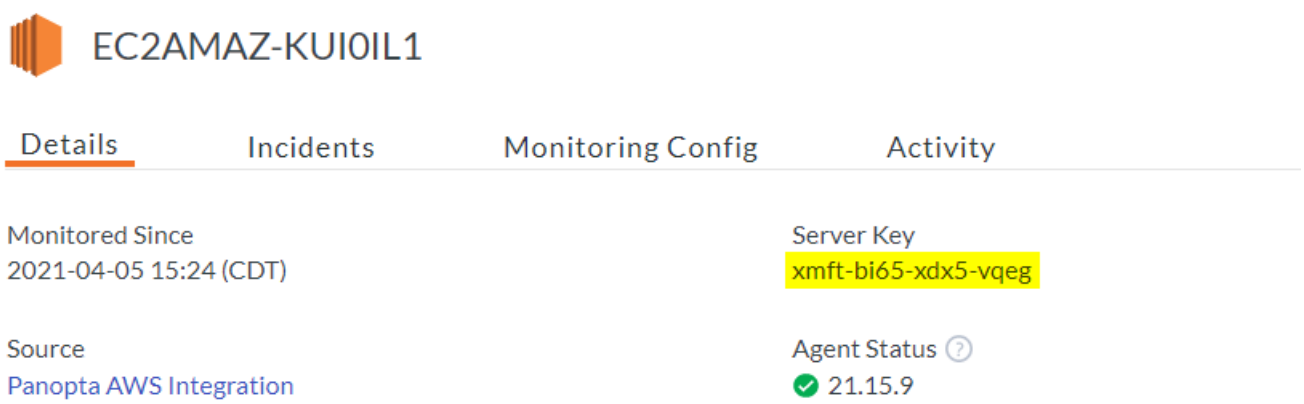
- Method 2: Getting the Server Key Directly From the agent.config File
  - Navigate to the following directory in Windows Explorer: **C:\Program Files (x86)\FortimonitorAgent** .
  - In this directory, you should find the **agent.config** file. Open that file in a text editor to find the server key.
  - The server key should be found towards the top of the file (example below for reference).

```
<agent>  <service>    <add key="AggregatorUrl"
value="https://rx.us01.fortimonitor.com" />    <add key="ServerKey" value="xmft-bi65-
xdx5-vqeg" />    <add key="MaintenanceDuration" value="" />    <add
key="ShowPublicStatus" value="" />    <add key="EnableCounterMeasures" value="" />
<add key="CounterMeasuresRemotePlugins" value="" />    <add
key="CounterMeasuresRefreshPlugins" value="" />    <add key="MetricIncomingDirectory"
value="" />    <!--
```

Once you have the agent server key, confirm that this key matches the key that you are seeing in the FortiMonitor Control Panel. If the server keys do not match, confirm you are looking at the right server.

If you are looking at the correct server, then you will need to change the server key on the instance to match the key that is found in the **agent.config** file.

Server Key Example as Shown on Server Instance:



Verify That Storage Disk is Not at Capacity

- Check that the storage capacity on the server to verify that there is still space available. If the disk is full, the agent will not continue to operate.

If the steps above do not identify the cause of the agent heartbeat problem, please send us a copy of your agent logs. Our team will look at them and follow-up with you.

FortiMonitor Agent Log Location:

```
C:\Program Files (x86)\FortimonitorAgent\logs
```

Agent/Metric not showing up in the Control Panel

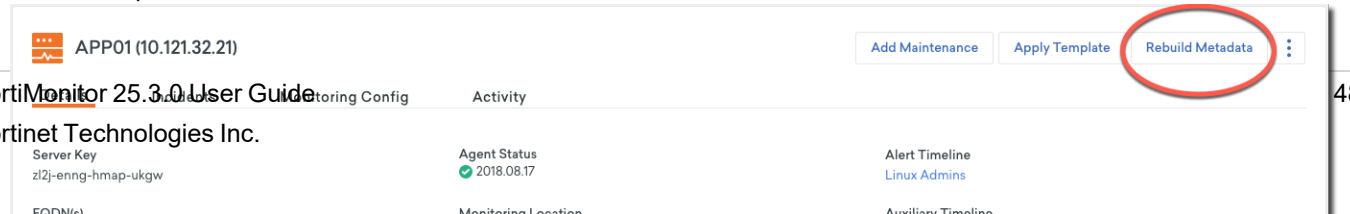
Why does this happen?

Normally the Windows and Linux Agents will sync to your control panel after a few moments once there is enough data gathered for the metrics. This also applies to newly added plugins like the ones for Nginx or Jboss. It is common to wait up to 5 minutes or more for the freshly installed Agent or plugin to show in your Control Panel.

This is caused by the Agent's metadata having not been synced for the first time yet or the information being out of date. In short, metadata is all the information about the plugins and metrics that are available to monitor for that server that can be synced to FortiMonitor. Once installed, our agent 'phones home' about once every minute and rebuilds and refreshes all the cached metadata once per hour. If you recently installed a new monitoring plugin, It is likely that it was recognized but not pulled into your agent metrics. After being added to your plugins folder and having the Agent recognize a new plugin is installed, it does take a few minutes to gather enough data to display metrics.

Using the Control Panel

You can manually tell the Agent to rebuild metadata by clicking the **Rebuild Metadata** button on a server's **Instance Details** page. This is useful if you drop in a new plugin, write your own, or make a change and need it to reload the code as soon as possible.



Linux Users

If you are a Linux user, you can also run the shell version of the Agent to rebuild the metadata with the command below. Keep in mind that there will be no output and the terminal may appear to freeze for up to a minute depending on how many plugins and the metrics that you are monitoring:

For the **panopta-agent**:

```
python /usr/bin/panopta-agent/panopta_agent.py --rebuild-metadata
```

For the **fm-agent**:

```
python3 /usr/bin/fm-agent/fm_agent.py --rebuild-metadata
```

Windows Users

If you are using the Windows agent, you can open your server's services, then right-click and restart the service. Once restarted the agent will automatically rebuild its metadata and sync to our servers.

Automate the FortiMonitor Agent installation

There are a few ways you can automate the installation of the Agent on your servers, you can read this installation article for servers that are not yet in FortiMonitor.

You can also use the Agent Manifest File as a template for Server Agent installation. This is extremely helpful if you also have a template that you would like to apply to a server.

Use the FortiMonitor Agent manifest file

Using the Agent manifest file is a great way to automate the configuration of servers in your FortiMonitor account. If you opt to use the Agent manifest file, you can seed many of the configuration values and have the Agent automatically add the server into FortiMonitor.

The Agent manifest file is only used during the initial installation of the Agent. Subsequent changes must be made using the **FortiMonitor Agent configuration file**. This file can be found in the following directories:

Windows **fm-agent**: **C:\Program Files (x86)\FortimonitorAgent\Agent.config**

The Agent manifest file can also be used in conjunction with the Agent oneliner which normally generates a new manifest file with the server's customer and/or server keys. If you have an existing manifest file on the server, the manifest file will be detected and the information from the Agent onliner will be injected into the existing file. Any settings on the existing manifest file will be preserved.

Windows PowerShell or MSI

Create a manifest file in **C:**, for example **C:\SampleAgent.manifest**. Once this is added, you can set the desired default values. At a minimum, you should add the customer key parameter like below:

Basic Agent Manifest File

```
[agent]
customer_key = xxxx-xxxx-xxxx-xxxx
```

If you are using PowerShell, run the following command from within the same directory as **install.ps1**. The Agent will install and add the server to your FortiMonitor account.

Powershell Install Command

```
./install.ps1 -ManifestFile "[Full File Path]\SampleAgent.manifest"
```

If you are using MSI, run the following command from within the same directory as the `MSI`. The Agent will install and add the server to your FortiMonitor account.

MSI Install Command

```
msiexec /i fm-agent-[version number].msi MANIFESTFILE="[FULL FILE
PATH]\SampleAgent.manifest"
```

Linux

Create the manifest file in **/etc/fm-agent-manifest**. Then you can install the Agent with either the following command:

Linux fm-agent

```
curl -s https://repo.fortimonitor.com/install/fm_agent_install.sh | bash /dev/stdin --
customer-key CUSTOMER_KEY_HERE
```

Example Manifest file

The content of the manifest file for both Windows and Linux are shown below. You do not need to specify values for everything. A detailed description of each parameter is explained below the sample content:

```
[agent]
customer_key = afsdyngoaeppmfqefa
server_key = ashe-pokf-bfhb-eabn
aggregator_url = myappliance.localsite.com
server_group = 3467
fqdn = www.fortimonitor.com
```

```
server_name = FortiMonitor
interface_mapping = private:10.100.100.2,private2:10.100.100.13
templates = 8
tags = tag, anothertag, anotherone
partner_server_ID = 828765
disable_server_match = true
custom_remote_plugins = https://s3.amazonaws.com/custom-panopta-plugins/my-custom-plugins.zip
enable_countermeasures = true
countermeasures_remote_plugins = https://s3.amazonaws.com/some-s3-bucket/custom-plugins.zip
countermeasures_refresh_plugins = 6
[attributes]
operating system = ubuntu
version = 14.04
```

Parameters

Parameter	Description
customer_key	This identifies you with your FortiMonitor account. By setting this value, the agent will automatically add the server to the FortiMonitor control panel. You can find this key in the control panel under **My Account** under your icon at the top right.
server_key	The server key is what allows the Agent to communicate with our servers securely. You have the option of setting this value if you like. However, we highly recommend that you allow a key to be generated for you and linked to the server in FortiMonitor accordingly. **Note**: all server keys must be unique.
aggregator_url	This controls the API which the Agent communicates with to report its results. You should not enter an aggregator URL unless you are using FortiMonitor Onsight as a proxy for servers that do not have outbound public internet access. If you are using FortiMonitor Onsight as a proxy and don't know the aggregator URL, please contact our support team.
server_group	This controls which group your server gets added to in the control panel. Any server template(s) that you have set to be applied to this group will be applied to this server as well. The value of the server group is identified by an ID or a name. If you go to an instance group in the control panel, you can see that group's ID number at the end of that page's URL.
FQDN	This field acts as an override for the hostname that is auto-detected by the Agent. If you do not include this field, your server will be given its natural FQDN in the control panel.
server_name	This field lets you set a logical name for this server. That name will identify the server in the control panel.
interface_mapping	This value is only used if: • A template will be auto-applied to the server being created

	• that template has place holders for multiple network interfaces. If your server has multiple network interfaces and the template which is being applied has placeholders for multiple network interfaces, you have to provide a mapping of the IP for that network interface to the placeholder. For example, if your server template has the network interface placeholder named Private, and your server has a network interface with the IP set to 10.100.100.2, you would configure it like this: **interface_mapping = private:10.100.100.2**
templates	This field lets you choose which templates you would like to apply to this server. The value of the template is identified by an ID. If you go to a server template in the control panel, you can see that template's ID number at the end of that page's URL.
tags	This field would allow you to add tags to your server. This can be set to any value but would need to be comma-separated to indicate different tags. **tags = tag1, tag2, tag3**
partner_server_id	This value can be used by FortiMonitor partners in order to store their own internal unique identifiers for that server. Setting the value in the manifest file sets it on the Server/Instance object which can then help in server lookups by the partner's ID (via API or within the control panel).
disable_server_match	The default behavior, when **customer_key** is provided, is to do some automatic server matching by hostname or IP. In some cases, this may be undesirable if you have overlapping private IP ranges across environments. This option allows you to disable the matching logic.
attributes	You can also add an additional field for attributes. This will need to go underneath the rest of the manifest file configurations, and you can set the attributes to any value.
custom_remote_plugins	If you have written your own custom plugins, you can host them centrally and have the agent retrieve them during installation so that you do not have to manually manage them post-install. The value of this variable can be a compressed archive (.zip, .tar or .tgz) or a single python file (with a .py extension). New plugins are fetched from the URL defined in **custom_remote_plugins** during metadata rebuilds. By default, metadata rebuilds occur once every hour. You also have the option to manually re-fetch new plugins by issuing a metadata rebuild command, either through the control panel or the command line. **Additional notes for the Windows Agent** When using **custom_remote_plugins** to remotely pull and install custom plugins: • Ensure that the zip file only contains **ps1** files for the Windows Agent. • Set the parameter **countermeasures_refresh_plugins = 1**. Using this parameter in an existing Windows Agent installation requires a service restart.
enable_ countermeasures	This flag enables the **CounterMeasures** remediation feature in the Agent. Read more about this here.

countermeasures_ remote_plugins	If you have written your own custom **CounterMeasures** plugins, you can host them centrally and have the Agent retrieve them during installation so that you do not have to manually manage them post-install. The value of this variable can be a compressed archive (.zip, .tar or .tgz).
countermeasures_ refresh_plugins	If you would like updates to your **CounterMeasures** plugins to automatically be applied, you can optionally set **countermeasures_refresh_plugins** to the number of hours for the refresh interval and the Agent will re-fetch the newest plugins from your remote plugin location.
plugin_configuration	To pass plugin configuration to a standard FortiMonitor plugin or one of your own custom plugins, specify those at the bottom of the file. Examples for Linux and Windows are shown in the following sections.
force_thread_culture	Set the current culture to English. To enable, add the following parameter: **force_thread_culture = true**
auto_update	**Windows fm-agent only** Enables the Windows Agent auto-update feature. To enable, add the following parameter: **auto_update = true**
scheduled_update	**Windows fm-agent only** Set a specific time for the Agent to update. This should be in military-time format; for example, 2:09 PM would be set as 14:09. The agent will query/attempt an update at this time. If this flag is not used, the Agent will select a random time within 24 hours of its startup to check for a download; this should be logged in the Agent log.

Linux custom plugin configuration

Manifest file

```
[agent]
customer_key = gfoadsvuhyadsflvhea
server_group = 3548
[attributes]
operating system = ubuntu
version = 14.04
[mysql]
username = USERNAME
password = PASSWORD
```

Windows custom plugin configuration

Manifest file

```
[agent]
    customer_key = gfoadsvuhyadsflvhea
    server_group = 3548

[attributes]
    operating system = ubuntu
    version = 14.04

<agent>
    <plugins>
        <ntp>
            <add key="ntp_host" value="pool.ntp.org">
                <add key="ntp_port" value="123">
                </add>
            </add>
        </ntp>
    </plugins>
</agent>
```

Remove an instance automatically (FortiMonitor Agent)

You can use the Agent Manifest File to automate the addition of instances into FortiMonitor upon the installation of the Agent on a server. This is written up in detail here. This, in conjunction with Templates, allows you to achieve a fully hands-off configuration experience. You can factor this into whichever server provisioning/automation mechanism you have in place.

In a dynamic auto-scaling environment, you may want to be able to automatically remove those instances from FortiMonitor as well. This can be done by initiating the uninstall with a special parameter. This will deregister the agent in Panopta (to avoid any alerts) as well as remove the instance from your Panopta account. When the server is going to be scaled down and removed, initiate the following commands based on your operating system:

```
sudo python /usr/bin/fm-agent/fm_agent.py --uninstall --remove-instance
```

For Linux

This feature is only available on newer versions of the Agent. If you are running an older version, upgrade the Agent first.

Custom metrics and incidents

In addition to the large catalog of out-of-the-box monitoring FortiMonitor provides, you also have the ability to collect your own custom metrics which may be specific to your own internal systems and applications. FortiMonitor provides several mechanisms for collecting custom metrics.

All methods, except for API, requires the FortiMonitor Agent.

Native agent plugin

Writing a native agent plugin allows for the most flexibility and allows for the agent to fully handle all elements of data collection. For Linux systems, plugins are written in Python. For Windows systems, plugins are written in PowerShell. Sample agent plugins to use as a starting point are provided for both platforms.

Command line calls

If you prefer to push metrics into the agent, you can register and report metrics from a command-line script. You can invoke this script from your own custom program or from a cron/scheduler. This method allows for less flexibility but can be very easy to bring to life.

JSON-based custom metrics

You can also push custom metrics via JSON files through the Agent. You can have an external program generate files for each custom metric value that you want to have aggregated for alarming within FortiMonitor.

JSON-based custom incidents

You can generate Custom incidents using JSON files that allows you to push event data to FortiMonitor. To use custom incidents, create a JSON file each time there is an incident to report and add the file to a special agent directory.

API

You can also create Custom incidents using API, which provides a mechanism for your own applications to push event data to our system and leverage FortiMonitor's alerting and dashboard functionality.

Custom Metric Management

FortiMonitor enables you to publish custom metrics in a variety of formats across our supported platforms. As well, you can manage the meta-data for your custom metrics once they've been reported to FortiMonitor - including name, category, unit, and type. This metadata is used throughout FortiMonitor - in graphs, incident alerts, threshold configuration, reports, and more.

You can learn more about it here.

Custom metric management

FortiMonitor enables you to publish custom metrics in a variety of formats across our supported platforms. As well, you can manage the meta-data for your custom metrics once they've been reported to FortiMonitor - including name, category, unit, and type. This metadata is used throughout FortiMonitor - in graphs, incident alerts, threshold configuration, reports, and more.

Using custom metrics is powerful but can also be a bit intimidating at first - if you need some assistance or input, please contact our support team.

Custom Metric Management

Custom Metric Workflow

A quick reminder of the workflow for custom metrics.

1. Define the custom metric using any of the following options:
   - command line
   - JSON files
   - custom Linux plugins
   - custom PowerShell plugins
     See Custom Metrics and Incidents.
2. Report your first metric value
3. We'll use the metadata you defined in your custom metric to stamp out the initial Custom Metric record in FortiMonitor
4. You can edit / override any of the settings identified below from the Custom Metric management page

Editing Custom Metric Metadata

You can view your reported custom metrics on the **Custom Metric** management page, which is located under **Monitoring** > **Advanced Metrics** navigation item. To edit the metadata about one of your metrics, locate it in the **Custom Metric** table, and then click the corresponding **3-dot menu** in its row and click **Edit**.

Fields

Field	Description	Editable?
Textkey	The Plugin and Metric Textkeys you defined when creating your custom metric locally.	No
Name	The name of your metric - this will appear throughout the Control Panel, including in dashboards, alert descriptions, graphs, etc	Yes
Category	You can think of the Category as a way to logically group related custom metrics. Our built-in categories are things like SQL Server, MySQL, Apache. This will also dictate how metrics are grouped together in the monitoring configuration catalog on your instances	Yes
Unit	The unit for your metric - transactions, orders, users, etc. Metrics with the same unit are grouped together in graphs	Yes

Metric Type	Number, boolean, or percent	Yes
Boolean Values	If your metric is a boolean type, you can set the **True** and **False** states - such as is running and is not running. They map to the 1/0 values you report. This will appear in metric descriptions and when configuring alerts.	Yes
Option	Provides context for the object type you're reporting on - such as **drive** or **network adapter**.	No

Custom Metrics in Templates

Once a Custom Metric has been reported-up for the first time, it will be available to add to your templates. Similar to the monitoring provided out of the box, it will be available in the Monitoring Catalog on a template page. Merely select `Add Monitoring` from the monitoring configuration table and locate the Custom Metric category in the catalog.

Custom metrics using the command line (FortiMonitor Agent)

Custom metrics allow you to report your own metrics straight to the agent with a command-line call. This process is simpler than creating a custom agent plugin and is a great way to add new functionality to our agent.

The way to create a custom metric is is a two-step process. First, you will have to create a custom metric using the registration script below. **Your metric key can be any value containing letters, numbers, periods, underscores, or dashes**.

Linux

```
python3 /usr/bin/fm-agent/register.py <metrickey> <description>
```

If this script is not found in the fm-agent folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

Windows

```
PS C:\Program Files (x86)\FortiMonitorAgent\bin> .\register.ps1 <metrickey>
<description>
```

If this script is not found in the folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

You will only need to do this once. Then, whenever you would like to report a value, you can use the following command line call:

Linux

```
python /usr/bin/fm-agent/report.py <metrickey> <value> <timestamp>
```

Windows

```
C:\Program Files\FortimonitorAgent\bin> .\report.ps1 <metrickey> <value> <timestamp>
```

The value must be an integer or a float, and the timestamp is an optional local time with ISO format (YYYY-mm-dd HH:mm:SS"). If the timestamp is left off it will default to the current local time time.

You can make any number of report calls for a given metric key. However, the system will record only the last value submitted in a given minute, and discard any previously submitted metric readings in the same minute.

You can now create you own scripts to reports these metrics automatically or integrate calls to the agent into your own application. Once you have registered your new metric, an agent resource type will be created for this server automatically in the control panel. You can edit this check as you would any other. You can create as many additional custom metrics as you like and they will all show up under the "Custom Metric" category.

If you want to verify that **report.py** is collecting those metrics, review the generated log entries.

If you have followed the steps above and the new metric and plugin do not show properly, you may need to manually rebuild the agent's metadata. You can do this by following the guide here.

Custom plugins for Linux (FortiMonitor Agent)

You can add your own custom plugins to the agent if there are any resources on your server that you would like to monitor, but that we do not currently support. For example, if you have created a custom application for your server and would like to check its performance then you can write a custom plugin for that application.

1. Create a new python file in **/usr/share/fm-agent/** directory named after the applications you're monitoring, such as **my_application.py**.
2. Start with a copy of the **template.py file**, which defines the overall layout of the plugin class and the methods necessary to operate. Each of the comments in this file marked with **TODO:** is an item that needs to be provided for your plugin to work.
3. Define a class attribute **textkey** for your plugin. This is a unique dotted string defining the very general service or resource that your plugin is monitoring. The textkey should be prefixed with an identifier for your organization. For example, **com.acme.myapplication**.
4. Implement **get_metadata()**. This is the method that determines what aspects of the thing you're checking are able to be monitored. For example, the standard Apache plugin has checks in it to determine if ExtendedStatus is available in **/etc/httpd.conf**, which allows the plugin to monitor things like connections per second. Without this enabled, **get_metadata**, can only say that it can monitor a few basic aspects of Apache. It takes 1 argument ""config"" which is your plugin's specific configuration. More on configuration later. The return value from this method is a dictionary whose keys are unique, monitorable aspects of the service (**resource textkeys**) which are different from the plugin textkey), and whose values are dictionaries with the following keys:
   - **label**: A human-readable string description of the metric, used for graph labels and alert messages in the control panel.
   - **options**: An optional list of strings describing different metric sources that could be monitored. For example, for the disk space plugin, the options are a list of all of the disk partitions on the server. If your metric doesn't have any options, you can set this to None.
   - **status**: One of **agent_util.SUPPORTED**, **agent_util.UNSUPPORTED** or **agent_util.MISCONFIGURED**, to indicate that the metric can be collected, is unavailable, or available but missing some configuration that is preventing the plugin from measuring the metric value.
   - **error_message**: For misconfigured metrics, this message will be shown in the control panel to provide users additional information needed to properly fix the configuration issues.
   - **unit**: A text string description of the units of measurement for the metric, for example, **MB** or **connections per second**. Metrics from a plugin that share a common unit will be graphed together in the control panel. Normally, if you're developing your own plugin, you probably won't have to do much in the way of making it portable for others, since you're assuming that your service is already available and monitorable, but if you

need more flexibility in detection, get_metadata is where you'd exercise it.

- **option_string**: This will allow the check to be created with an open value field in the Control Panel that will be passed down to the agent on execution. This option can be any value you want, from a file name to a port number.

5. Implement the **check()** method. The purpose of this method is to actually measure the thing you want to monitor and return its value. It takes 3 arguments: **textkey**, **data**, and **config**. Textkey is one of the resource textkeys in the dictionary's keys you returned from **get_metadata**. It defines the specific aspect of the service you wish to monitor. Data is one of the items from the **options** list you returned in the **get_metadata()** (or None if no option is selected). Config is your plugin's specific configuration, the exact same object passed into **get_metadata**.

   Check() should be the bulk of your work. This may include operations such as slicing, aggregating, monitoring, or comparing values as needed. The method should return a measurable numeric result, either an integer or a float, that represents the computed metric.
   If you included **option_string**, the value you defined on the check will be passed on to the agent as the data parameter. You can then proceed to use this value as your custom plugin needs.

Advanced Techniques

Using result data between multiple calls of **check()** (for example, for creating amount-per-second averages) can be accomplished by using the **cache_result** and **get_cache_results** methods from within **check()**. **cache_result** takes 3 arguments: the resource textkey and the data being passed in to **check()** (for example, the net interface to monitor) and a result that you want to store. **get_cache_results** takes 2 required arguments: the resource textkey, the data, and 1 optional argument: the number of results to return. It returns a list of cached results (default: 1) of the format:

```
(delta, result)
```

Where delta is how many seconds ago the result was retrieved, and the result is the result stored when cache_result was called. Take a look at the bandwidth plugin to see how these method calls can be used to create and return kilobytes/second transfer rates.

Automating Deployment

If you have a large infrastructure footprint and want to automate the deployment and handling of custom plugins across different servers, you can set your custom plugins to be automatically fetched during installation. This saves you from having to manage this separately. More information on this can be found in Use the FortiMonitor Agent manifest file.

template.py file

```
import os
try: import json
except ImportError: import simplejson as json
import agent_util
import datetime

    # TODO: Rename class to an appropriate name, must end in "Plugin"
    class TemplatePlugin(agent_util.Plugin):

    # TODO: Replace textkey and label with appropriate values
    textkey = "template_textkey"
```

```
    label = "Template Name"
    @classmethod
    def get_metadata(self, config):
        status = agent_util.SUPPORTED
        msg = None
        # TODO: Check that the plugin can run successfully and has all of the necessary
        # prerequesites. If not, set status to agent_util.MISCONFIGURED and set a message
        # to describe what is wrong.
        if not True:
            status = agent_util.MISCONFIGURED
            msg = "Status message"
        # TODO: add dictionaries for each metric that can be measured. Each key in the
        # metadata dictionary is a textkey for that metric (all lower case letters, numbers
        # and underscores, no spaces or special characters)
        metadata = {
            "metric1": {
                "label": "Description of metric",
                "options": None,
                "status": status,
                "error_message": msg,
                "unit": "unit description"
            },
            "metric2": {
                "label": "Description of metric",
                "options": None,
                "status": status,
                "error_message": msg,
                "unit": "unit description"
            }
        }
        return metadata
def check(self, textkey, data, config):
    # TODO: Add code to perform the actual checks for the metric described by textkey.
    # If options are avaiable for the metric and one is selected it will be included
    # in the data parameter. Return value should be an integer or floating point value.
    if textkey == 'metric1':
        return 1
    elif textkey == 'metric2':
        return 2
    return 0
```

Custom plugins for Windows (FortiMonitor Agent)

You can add your own custom PowerShell plugins to the agent if there are any resources on your server that you would like to monitor, but that we do not currently support out of the box or where there is no PerfMon counter. For example, if you have created a custom application and would like to check its performance/functionality then you can write a custom plugin for that. Here are the steps:

1. Create a new file in the **%Program Files%/FortiMonitorAgent/ps_plugins** directory named after the applications you're monitoring, such as **my_app_plugin.ps1**.
2. Start with a copy of the **template.ps1 file**, which defines the overall layout of the plugin class and the methods necessary to operate.
3. Implement **Plugin-Configuration**. Define a value for the variable **textkey** for your plugin. This is a unique string defining the very general service or a resource that your plugin is monitoring. The textkey should be prefixed with an identifier for your organization. For example, **com.acme.myapplication**.
4. Implement **Discover**: This is the method that determines what aspects of the application/metric you're checking are able to be monitored. This returns an array of dictionaries that contain the metrics and the options the plugin can effectively monitor. Options can be static values or could be dynamically discovered based on the environment/server. Normally, if you're developing your own plugin, you probably won't have to do much in the way of making it portable for others, since you're assuming that your service is already available and monitorable, but if you need more flexibility in detection, Discover is where you'd exercise it. Each object in the array has an attribute and those attributes are explained below:
   - **label**: A human-readable string description of the metric, used for graph labels and alert messages in the control panel.
   - **options**: An optional list of strings describing different metric sources that could be monitored. For example, for the disk space plugin, the options are a list of all of the disk partitions on the server. If your metric doesn't have any options, you can set this to **$null**.
   - **unit**: A text string description of the units of measurement for the metric, for example, "MB" or "connections per second". Metrics from a plugin that share a common unit will be graphed together in the control panel.
5. Implement **Collect-Metric**: This is responsible for actually doing the retrieval/calculation based on the metric textkey, an optional value, and other optional config parameters that you may have specified within the Agent's configuration file. Collect-Metric should be the bulk of your work. It should read, parse, chop, measure, compare, analyze, aggregate, monitor, slice, dice, mix, and mash whatever required to return a measurable value of your resource, and ultimately return an integer or float value for the metric. This function takes in two parameters. The **$resource_textkey** is a reference to the metric type which is being passed in from the agent to be collected. If your plugin supports multiple metrics (returned by the Discover function), you will need some branching logic here to execute different logic based on the resource_textkey. $**option** is a reference to the specific option value for that metric. This will influence the specific metric value you need to query/extract and return.

Plugin Configuration

If your plugin requires additional configuration to account for settings or environment-specific variables, you can leverage the Agent's configuration file. Plugin configuration is defined in your Agent's main XML configuration file, under its own section titled after the textkey of the plugin. To define your plugin's section, create a new XML block under config with the node matching the textkey from Plugin-Configuration. Underneath that new section, you can put different key/values that are better not hard-coded into your Collect-Metric method--things like login credentials and other adjustable parameters. An example would be:

Plugin Config

```
<config>
    <systemp>
        <add key="myparam" value="myvalue"/>
    </systemp>
</config>
```

When your plugin's Discover and Collect-Metric methods are called, the Agent configuration file is parsed and searched for a plugin config section that matches the name of your plugin class. If any config is found, the variables and values are passed into the plugin functions.

template.ps1 file

```
function Plugin-Configuration {
    return @{
        textKey = "systemp"
        category = "Get System Temperature"
    };
}

function Discover {
    return @(
        @{
            resource_textkey = "temp_in_F"
            label = "Temperature Fahrenheit"
            unit = "degrees F"
            options = @("fan", "intakeFan", "exitFan")
        },
        @{
            resource_textkey = "temp_in_C"
            label = "Temperature Celsius"
            unit = "degrees C"
            options = @("fan", "intakeFan", "exitFan")
        }
    )
}

function Collect-Metric($resource_textkey, $option) {
    return Get-Temperature $resource_textkey;
}

function Get-Temperature($scale) {
```

```
$t = Get-WmiObject MSAcpi_ThermalZoneTemperature -Namespace "root/wmi"

$currentTempKelvin = $t.CurrentTemperature / 10;
$currentTempCelsius = $currentTempKelvin - 273.15;

if ($scale -eq "temp_in_F") {
    return (9/5) * $currentTempCelsius + 32;
}

if ($scale -eq "temp_in_C") {
    return $currentTempCelsius;
}

return $currentTempKel
```

Create custom incidents

FortiMonitor supports generating custom incidents, which provides a mechanism for your own applications to push event data to our system and leverage FortiMonitor's alerting and dashboard functionality. To use custom incidents, you need to have the latest version of the Agent.

Create a custom incident

To create custom incidents, perform the following steps:

1. Go to the Instance Details page of the instance where you want to create a custom incident.
2. Select the **Custom Config** tab.
3. Click **Add Incident Type**.
4. Fill in the required information. Refer to the following table below.

Field	Description
Plugin Textkey	Textkey which specifies the application category the incident relates to. For example: `com.mysite.network`
Resource Textkey	Textkey specifying the specific metric for this incident type. Must be letters, numbers, period and underscore only
Category	Name for the category that this incident type belongs to.
Name	Name for the metric that this incident type belongs to.
Severity	The **severity** of the outage/anomaly, either critical or warning.
Alert Timeline	The Alert timeline to send notifications to.

5. Click **Create Incident Type**. The custom incident will be created and will be displayed in the **Custom Incident** table.

Edit a custom incident

To change the configuration of a custom incident, perform the following:

1. Click the **3-dot menu** option then select **Edit**.
2. Modify the configuration as needed then click **Edit Incident Type** to apply the changes.

Delete a custom incident

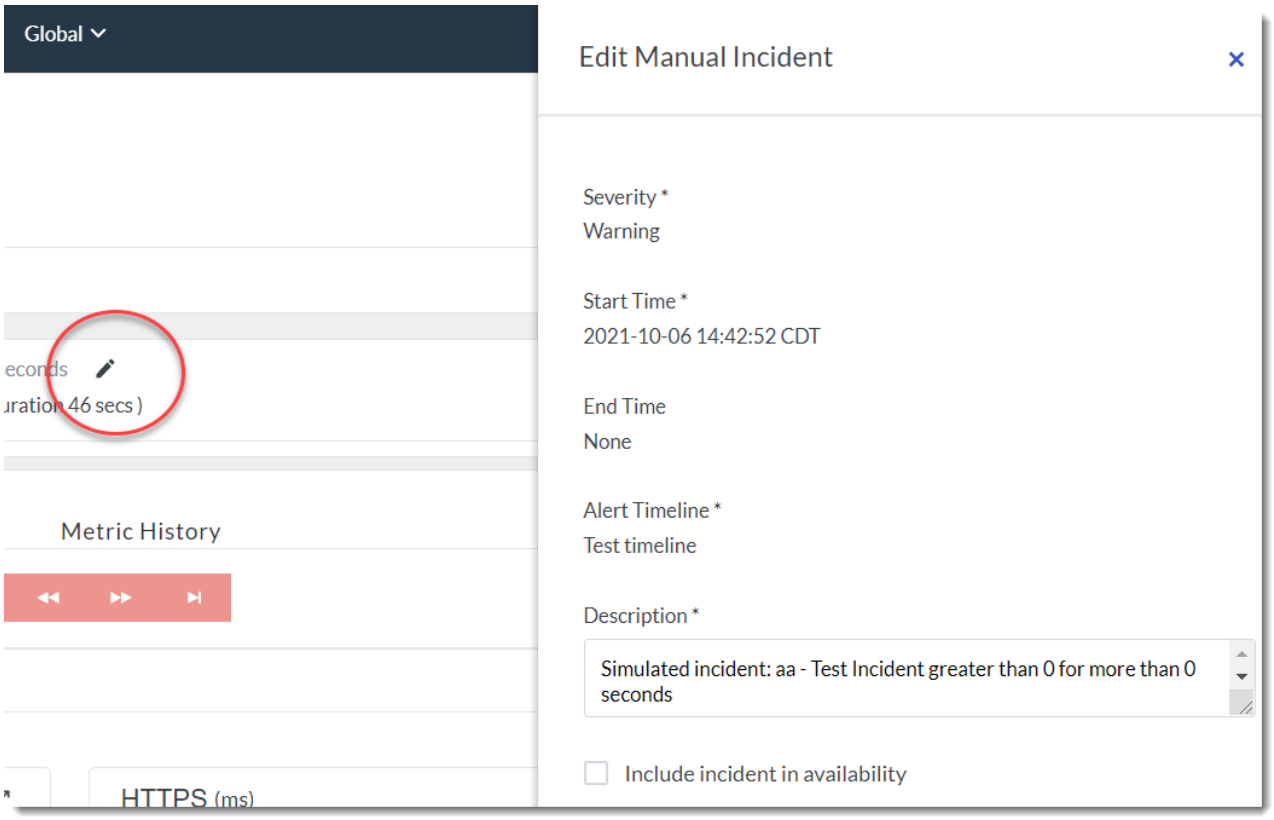
To delete a custom incidents, perform the following:

1. Click the **3-dot menu** option then select **Delete**.
2. Click the **I'm sure** button to confirm. The custom incident will be deleted from the **Custom Incident Config** table.

Edit incident description

Custom incident descriptions can be edited to provide more context about the incident or simply modify an existing description. This allows you to change the description of a custom incident without deleting and recreating it.

To edit the description of a custom incident, perform the following:

1. Open an incident's details page.
   Click the **Pen** icon. The **Edit Manual Incident** drawer slides out.
2. 

3. Edit the the description then click **Save**.

Custom incidents using JSON files (FortiMonitor Agent)

The FortiMonitor Agent supports generating custom incidents, which provides a mechanism for your own applications to push event data to our system and leverage FortiMonitor's alerting and dashboard functionality. To use the custom incidents, you need to have the latest version of the Agent and have custom incidents enabled on your account. Please contact our support team to have that enabled.

To use the custom incident support, you create a JSON file each time there is an incident to report and place it in a special agent directory. The next time the agent syncs, it picks up that file and sends it to the FortiMonitor Monitoring Cloud where it is processed and begins alerting.

Incident File Generation

The incident files are JSON files that get placed in the following directories:

Linux

/usr/share/fm-agent/incoming

Windows

C:\Program Files (x86)\FortimonitorAgent\custom_metric

Additional directories can be specified by setting a **metric_incoming_directory** variable with a comma-separated list of directories in your Agent configuration file (Linux only) located in **/etc/fm-agent/panopta_agent.cfg**.

Each file should look like the following:

```
{
  "agent_payload_version": 1.0,
  "plugin_textkey": "sample_custom_plugin",
  "plugin_category_name": "Sample Application Plugin",
  "incidents": [
    {
      "textkey": "billing.batch_error",
      "timestamp": "2017-08-17 10:50:00",
      "description": "Error processing billing batch #23",
      "action": "open",
      "metadata": {
        "lines_processed": 100,
        "error_count": 27
      }
    },
    {
      "textkey": "billing.batch_error",
      "timestamp": "2017-08-17 11:50:00",
```

```
      "description": "Error processing billing batch #24",
      "action": "resolve",
      "match_key": "billing-batch-24",
      "metadata": {
        "lines_processed": 100,
        "error_count": 27
      }
    }
  ]
}
```

The top-level keys in the document set the agent version, which should be left as 1.0, and the plugin textkey and category name. The textkey is used to reference this application internally, and can be any string of letters, numbers, periods, and underscores. The plugin category name is used for display in the control panel's metric catalog and for alert descriptions.

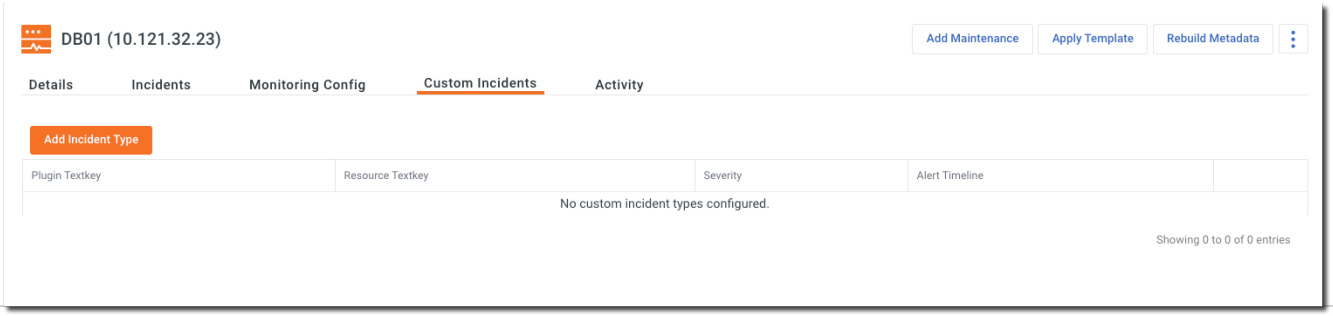
The incidents list contains a separate entry for each incident that you wish to report. The keys in each entry are:

- textkey: an internal reference for the resource the incident applies to. Follows the same character convention as the plugin textkey.
- timestamp: The time of the event is starting or being resolved. The timestamps are in **YYYY-MM-DD HH:MM:SS** format, and need to be in the UTC timezone.
- description: A human readable description of the event, which is shown in the control panel and alerts.
- action: A value of **open** or **resolve**, which indicates if the event is being created or resolved.
- match_key: An optional field, if provided it is used to connect open and resolve events that relate to the same issue. These are arbitrary strings, so they can be any identifier that makes sense for your application. If not provided, incidents can be closed through the control panel or will automatically get closed after 60 minutes.
- metadata: An optional dictionary of key/value fields that will get added to the incident as additional data that is then shown in the control panel. Use this to pass in any additional details that your users will want to see when diagnosing the issue.

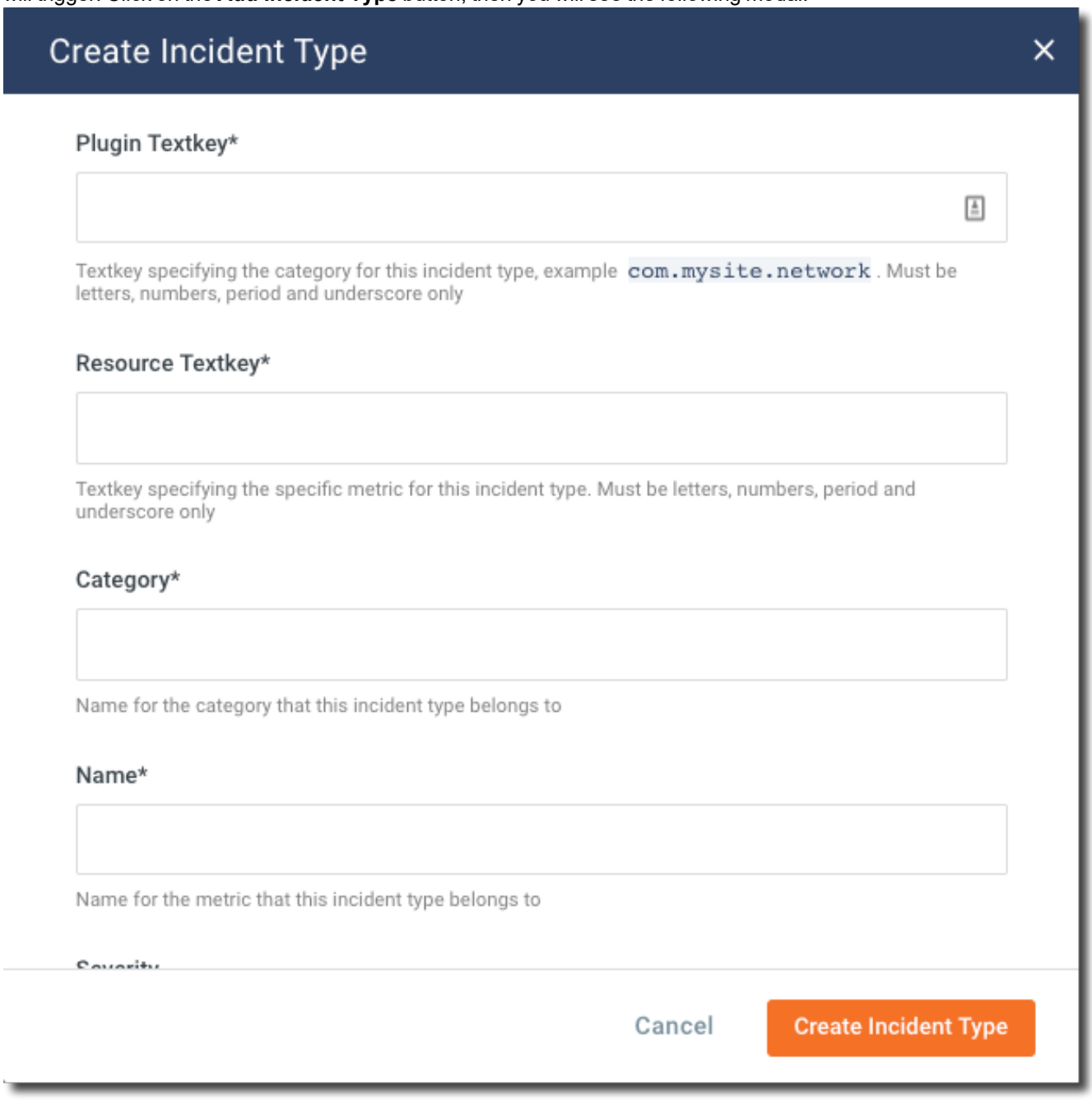
You can bundle together multiple incidents into one file, or create each one as a separate file. The agent will pick up and process each file found in the import directory. We recommend naming files based on the application name and current timestamp to avoid running into conflicts with previous files that are waiting to be picked up.

Control Panel configuration

Before you can start sending up custom incidents, you need to enable support on the server whose agent will be sending up the incident data. Navigate to the server's **Instance Details** page in the control panel, where you should see a **Custom Incidents** tab:



Here you will need to set up filtering rules that map incident plugin and resource textkeys to the type of alerting that they will trigger. Click on the **Add Incident Type** button, then you will see the following modal:



Enter the plugin textkey you are using in the JSON files, and optionally specific resource textkeys (either as exact matches or using a * for partial matches. Once the filter is in place, any custom incident data that is sent through the agent will create incidents that will appear in your dashboards and begin sending alerts using the specified timeline.

As a quick way to get started, you can create a single filter that matches all resource textkeys, and later can refine this with more specific resource textkey filters if you wish to use different alert timelines for different type of incidents.

Custom metrics using JSON files

The FortiMonitor Agent supports generating custom metrics, which provides a mechanism for your own applications to push custom data to our system and leverage FortiMonitor's alerting and dashboard functionality. To use the custom metrics, you need to have the latest version of the Agent.

To use the custom metric support, you create a JSON file each time there is a metric result to report and place it in a special agent directory. The next time the agent syncs, it picks up that file and sends it to the FortiMonitor Monitoring Cloud where it is processed and begins alerting.

Metric File Generation

The metric files are JSON files that get placed by default in the following directories:

Linux

/usr/share/fm-agent/incoming

The default permissions for the directory **/usr/share/fm-agent/incoming** is **755**. This means that the permissions may need to be changed if the metric collection is run by a user other than **fm-agent**. If you modify the permissions, ensure that **fm-agent** is able to read from and write to that directory.

Windows

C:\Program Files (x86)\FortiMonitorAgent\custom_metric

Additional directories can be specified by setting a metric_incoming_directory variable with a comma-separated list of directories in your agent configuration file (Linux Only) located in **/etc/fm-agent/panopta_agent.cfg**.

On Windows, you can provide a single path in the MetricIncomingDirectory entry in the agent config file, Agent.config, which is located in the Agent directory. Make sure to also restart the FortiMonitor Agent service after saving.

Each file should look like the following:

```
{
  "agent_payload_version": 1.0,
  "plugin_textkey": "sample_custom_plugin",
  "plugin_category_name": "Sample Application Plugin",
  "metrics": [
    {
      "textkey": "database_connection",
      "label": "Database is accessible",
      "value": 1,
      "unit": "boolean",
      "timestamp": "2017-08-17 10:53:00"
    },
    {
      "textkey": "generated_lines",
      "label": "Number of lines in batch file",
      "value": 800,
      "unit": "lines",
      "timestamp": "2017-08-17 10:53:00"
```

```
    }
  ]
}
```

The top-level keys in the document set the agent version, which should be left as 1.0, and the plugin textkey and category name. The textkey is used to reference this application internally, and can be any string of letters, numbers, periods, and underscores. The plugin category name is used for display in the control panel's metric catalog and for alert descriptions.

The metric list contains a separate entry for each metric result that you wish to report. The keys in each entry are:

- textkey: an internal reference for the resource the metric applies to. Follows the same character convention as the plugin textkey.
- label: a human readable label for the metric, which is shown in the control panel and alerts.
- value: the actual numeric metric values associated with the metric.
- unit: the unit of measurement for this metric. `boolean` and `percentage` are recognized unit types which will cause special behavior in the app
- timestamp: The time associated with the metric result. The timestamps are in `YYYY-MM-DD HH:MM:SS` format, and need to be in the UTC timezone.

You can bundle together multiple metric results into one file, or create each one as a separate file. The agent will pick up and process each file found in the import directory. We recommend naming files based on the application name and current timestamp to avoid running into conflicts with previous files that are waiting to be picked up.

Application plugins

The FortiMonitor server agent is bundled with a wide variety of plugins which allow you to monitor various operating system and application performance health metrics. Each plugin has its own optional configuration requirements that need to be set in the Agent's configuration file or on the server itself.

Available plugins are detailed in the following sections.

Active Directory

The Active Directory plugin allows you to monitor your Active Directory service.

Control Panel Configuration

Select the **Active Directory** plugin from the monitoring catalog. To add this resource, please see Add FortiMonitor Agent checks.

The available metrics are as seen below:

Metric	Unit
DRA Inbound Bytes Compressed (Between Sites, Before Compression)/sec	bytes/sec
DRA Inbound Bytes Total/sec	bytes/sec
DRA Inbound Objects/sec	objects/sec
DRA Inbound Values Total/sec	bytes/sec
DRA Outbound Bytes Total/sec	bytes/sec
DRA Outbound Objects/sec	objects/sec
DRA Pending Replication Synchronizations	synchronizations
DS Client Binds/sec	binds/sec
DS Directory Reads/sec	reads/sec
DS Directory Searches/sec	searches/sec
DS Directory Writes/sec	writes/sec
LDAP Bind Time	ms
LDAP Client Sessions	sessions
LDAP Searches	searches
LDAP Searches/sec	searches/sec
LDAP Successful Binds/sec	binds/sec
LDAP Writes	writes
LDAP Writes/sec	writes/sec

If you have any further questions, please contact our support team.

Apache Kafka (Linux)

The Apache Kafka (Linux) plugin allows you to monitor and collect Kafka performance metrics using JMX.

Enable JMX on the Kafka cluster

To enable JMX on the Kafka cluster, perform the following steps:

1. Add the following to the **KAFKA_JMX_OPTS** variable of the startup script being used. By default, this file is called **kafka-run-class.sh**.

```
-Dcom.sun.management.jmxremote.port=<MY_JMX_PORT> -Dcom.sun-
.management.jmxremote.rmi.port=<MY_JMX_PORT> -Djava.rmi.serv-
er.hostname=<MY_IP_ADDRESS>
```

Where:

My_JMX_Port is the JMX port of the Kafka host.

My_IP_ADDRESS is the IP address of the Kafka host.

2. Make sure that you have the latest version of the Agent. In the **/etc/fm-agent/fm_agent.cfg** file or the manifest file, make sure that the Kafka JMX information is filled in as follows:

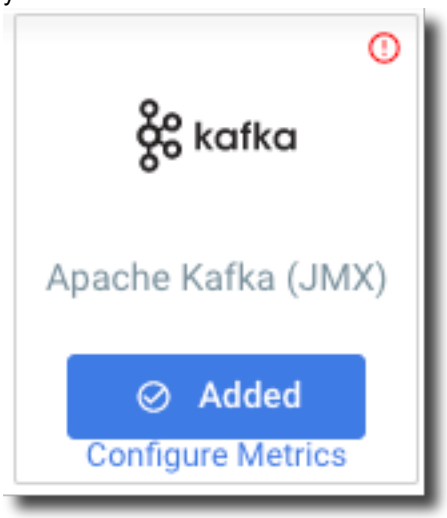
```
[apache_kafka_jmx]
host = <MY_IP_ADDRESS>
port = <MY_JMX_PORT>
```

3. Rebuild the FortiMonitor Agent metadata.

We currently do not support authentication so we highly recommend only allowing JMX on a local IP (like 127.0.0.1 or localhost)

Control panel configuration

Apache Kafka monitoring is added to a server from the **Monitoring configuration tab** of the **Instance Details page**. If you do not see Kafka as a discovered resource, rebuild the FortiMonitor Agent metadata.



Available metrics

Metric	Unit
Heap Memory - Committed	Bytes
Heap Memory - Max	Bytes
Heap Memory - Used	Bytes
Queued messages	Count
Replica Manager Unreplicated Partitions	Count
Topic Bytes In/sec	Bytes per second
Topic Bytes Out/sec	Bytes per second
Topic Messages In/sec	Bytes per second

If you have any additional questions, please contact our support team.

Apache Webserver

Apache Configuration

Redhat/CentOS

Add the following module to your `/etc/httpd/conf/httpd.conf` file.

```
LoadModule status_module /usr/lib64/httpd/modules/mod_status.so
<IfModule status_module>
ExtendedStatus On
   <Location /server-status>
       SetHandler server-status
       Order deny,allow
       Allow from all
   </Location>
</IfModule>
```

After you make this update to the Apache `conf file, restart Apache with sudo service httpd restart.`

Ubuntu

Add the following module to your `/etc/apache2/apache2.conf file.`

```
LoadModule status_module libexec/apache2/mod_status.so
<IfModule status_module>
ExtendedStatus On
   <Location /server-status>
       SetHandler server-status
       Order deny,allow
       Allow from all
   </Location>
</IfModule>
```

After you make this update to the Apache `conf` file, restart Apache with `sudo /etc/init.d/apache2 restart`

Log File Access

To monitor 2xx / 3xx / 4xx request count, the FortiMonitor Agent user needs to be able to read the server's Apache access log file and traverse the Apache log directory. It can be found at the following location, depending on your Linux distro:

- Debian / Ubuntu: `/var/log/apache2/access.log`
- RHEL / Red Hat / CentOS: `/var/log/httpd/access_log`
- FreeBSD: `/var/log/httpd-access.log`

You may also need to update any logrotate files to ensure that permissions persist after log rotation.

Agent Configuration

The agent config file can be used to optionally override default Apache settings.

- server_status_url: Location of the Apache server status endpoint. Defaults to server-status, as in localhost/server-status
- apache_log_files: Log file containing your 2xx/3xx/4xx requests. Defaults are in the Log File Access section contained in this document. You can use comma-separated values for this setting.
- server_status_protocol: HTTP protocol to be used when accessing the server-status endpoint. Can be set to http or https and defaults to http
- server_status_port: Port to use if the status page is running on a non-standard port
- server_status_host: IP to use if Apache is not listening on localhost

/etc/fm-agent/fm_agent.cfg

```
[apache]
server_status_url = server-status
apache_log_files = /var/log/apache2/access.log,/var/log/apache2/access2.log
server_status_protocol = http
```

Control Panel Configuration

Apache monitoring is added to a server via the Monitoring Catalog. If you do not see Apache as a discovered resource, you may need to rebuild your Agent metadata. You can read about that here.

Metric	Unit
Workers used	%
Workers idle	%
Workers idle count	workers
Workers used count	workers
Uptime	seconds
Total accesses	requests
Total traffic	mb
CPU load	%
Connections	connections

Transfer rate	mb/s
Avg. request size	mb
2xx Requests	requests
3xx Requests	requests
4xx Requests	requests

Apache Zookeeper

The **Apache ZooKeeper (Linux)** plugin allows you to monitor and collect ZooKeeper metrics.

If you haven't read Add a monitoring agent resource yet, do that first.

Agent Configuration

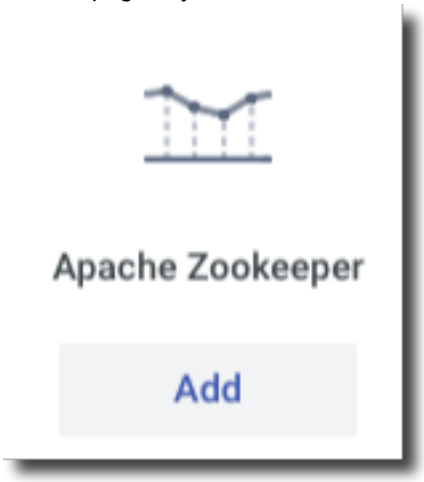
The Agent must be configured to allow metric collection from Apache ZooKeeper. To do this, add the following to your Agent configuration file.

```
[apache_zookeeper]
host = <IP ADDRESS>
port = <PORT NUMBER>
```

The **host** parameter could be **localhost**.
The **port** is usually 2181. You can verify this by checking your ZooKeeper configuration file (**zoo.cfg**).

Control panel configuration

Apache ZooKeeper monitoring will then be added to a server from the **Monitoring configuration** tab of the **Instance Details** page. If you do not see Zookeeper as a discovered resource, rebuild the Agent metadata.



To add and configure Apache ZooKeeper metrics, see Add FortiMonitor Agent checks.

Available metrics

Metric	Unit
Average request latency	ms
Maximum request latency	ms
Packets received	packets
Packets sent	packets
Packets received/sec	packets/sec
Packets sent/sec	packets/sec
Outstanding Requests	count
Server Mode	count
Node count	count
Watcher Count	count
Approximate data size	bytes
Open file descriptors	files
Slow fsync count	count
Node error state	count

ASP.NET

ASP.NET Configuration

The ASP.net plugin of agent resources lets you monitor the queue and the number of applications running on the ASP.NET framework of your server.

Control Panel Configuration

First select **ASP.NET** from the monitoring catalog. If you haven't yet read the How to add an Agent resource article, you should do that first.

The available metrics are listed below.

Metric	Unit
Number of Excepts Thrown/sec	
% Time in GC	
Applications Running	Applications Running
Errors Total	

Errors Total/sec	
Requests Executing	
Requests Executing/sec	
Requests Queued	Requests Queued
Requests/sec	

If you have any questions, please don't hesitate to reach out to our Support team!

Bandwidth (Linux)

The Bandwidth plugin is one of the Agent's most basic features, it collects the bandwidth either going in or out of your server. If you haven't read Add a monitoring agent resource yet, do that first.

Control Panel Configuration

First select **Linux** from the monitoring catalog, then scroll to bandwidth.

Then you should select the type of bandwidth data you would like collected from the list and the interface you would like to monitor.

Metric	Unit
Kilobytes IN for the month	KB
Kilobytes IN per second	KB
Kilobytes OUT for the month	KB
Kilobytes OUT per second	KB
Packets IN per second	packets
Packets OUT per second	packets

You can also change the frequency at which this data is taken. The specific number that this resource returns is defined by the above options, please take this into account if you choose to change your alert threshold for this resource.

You can collect multiple bandwidth metrics if you would like, it is actually quite common to do so especially if your server has multiple devices.

If you would like to inspect the Bandwidth plugin in more detail you can find the python script for it in /usr/lib/fm-agent/plugins/bandwidth.py.

If you have any additional questions, please contact our support team.

Bandwidth (Windows)

The Bandwidth plugin allows you to monitor the bandwidth of specific network connections.

To add a Bandwidth agent resource to your server first see Add a monitoring agent resource if you have not already.

Control Panel Configuration

First select **Windows** from the **Monitoring Catalog** menu.

There are a number of available metrics to monitor.

Metric	Units
Bytes received per month	bytes
Bytes received/sec	bytes
Bytes Sent/sec	bytes
Bytes Total/sec	bytes
Bytes transmitted per month	bytes
Bytes Transmitted/sec	bytes
Current Bandwidth	bps
Packets Received/sec	packets
Packets Sent/sec	packets

The Options dropdown will allow you to choose the specific network connection you would like to monitor.

The specific number that this resource returns is defined by the above options, please take this into account if you choose to change your Alert Threshold for this resource.

You can add multiple Bandwidth agent resources to one server if you would like, which is especially useful if you have a large number of network connections.

If you have any additional questions, please contact our support team.

Cassandra

The Cassandra plugin allows you to monitor a number of metrics that provide insight into the performance of your distributed databases. If you can not find this plugin in the /usr/lib/fm-agent/plugins folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

If you haven't read Add a monitoring agent resource yet, do that first.

Agent Configuration

If you installed Cassandra via repositories you will not need to edit the Agent configuration file and add the following configuration. However, if you did not install via a repository, then you will need to specify **install_location** in the agent config file (**/etc/panopta-agent/panopta_agent.cfg**):

```
[cassandra]
cassandra_installed_location = /home/yourname/apache-cassandra-2.0.8
```

Control Panel Configuration

After making this change give the agent a few minutes to connect with FortiMonitor's servers, then you should see the new server resource available to add for your server in the control panel.

If you have not already, read the Add FortiMonitor Agent checks article to learn how to set up monitoring for any agent plugin. To use the Cassandra plugin, first select **Cassandra** from the monitoring catalog.

Metric	Unit
Dropped Messages	
Live Space Used	bytes
Pending Tasks	tasks
Read Count	requests
Read Latency	ms
Thread pool task: active	
Thread pool task: blocked	
Thread pool task: completed	
Thread pool task: pending	
Total Space Used	
Total Space Used by Snapshots	
Write Count	requests
Write Latency	ms
SSTable Compression Ratio	
Memtable Cell Count	
Memtable Data Size	
Memtable Switch Count	
MemCompacted Partition Mean	
Bloom Filter False Positives	
Bloom Filter False Ratio	
Bloom Filter Space Used	
Row Size	
Row Count	

If you have any additional questions, please contact our support team.

CouchDB

You can use the CouchDB plugin to report on the health of your Database server. This includes information regarding the performance of your database, and the status codes being returned by the database. If you can not find this plugin in the /usr/lib/fm-agent/plugins folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

If you haven't read Add a monitoring agent resource yet, please do that first.

Agent Configuration

In order to use this plugin, you will need to add the host server address and port of your CouchDB stats page. This would be the IP address your database could be accessed from . You would need to add the address and port to the **/etc/fm-agent/fm_agent.cfg** file as shown below:

```
[couch]
host = 127.0.0.1
port = 1313
```

Control Panel Configuration

To add a CouchDB agent resource to your server, read Add FortiMonitor Agent checks. To use the CouchDB plugin, first, select **CouchDB** from the monitoring catalog by clicking **+ Add** on the icon.

If you have followed the steps above and the new metric and plugin do not show properly, you may need to manually rebuild the agent's metadata. You can do this by following the guide here.

Available CouchDB Metrics

Metric	Unit
Length of a request inside CouchDB without MochiWeb	ms
Number of bulk requests	requests
Number of file descriptors CouchDB has open	files
Number of HTTP 200 OK responses	responses
Number of HTTP 201 Created responses	responses
Number of HTTP 202 Accepted responses	responses
Number of HTTP 301 Moved Permanently responses	responses
Number of HTTP 304 Not Modified responses	responses
Number of HTTP 400 Bad Request responses	responses
Number of HTTP 401 Unauthorized responses	responses
Number of HTTP 403 Forbidden responses	responses

Number of HTTP 404 Not Found responses	responses
Number of HTTP 405 Method Not Allowed responses	responses
Number of HTTP 409 Conflict Responses	responses
Number of HTTP 412 Precondition Failed responses	responses
Number of HTTP 500 Internal Server Error responses	responses
Number of HTTP COPY requests	requests
Number of HTTP DELETE requests	requests
Number of HTTP GET requests	requests
Number of HTTP HEAD requests	requests
Number of HTTP MOVE requests	requests
Number of HTTP POST requests	requests
Number of HTTP PUT requests	requests
Number of HTTP requests	requests
Number of open databases	databases
Number of temporary view reads	reads
Number of times a database was changed	times
Number of times a document was read from a database	times
Number of view reads	reads

If you have any additional questions, please contact our support team.

CPU (Linux)

The CPU agent resource plugin is perfect if you would like to monitor your server's CPU usage. A significant slow-down of your CPU can cause a large problem for your customers and you should be aware of it.

To add a CPU agent resource to your server first see Add a monitoring agent resource if you have not already.

Control Panel Configuration

First select Linux from the monitoring catalog, scroll to the CPU category. There are a few metrics available to monitor.

Metric	Units
1 Minute CPU Load Average	Avg
15 Minute CPU Load Average	Avg
5 Minute CPU Load Average	Avg

Entropy Level	
Hardware IRQ Usage Percentage	%
I/O Wait Usage Percentage	%
Idle Usage Percentage	%
Nice Usage Percentage	%
Steal Time Usage Percentage	%
System Usage Percentage	%
Usage Percentage	%
User Usage Percentage	%

You can also monitor the usage percentage of all of your CPUs separately or as an aggregate. The specific number that this resource returns is defined by the above options, please take this into account if you choose to change your Alert Threshold for this resource.

You can monitor multiple CPU agent resources on one server if you need it. This could be useful for comparing and contrasting your CPU load averages on a micro vs macro scale, or for creating separate alerts for long, but not massive, CPU slowdowns as opposed to brief and intense ones.

If you would like to inspect the CPU plugin in more detail you can find the python script for it in /usr/lib/fm-agent/plugins/cpu_usage.py.

If you have any additional questions, please contact our support team

CPU (Windows)

The CPU agent resource plugin is perfect if you would like to monitor your server's CPU usage. A significant slow-down of your CPU can cause a large problem for your customers and you should be aware of it.

To add a CPU agent resource to your server first see Add a monitoring agent resource if you have not already.

Control Panel configuration

First, select Windows from the monitoring catalog and scroll to the CPU category. There are a few metrics available to monitor.

Metric	Unit
% Privileged Time	%
% User Time	%
Percent Used	%

You can also monitor the usage percentage of all of your CPUs separately or as an aggregate. The specific number that this resource returns is defined by the above options, please take this into account if you choose to change your Alert Threshold for this resource.

You can monitor multiple CPU agent resources on one server if you need it. This could be useful for comparing and contrasting your CPU load averages on a micro vs macro scale, or for creating separate alerts for long, but not massive, CPU slowdowns as opposed to brief and intense ones.

If you have any additional questions, please contact our support team.

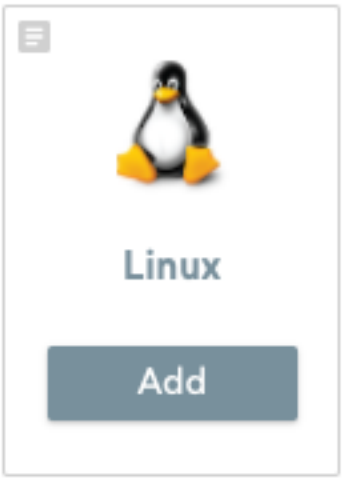
Disk (Linux)

The Disk plugin lets you monitor how much space you have available on your server. This tool can be particularly useful for server management, for example, you can trigger an alert when the disk space falls below a certain threshold. Monitoring your disk space, and preventing disk filling, is one of the greatest ways to prevent outages.

To add a Disk resource type to your server first see Add FortiMonitor Agent checks if you have not already.

Control Panel Configuration

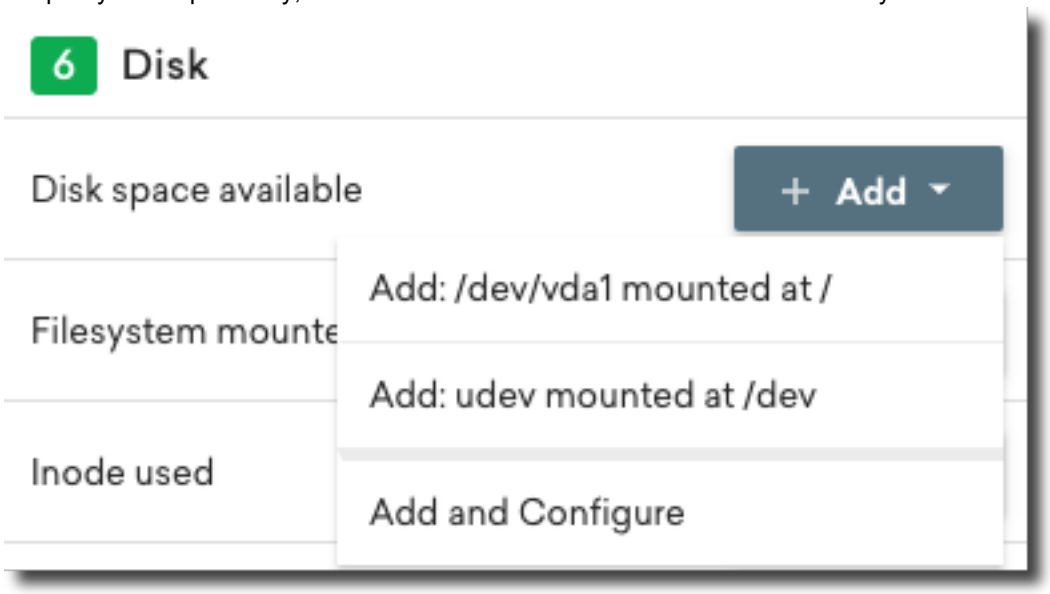
First select Linux from the monitoring catalog.



Scroll to the Disk section, there are a number of metrics to choose from.

Metric	Unit
Disk Space Available	kb
Filesystem mounted	
Inode used	Inodes
Inodes Available	Inodes
Inodes percent used	Inodes
Percentage of disk used	%

The metric will have three options: "Percentage of Disk Used", "Available Disk Space" and "File Systems". The first two are pretty self-explanatory, but the third is a Boolean test to see if a remote file system is mounted to your server.



Clicking **Add** will give you finer control over the alert timeline used as well as thresholds. The specific number that this resource returns is defined by the above options, please take this into account if you choose to change your Alert Thresholds for this resource. For this resource, we recommend having at least two alert thresholds. The first is send out email notifications if the disk space is getting low, and the second to send out SMS/voice alerts when the disk space is dangerously low and requires immediate attention.

You can add multiple Disk agent resources to one server if you would like, which is especially useful if you have a large number of file systems.

If you would like to inspect the Disk plugin in more detail you can find the python script for it in **/usr/lib/fm-agent/plugins/disk.py**.

Use fstab to specify mount points that should be monitored

You can use the fstab file to monitor unmounted filesystems that have never by adding the following parameter to the Agent configuration file:

```
[disk]
use_findmnt = True
```

Override filesystem discovery

By default, the following partitions are excluded from discovery:

- tmpfs
- devtmpfs
- none
- proc
- swap
- devices

- cgroup
- /dev/loop

To override the default values, open the Agent configuration file and modify the following parameter:

```
[disk]
device_ignore_list = ("tmpfs", "devtmpfs", "none", "proc", "swap", "devices", "cgroup",
     "/dev/loop")
```

Disk (Windows)

The Disk plugin of agent resources lets you know how much space you have available on your server. This tool can be particularly useful for server management, for example you can trigger an alert when the disk space falls below a certain threshold.

To add a Disk check to your server first see Add FortiMonitor Agent checks if you have not already.

Control Panel Configuration

First scroll to **Disk** under the **Windows** plugin.

There are a number of metrics available to be monitored.

Metric	Unit
Avg. Disc sec/read	
Avg. Disc sec/write	
Current Disk Queue Length	requests
Disk bytes/sec	bytes
Disk reads/sec	reads/sec
Disk writes/sec	writes/sec
Percent Free	%
Percent Used	%
Space Free	KB

Physical Disk vs Logical Disk

You will notice two different disk space monitoring categories. For a more detailed explanation of the difference between physical disk and logical disk, please reference the below article:https://blogs.technet.microsoft.com/askcore/2012/03/16/windows-performance-monitor-disk-counters-explained/

The disc metric will allow you to choose between monitoring the total free space in megabytes, or as a percentage. The “Options” drop-down will allow you to select which partition you would like to check, including the options to monitor the aggregate of all partitions on the server (this option is called “_Total”).

The specific number that this resource returns is defined by the above options, please take this into account if you choose to change your Alert Threshold for this resource.

You can add multiple Disk agent resources to one server if you would like, which is especially useful if you have a large number of partitions.

If you have any additional questions, please reach out to our support team.

Elasticsearch

You can use the Elasticsearch plugin to monitor your cluster health and sharding within Elasticsearch. If you can not find this plugin in the /usr/lib/fm-agent/plugins folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

Agent Configuration

In order to use this plugin, you will have to ensure you can load the **/_clusterhealth** endpoint locally from your machine. Enter the IP address and port for the **/_clusterhealth** endpoint into your Agent Config file as shown below:

```
[elasticsearch]
hostname = 127.0.0.1
port = 1313
username =
password =
```

Note:The **Username** and **Password** fields are optional.

Control Panel Configuration

To add an ElasticSearch agent resource to your server, read Add FortiMonitor Agent checks. To use the ElasticSearch plugin, first select **ElasticSearch** from the monitoring catalog.

There are a number of available metrics to choose from.

Metric	Unit
Number of active primary shards	shards
Number of data nodes in cluster	nodes
Number of initializing shards	shards
Number of shards getting relocated	shards
Number of unassigned shards	shards
Total number of nodes in a cluster	nodes
Total number of shards	shards

From the configuration screen, you can choose the frequency of the check, the alert schedule, set thresholds, and tag the check.

If you have any additional questions, please contact our support team.

Exim

Exim is a mail transfer agent and the Exim plugin allows you to monitor the length of your Exim queue. A large queue tells you that your server is not handling requests fast enough to keep up with demand.

To add an Exim agent resource to your server first see Add FortiMonitor Agent checks if you have not already.

Control Panel Configuration

First select **Exim** from the Monitoring catalog.

You'll be taken to a page with the pre-existing metrics for Exim are listed, as well as an +Add Monitoring button option at the top right. A drawer will slide open from the right with the options available with this plugin.

Metric	Unit
Queue Depth	depth

The depth of this queue is the number that this resource returns, please take this into account if you choose to change your Alert Thresholds.

If Exim is not an option in the dropdown, then you may need to add the panopta-agent user to the group that owns Exim on your server. If that does not work, then you can change the queue_list_requires_admin config entry to allow non-admin users to view the queue as described in this article.

If you would like to inspect the Exim plugin in more detail you can find the python script for it in /usr/lib/fm-agent/plugins/exim.py.

If you have any additional questions, please contact our support team.

File Information (Linux)

The File Information agent resource plugin is used to monitor one of three things: whether a file exists, how long a file has existed, and how long since a file was last modified. This is a pretty flexible tool and you can use it in a number of different ways. For example, if you would like to track the completion of regularly scheduled jobs, then you can configure your job to touch a file once it finishes and we can alert you if the file's age gets beyond a specified threshold.

To add a File Information agent resource to your server first see Add FortiMonitor Agent checks if you have not already.

Control Panel Configuration

First select Linux from the monitoring catalog.

Scroll to the File section. There a number of available metrics.

Metric	Unit
Creation age	minutes
File exists	boolean
File size (KB)	KB
Modification age	minutes

You can choose any file path you would like to monitor.

The specific number that this resource returns is defined by the above options, please take this into account if you choose to change your alert thresholds for this resource.

You can add as many File Information agent resources to one server as you need.

If you would like to inspect the File Information plugin in more detail you can find the python script for it in **/usr/lib/fm-agent/plugins/file_presence.py**.

If you have any additional questions, please contact our support team.

File Information (Windows)

The File Information plugin is used to check one of three things: whether a file exists, how long a file has existed, and how long since a file was last modified. This is a pretty flexible tool and you can use it in a number of different ways. For example, if you have a file that is supposed to be updated periodically, this will check to make sure that this update is actually occurring.

Control Panel Configuration

To add a File Information agent resource to your server first see Add FortiMonitor Agent checks if you have not already.

First, scroll to File from the Windows metrics options.

Metric	Unit
Creation age	minutes
File exists	boolean
Modification Age	minutes

The modification age resource will return how long (in seconds) it has been since the specified file was modified. The file exists will return either true of false for whether the specified file exists (remember true = 1). The creation age will return how long a file has existed (in seconds). You can choose any file path you would like to monitor.

The specific number that this resource returns is defined by the above options, please take this into account if you choose to change your Alert Threshold for this resource.

You can add as many File Information resources to one server as you need. If you have any additional questions, please contact our support team.

HAProxy

The HAProxy plugin allows you to monitor a number of metrics to ensure that HAProxy is successfully load balancing and serving requests. HAProxy provides two methods to monitor its performance:

- Via the Stats page
- Via a Unix socket

Stats page

The most common way to collect HAProxy metrics is via the stats page. To allow the Agent to access HAProxy metrics via the stats page, enable it by adding the following to the bottom of the **/etc/haproxy/haproxy.cfg** file.

```
listen stats
  bind :8080
  mode http
  stats enable
  stats hide-version
  stats realm Haproxy\ Statistics
  stats uri /haproxy_stats  # Stats URI
  stats auth username:password  # Authentication credentials
```



Authorization credentials are optional. **user** is the only level of permissions required for the Agent. However, **admin** or **operator** can be used as well.

After modifying the configuration file, restart HAProxy. After restarting the HAProxy, make sure that you can access your HAProxy's stats page.

Agent configuration

The Agent must also be configured to collect metrics from the HAProxy stats page. To do this, add the following to your Agent configuration file.

```
[haproxy]
stats_http_host = localhost
stats_http_port = 8080
stats_http_path = /haproxy_stats
stats_http_user = username
stats_http_password = password
```



Authorization credentials are only required if they are used in the HAProxy configuration file.

Unix socket

Another method to enable metric collection is via a Unix socket. Your HAProxy configuration file will need to have stats enabled. To enable stats, add the following to the bottom of the **/etc/haproxy/haproxy.cfg** file.

HAProxy configuration

```
global
  stats socket /var/run/haproxy.sock mode 666 level user
  stats timeout 2m
listen admin
  bind *:8080
  stats enable
```

user is the only level of permissions required for the Agent. However, **admin** or **operator** can be used as well.

Agent configuration

The Agent can collect metrics via the HAProxy socket. By default, the Agent connects to **/var/run/haproxy.sock**. Optionally, if you have customized this location in your installation, you can let us know in the Agent config file using the socket_cfg_file variable.

```
[haproxy]
socket_cfg_file =/var/run/haproxy.sock
```

Control Panel Configuration

Like other application monitoring, HAProxy monitoring is added to a server via the Monitoring Catalog. If you don't see HAProxy as a discovered resource, you may need to rebuild your Agent metadata. You can read about that here.

Once HAProxy is added, you can begin monitoring any of the below metrics.

Metric	Unit
Bytes In	bytes/sec
Bytes Out	bytes/sec
Queued requests	requests
Sessions	sessions
Request errors	errors
Response errors	errors
Active servers	servers
Backup servers	servers
Connection Errors	errors
1xx's	requests
2xx's	requests
2xx's	requests

If you have any additional questions, please contact our support team.

Microsoft Exchange

The Microsoft Exchange plugin allows you to monitor the Exchange service on your server.

If you haven't yet read the Add FortiMonitor Agent checks article, please start there!

Control Panel Configuration

Select MS Exchange from the monitoring catalog.

There are a number of metrics available.

Category	Metric	Unit
AD Access Caches	LDAP Searches/sec	searches
AD Access Domain Controllers	Long Running LDAP operations/min	operations/min
OWA	Average Search Time	ms
rpclientaccess	RPC Averaged Latency	ms
	RPC Requests	requests
Client	Messages Queued for Submission	messages
Mailbox Total	Messages Queued for Submission	messages
Public Total	Messages Queued for Submission	messages

If you have any further questions, please contact our support team.

Microsoft Hyper-V

The Hyper-V plugin allows you to monitor the Hyper-V application running on your servers.

Before you start, see How to add a monitoring agent check.

Control Panel Configuration

Select the Hyper-V plugin from the monitoring catalog.

There are a few different metrics to choose from.

Metric	Unit	Description
Logical Processor		
Context Switches/sec	context switches/sec	The number of times per second each logical CPU changes what virtual processor it is running.

% Total Run Time	%	Utilization of the physical processors in the host.
Virtual Processor		
% Hypervisor Run Time	%	The total processor utilization of the hypervisor for the entire system.
% Guest Run Time	%	The percentage of the time spent by the virtual processor in guest code for a specific VM.
% Total Run Time	%	The percentage of the time spent by the processor in guest and hypervisor code for a specific VM.
% Remote Run Time	%	The percentage of the time spent by the processor in remote sessions.

If you have any questions, please reach out to FortiMonitor Support.

I/O

The I/O plugin allows you to monitor the percentage of CPU time during which I/O requests were issued to the device. In order to have access to this plugin, you will need to install the sysstat package to your server. The process for installation is different depending on OS, so follow the process per your needs.

To add an I/O agent resource to your server first see Add FortiMonitor Agent checks if you have not already.

Control Panel Configuration

First, select I/O from the Linux category.

There are a number of available metrics.

Metric	Unit
% of I/O CPU time	%
Average I/O request service time	
Read requests per second	
Read requests queued per second	
Write requests per second	
Write requests queued per second	

If you would like to inspect the I/O plugin in more detail you can find the python script for it in /usr/lib/fm-agent/plugins/io_stats.py.

If you have any additional questions, please contact our support team.

IIS

IIS Plugin Configuration

The IIS plugin allows you to monitor the state of any of the application pools on your server.

To add an IIS agent resource to your server first see Add FortiMonitor Agent checks if you have not already.

The plugin will only have one option; Current Application Pool State. The “Options” drop-down will allow you to choose the application pool you would like to monitor, where “Total” lets you monitor the aggregate of your application pools.

Control Panel Configuration

First select IIS from the monitoring catalog.

There are a number of metrics to choose from.

Metric	Unit
Bytes received/sec	bytes
Bytes sent/sec	bytes
Bytes total/sec	bytes
Connection attempts/sec	connections
Current Application Pool State	The current status of the IIS application pool running on the system. 1 - Uninitialized 2 - Initialized 3 - Running 4 - Disabling 5 - Disabled 6 - Shutdown Pending 7 - Delete Pending
Current Connections	connections
Get Requests	requests
Get Requests/sec	requests
Head requests/sec	requests
Maximum Connections	connections
Post Requests	requests
Post Requests/sec	requests

You can monitor as many application pools as you like. If you have any additional questions, please contact our support team.

JBoss

The JBoss plugin allows you to monitor a number of metrics that provide insight into the performance of your application server. If you can not find this plugin in the /usr/lib/fm-agent/plugins folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

Agent Configuration

Once the Agent is installed you will need to add the following information to the agent config file (**/etc/fm-agent/fm_agent.cfg**):

```
[jboss]
username = <your_provided_user_name>
password = <your_provided_password>
console_url = http://localhost:999
```

You will need to use the username and password you use to login to your JBoss console URL. After making this change give the agent a few minutes to connect with FortiMonitor's servers, then you should see new server resources available to add for your server in the control panel.

Control Panel Configuration

If you have not already, read the Add FortiMonitor Agent checks article to learn how to set up monitoring for any Agent plugin. Select JBoss from the monitoring catalog.

There are a number of metrics to choose from:

- Number of daemon threads used
- Pool active count
- Pool available count
- Pool average blocking time
- Pool average creation time
- Pools created
- Pools destroyed
- Pools max creation time
- Pools max used count
- Pools max wait time
- Pools timed out
- Pools total blocking time
- Pools total creation time
- Prepared statement cache access
- Prepared statement cache add
- Prepared statement cache current size
- Prepared statement cache delete
- Prepared statement cache hit
- Prepared statement cache miss
- Time spend for idle pools to be timeout
- Total heap memory committed
- Total heap memory used
- Total non-heap memory committed

- Total non-heap memory used
- Total number of live threads used

If you would like to inspect the JBoss plugin in more detail you can find the python script for it in the /usr/lib/fm-agent/plugins folder.

If you have any additional questions, please contact our support team.

JMX (Linux)

The JMX plugin allows you to connect and remotely monitor metrics from your Java applications.

Please note: We will not go into detail on enabling JMX or configuring it to use password authentication. More information can be found on Oracle's site. Please make sure that your server is using Oracle's JDK and Python's JPype or OpenJDK with the proper workaround for JPype.

Agent Configuration

Once the agent is installed, you will need to add the following information to the agent config file (**//etc/fm-agent/fm_agent.cfg**). If you have multiple Java processes running on different JMX ports, use comma-separated values as shown below:

```
[jmx]
username = username1, username2
password = password1, password2
port = port1, port2
host = hostname1, hostname2
jvm_path = /PATH/TO/JVM
```

Fill in the username and password fields with the ones you have set for your application, in our case, our Tomcat admin username and password was used. You can also configure this without the username/password fields if you have your java application configured without the need for authentication. Specify the port/s and host/s from the setenv.sh you created and then replace the last field with the correct path to your Java install. If you need the path for the Java install, it can be found by grabbing the output of the command below:

```
locate libjvm.so
```

If you have multiple locations returned, make sure to use the one for your current Java version. In our case it was:

```
/usr/java/jdk1.8.0_25/jre/lib/amd64/server/libjvm.so
```

Control Panel Configuration

To use the JMX plugin, first select Java from the monitoring catalog.

Then you can add JVM metrics to monitor from the agent using the example below.



Metric Syntax

The syntax for this metric is a forward-slash separated URI. You can obtain the MBean object from your JMX browser and the attribute by drilling down in the browser to get the attribute name.

```
Copy Codejava.lang:type=Threading/ThreadCount
java.lang:type=Memory/HeapMemoryUsage/used
kafka.server:type=BrokerTopicMetrics,name=MessagesInPerSec,topic=check_result_stream/Count
```

If you would like to inspect the JMX plugin in more detail you can find the python script for it in the **/usr/lib/fm-agent/plugins** folder.

If you have followed the steps above and the new metric and plugin do not show properly, you may need to manually rebuild the agent's metadata. You can do this by following the guide here.

JMX (Windows)

JMX comes as a base plugin with the Windows Agent. Make sure the server has the agent installed and JMX is enabled for your Java installation and the JRE location is set in your path variable. This is more in-depth than this tutorial will cover, but the best way to know if you need to set this is to open a new CMD prompt and type:
`echo %JAVA_HOME%`

Agent Configuration

You will then need to edit your agent config file located at **C:\Program Files\FortiMonitorAgent\Agent.config**. Close to the bottom of the file, you will see tags that start with **<config>** make sure to put your code in this block and substitute your information for the placeholders below:

```
<jmx>
  <add key="username" value="admin" />
  <add key="password" value="password" />
  <add key="host" value="localhost" />
  <add key="port" value="8686" />
  <add key="serverPort" value="8989"/>
  <add key="socketTimeout" value="3500"/>
  <add key="useSSL" value="true"/>
  <add key="trustStore" value="c:\jmx-test\keystore.jks" />
  <add key="trustStorePassword" value="keystore-password" />
</jmx>
```

Where:

- **username / password** - the credentials used for JMX.
- **host/port** - the host and port configured for JMX.
- **serverPort** - the port set for the JMX agent to accept connections. The default is 8989.
- **socketTimeout** - the socket timeout from the Windows agent to the JMX agent. The default is 2500ms.

If the JMX server is running with SSL enabled, the following options need to be specified:

1. **useSSL** - set to true.
2. **trustStore** - the file path to the trustStore used for key/certificate management.
3. **trustStorePassword** - the password for the trustStore specified above.

Look for the **<plugin>** tag then add the following:

```
<resource categoryTextKey= "jmx" category="JMX" textKey="jmx.custom" label="Custom JMX Metric" />
```

Ideally, the above line should be added after the **event_log** line. For example:

```
<plugin>
        <resource categoryTextKey="event_log" category="Event Logs" textKey="count" unit="count" label="Event logs count" />
        <resource categoryTextKey="jmx" category="JMX" textKey="jmx.custom" label="Custom JMX Metric" />
</plugin>
```

Control Panel Configuration

To use the JMX plugin, first select Java from the monitoring catalog.

Then you can add JVM metrics to monitor from the agent using the example below.



Metric Syntax

The syntax for this metric is a forward-slash separated URI. You can obtain the MBean object from your JMX browser and the attribute by drilling down in the browser to get the attribute name.

```
java.lang:type=Threading/ThreadCount
java.lang:type=Memory/HeapMemoryUsage/used
kafka.server:type=BrokerTopicMetrics,name=MessagesInPerSec,topic=check_result_stream/Count
```

The entries shown above are the combination of the BEAN and the attributes which you would like to query on. You can run these in a command-line interface as shown here.

If you have followed the steps above and the new metric and plugin do not show properly, you may need to manually rebuild the agent's metadata. You can do this by following the guide here.

Linux Log Monitoring

The Linux log monitoring agent plugin allows you to monitor for occurrences of a defined pattern over a given period of time in a specific log file. A simple example would be to monitor the number of times the string "ERROR" appeared in your Apache logs over the past 5 minutes.

Metric Configuration

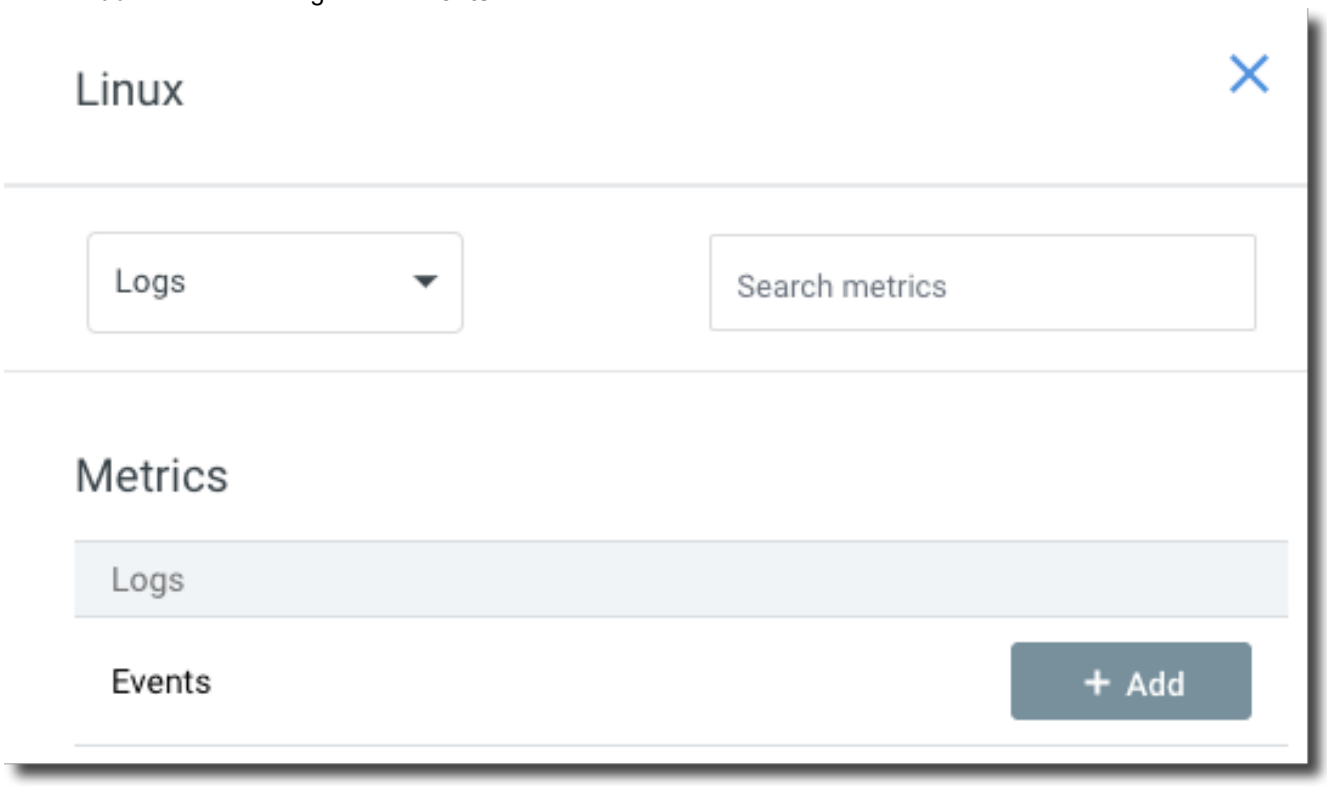
As a prerequisite, ensure you have the latest version of the Linux agent installed - if you need a refresher, go here.

Within the Linux monitoring category in the monitoring configuration table, click **Add Metric**. The Linux metric configuration drawer should now be open on the right side of the page.

Navigate to the **Logs** section.

Metric	Unit
Events	count

Select **Add** to add and configure the **Events** metric.



Field	Description
Log Location	The full path to the log file which you would like to monitor.
Log Message Regex Filter	A regex to utilize on each logline. This should be used to look for the specific occurrence you are concerned with. If left blank, this will count every line in the file.
Time Frame	How far back to look for occurrences. For instance, if you want to check for "errors in the past 5 minutes", select 5 minutes.
Frequency	How often the metric should be collected.
Tags	Add tags to the metric.

Because of how the results buffer is built, it will take a period of time to build the initial metric value. This period is the same length of time as the configured **Time Frame**.

If you have any additional questions, please contact our support team.

Memcache

The Memcache plugin will allow you to monitor a number of metrics on the Memcache and Memcached services of your server. Use it to monitor if your cached items were accessible, how often they are being accessed, the number of worker threads et If you can not find this plugin in the /usr/lib/fm-agent/plugins folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

Agent Configuration

In order for the Memcache plugin to be operational, you need to add the following snippet to **/etc/fm-agent/fm_agent.cfg**:

```
[memcache]
hostname = 127.0.0.1
port = 11211
```

Control Panel Configuration

To add a Memcache agent resource to your server, read Add FortiMonitor Agent checks if you have not already. To use the Memcache plugin, first select “Memcache” from the monitoring catalog.

There are a number of available metrics to choose from.

Metric	Unit
Current number of items	items
Current number of open connections	connections
Number of bytes	B
Number of connection structures	structures
Number of items that have been requested and not found	items
Number of keys that have been requested and found present	keys
Number of valid items removed from cache to free memory for new items	items
Number of worker threads requested	threads
Size of pointers	bits
Total number of bytes read	B
Total number of bytes sent	B

Total number of connections opened	connections
Total number of items stored	items
Total number of retrieval requests (get operations)	requests
Total number of storage requests (set operations)	requests
Total system time	
Total user time	

If you would like to inspect the Memcache plugin in more detail you can find the python script for it in the **/usr/lib/panopta-agent/plugins** (panopta-agent) or **/usr/lib/fm-agent/plugins** (fm-agent) folder.

If you have any other questions, please contact our support team

Memory (Linux)

The Memory plugin monitors how much memory (either swap or RAM) your server is using (either as a percentage or amount). This can be useful for sending out alerts when your server is using too much memory.

To add a Memory resource type to your server first see Add FortiMonitor Agent checks if you have not already.

Control Panel Configuration

First, select the Linux category from the catalog, and scroll to the Memory section.

There are a number of metrics to choose from.

Metric	Unit
Cached	kb
RAM active	kb
RAM buffer	kb
RAM inactive	kb
RAM percentage used	%
RAM used	kb
Swap cached	kb
Swap percentage used	%
Swap used	kb

The metric options will let you choose between Swap and RAM, as well as letting you choose between whether the check returns a percentage or a total used figure. The specific number that this resource returns is defined by the above options, please take this into account if you choose to change your alert thresholds for this resource.

You can monitor multiple Memory Agent resources on one server if you would like, but that is not often recommended as one should be plenty.

If you would like to inspect the Memory plugin in more detail you can find the python script for it in **/usr/lib/fm-agent/plugins/memory_usage.py**.

MongoDB

The MongoDB plugin allow you to monitor a number of metrics on the Mongo database on your server. If you have not installed MongoDB and you would like to install it go to https://docs.mongodb.com/manual/installation/ for instructions. If you can not find this plugin in the **/usr/lib/fm-agent/plugins** folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

Agent Configuration

Once the agent is installed, you will need to add the following information to the agent config file (**/etc/fm-agent/fm_agent.cfg**):

```
[mongo]
hostname = 10.10.10.100
port = 27017
username = mongo_username
password = mongo_password
database = optional_specific_database_to_access
authenticationdb = optional_db_used_for_authentication
```

Control Panel Configuration

To add a MongoDB Agent resource to your server first see Add FortiMonitor Agent checks if you have not already.

First select “MongoDB” from the monitoring catalog.

There are a number of metrics available to monitor.

Metric
Available connections
Current connections
Difference - member's ping ms from primary node's ping ms
Difference - primary node optime date and realtime
Is replica set PRIMARY member
MongoDB replica set member health
MongoDB replica set member state
Number of operations queued waiting for read lock
Number of operations queued waiting for write lock

Open cursors
Timed out cursors
Total number of operations queued waiting for lock

The number that this resource returns is defined by the above options, please take this into account if you choose to change your alert thresholds.

If you would like to inspect the MongoDB plugin in more detail you can find the python script for it in **/usr/lib/fm-agent/plugins/mongo.py**.

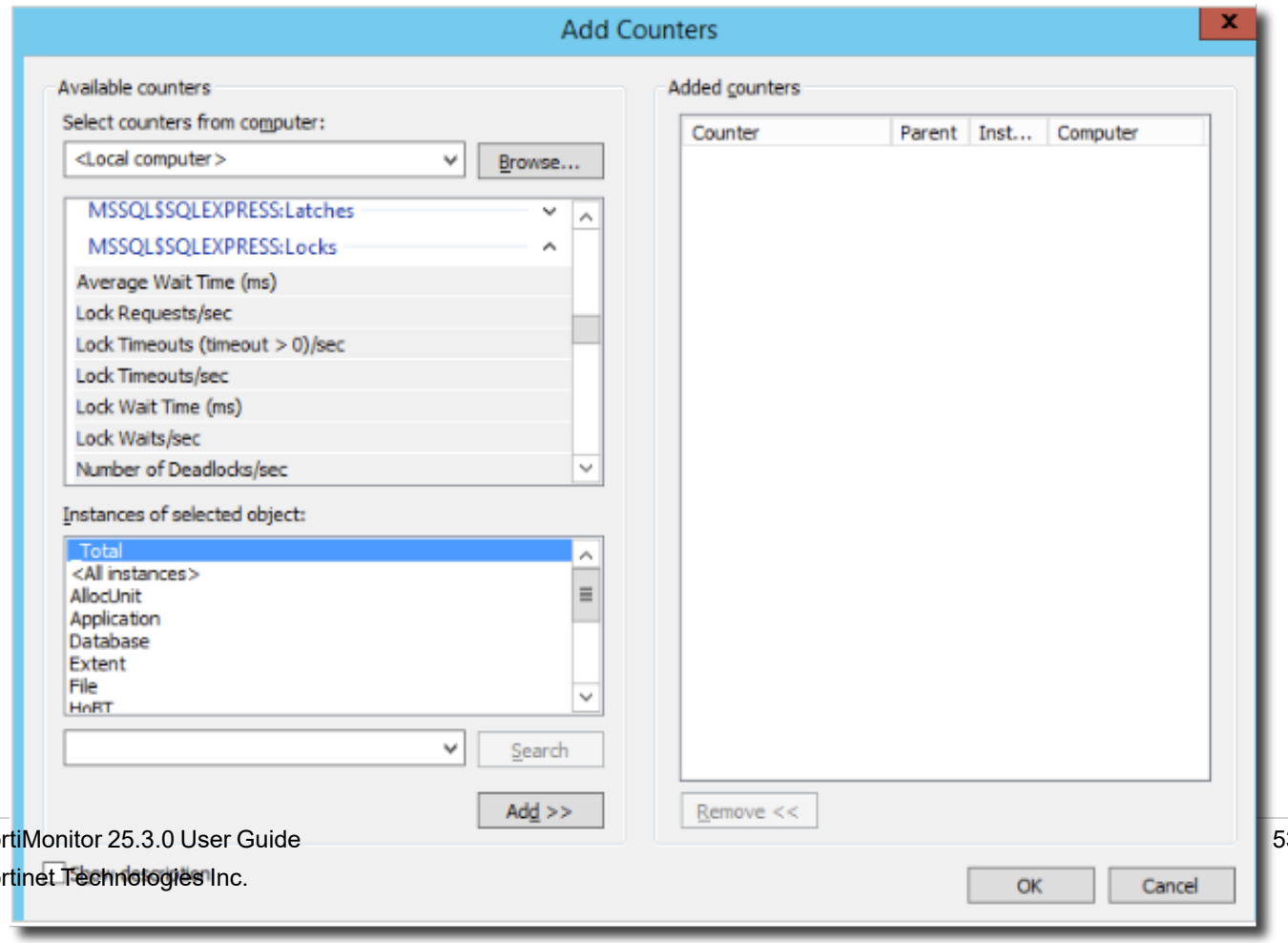
MSSQL

The MSSQL plugin allows you to monitor a number of useful metrics on your database servers. You can select the MSSQL from the monitoring catalog,

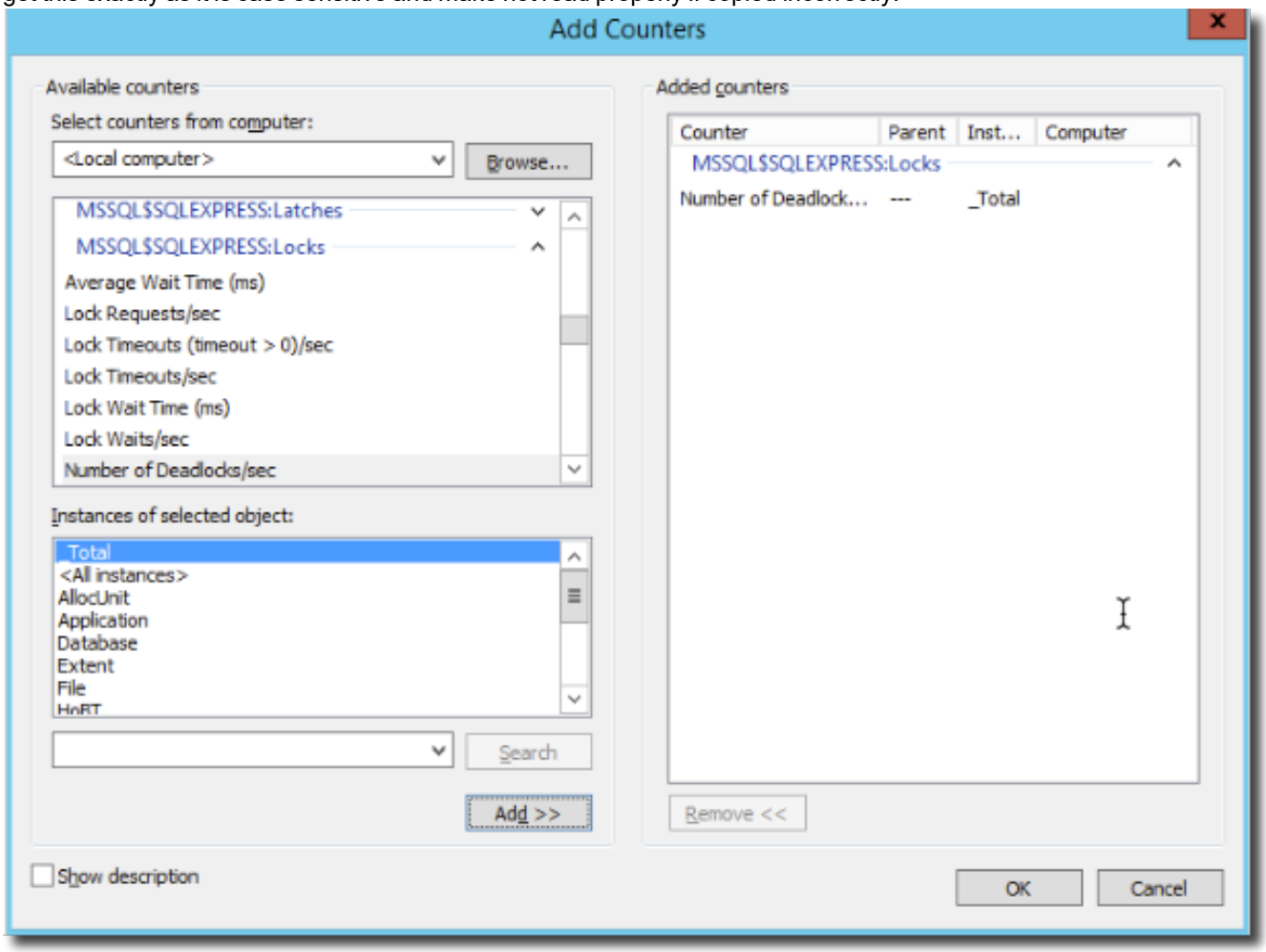
MSSQL Server Configuration

If you installed MSSQL as a 'Named Instance' there will be extra configuration changes in the C:\Program File\PanoptaAgent\Agent.config, you need to make sure your MSSQL install is not running as a named instance. If it is, please follow the guidelines for getting the correctly named metrics with Perfmon here.

Get the instance named category which in this case will be 'MSSQL$SQLEXPRESS:Locks" yours may be something different such as "MSSQL$EXPRESS":



Then make sure to grab the exact label name. In this case it is “Number of Deadlocks/sec” it is important to make sure to get this exactly as it is case sensitive and make not read properly if copied incorrectly:



Make sure to grab the correct categories and add them. They will look similar to the text below. You can choose to do a find and replace of the plugin categories but the labels should be the same.

```
<resource category="MSSQL$SQL2012:Databases" label="Repl. Trans. Rate" unit="transactions" />
<resource category="MSSQL$SQL2012:Databases" label="Shrink Data Movement Bytes/sec" unit="bytes"
/>
<resource category="MSSQL$SQL2012:Databases" label="Tracked transactions/sec"
unit="transactions" />
<resource category="MSSQL$SQL2012:Databases" label="Transactions/sec" unit="transactions" />
<resource category="MSSQL$SQL2012:Databases" label="Write Transactions/sec" unit="transactions"
/>
<resource category="MSSQL$SQL2012:Locks" label="Number of Deadlocks/sec" />
```

Control Panel Configuration

To add a MSSQL agent resource to your server first see Add FortiMonitor Agent checks if you have not already.

Select Microsoft SQL server from the monitoring catalog.

There are a number of metrics available to be monitored.

Metric	Unit
Active jobs	
Active steps	
Active transactions	
Backup/Restore Throughput/sec	reads/writes
Batch Requests/sec	requests
Buffer cache hit ratio	ratio
Bulk copy rows/sec	row
Bulk copy throughput/sec	kilobytes
Checkpoint pages/sec	pages
Commit table entries	kilobytes
Compilations/sec	comps
Data file(s) size KB	kilobytes
DBCC Logical scan bytes/sec	bytes
Deadlocks/sec	
Failed jobs	
Log cache hit ratio	ratio
Log cache reads/sec	reads
Log file(s) size KB	kilobytes
Log file(s) used size KB	kilobytes
Log flush wait time	milliseconds
Log flush waits/sec	commits
Log flushes/sec	flushes
Log growths	growths
Log pool cache misses/sec	misses
Log pool disk reads/sec	reads
Log pool requests/sec	requests
Log shrinks	shrinks
Log truncations	truncations
Memory grants pending	

Page life expectancy	seconds
Page splits/sec	splits
Percent log used	%
Processes blocked	processes
Queued steps	
Re-Compilations/sec	re-comps
Repl. Pending Xacts	transactions
Repl Trans. Rate	transactions
Shrink data movement bytes/sec	bytes
Successful jobs	
Tracked transactions/sec	transactions
Transaction delay	milliseconds
Transactions/sec	trans
User connections	connections
Write transactions/sec	transactions

The specific number that this resource returns is defined by the above options, please take this into account if you choose to change your Alert Threshold for this resource.

MySQL

The MySQL plugin allows you to monitor a number of metrics about the MySQL database instances running on your server. If you can not find this plugin in the **/usr/lib/fm-agent/plugins** folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

Agent Configuration

In order to use this plugin you will need to add the usernames and passwords for all of your MySQL instances to the **/etc/fm-agent/fm_agent.cfg** file. If you have multiple MySQL instances running on the server, use comma-separated values as shown below:

```
[agent]
[mysql]
host = 127.0.0.1, 127.0.0.1
port = 3306, 3307
username = USERNAME1, USERNAME2
password = PASSWORD1, PASSWORD2
```



The **host** is optional and defaults to **127.0.0.1**.

Replace USERNAME and PASSWORD with the appropriate username and password for a user that has local MySQL access. We recommend setting up a monitoring user that is used only for this purpose. In order to access replication status information, the user needs to be granted either SUPER or REPLICATION CLIENT permissions.

You would also need to have both MySQL and MySQLD or MySQLD-safe installed onto the server you were trying to configure this type of monitoring for.

Control Panel Configuration

After making this change, give the Agent a few minutes to sync with FortiMonitor's servers then you should see new server resources available to add for the server in the control panel.

To add a MySQL Agent resource to your server first see Add FortiMonitor Agent checks if you have not already.

First select **MySQL** from the monitoring catalog.

There are a number of metrics available to monitor.

Metric	Unit
Average number of slow queries/second	queries per second
Average queries/second	queries per second
Innodb row lock current waits	waits
Innodb row lock time avg	ms
MySQL connections	connections
MySQL query cache amount free (KB)	KB
MySQL query cache percent free	%
MySQL slave server is connected to the master	boolean
MySQL slave server is replicating	boolean
MySQL slave server latency (seconds)	seconds
Query cache amount free	KB
Threads connected	threads
Threads created	threads
Threads running	threads
Uptime	seconds

There is a lot you can do with this plugin, so feel free to add as many MySQL Agent resources to your server as you would like.

If you would like to inspect the MySQL plugin in more detail you can find the python script for it in **/usr/lib/fm-agent/plugins/mysql.py**.

Optional Metrics

If there are MySQL metrics that you would like monitor that are not included with our standard package, then you can use the **fm_agent.cfg** file to add additional metrics. First edit the **/etc/fm-agent/fm_agent.cfg** file and add an extended_ metrics line to your **[mysql]** block. Your MySQL block should now look something like this:

```
[mysql]
username = USERNAME
password = PASSWORD
extended_metrics =
```

You can then fill out the extended_metrics field with a comma-separated list of metrics. The following list will give you all of the possible metrics to monitor.

- Aborted_clients
- Aborted_connects
- Binlog_cache_disk_use
- Binlog_cache_use
- Binlog_stmt_cache_disk_use
- Binlog_stmt_cache_use
- Bytes_received
- Bytes_sent
- Com_admin_commands
- Com_assign_to_keycache
- Com_alter_db
- Com_alter_db_upgrade
- Com_alter_event
- Com_alter_function
- Com_alter_procedure
- Com_alter_server
- Com_alter_table
- Com_alter_tablespace
- Com_analyze
- Com_begin
- Com_binlog
- Com_call_procedure
- Com_change_db
- Com_change_master
- Com_check
- Com_checksum
- Com_commit

- Com_create_db
- Com_create_event
- Com_create_function
- Com_create_index
- Com_create_procedure
- Com_create_server
- Com_create_table
- Com_create_trigger
- Com_create_udf
- Com_create_user
- Com_create_view
- Com_dealloc_sql
- Com_delete
- Com_delete_multi
- Com_do
- Com_drop_db
- Com_drop_event
- Com_drop_function
- Com_drop_index
- Com_drop_procedure
- Com_drop_server
- Com_drop_table
- Com_drop_trigger
- Com_drop_user
- Com_drop_view
- Com_empty_query
- Com_execute_sql
- Com_flush
- Com_grant
- Com_ha_close
- Com_ha_open
- Com_ha_read
- Com_help
- Com_insert
- Com_insert_select
- Com_install_plugin
- Com_kill
- Com_load
- Com_lock_tables
- Com_optimize
- Com_preload_keys
- Com_prepare_sql
- Com_purge

- Com_purge_before_date
- Com_release_savepoint
- Com_rename_table
- Com_rename_user
- Com_repair
- Com_replace
- Com_replace_select
- Com_reset
- Com_resignal
- Com_revoke
- Com_revoke_all
- Com_rollback
- Com_rollback_to_savepoint
- Com_savepoint
- Com_select
- Com_set_option
- Com_signal
- Com_show_authors
- Com_show_binlog_events
- Com_show_binlogs
- Com_show_charsets
- Com_show_collations
- Com_show_contributors
- Com_show_create_db
- Com_show_create_event
- Com_show_create_func
- Com_show_create_proc
- Com_show_create_table
- Com_show_create_trigger
- Com_show_databases
- Com_show_engine_logs
- Com_show_engine_mutex
- Com_show_engine_status
- Com_show_events
- Com_show_errors
- Com_show_fields
- Com_show_function_status
- Com_show_grants
- Com_show_keys
- Com_show_master_status
- Com_show_open_tables
- Com_show_plugins
- Com_show_privileges

- Com_show_procedure_status
- Com_show_processlist
- Com_show_profile
- Com_show_profiles
- Com_show_relaylog_events
- Com_show_slave_hosts
- Com_show_slave_status
- Com_show_status
- Com_show_storage_engines
- Com_show_table_status
- Com_show_tables
- Com_show_triggers
- Com_show_variables
- Com_show_warnings
- Com_slave_start
- Com_slave_stop
- Com_stmt_close
- Com_stmt_execute
- Com_stmt_fetch
- Com_stmt_prepare
- Com_stmt_reprepare
- Com_stmt_reset
- Com_stmt_send_long_data
- Com_truncate
- Com_uninstall_plugin
- Com_unlock_tables
- Com_update
- Com_update_multi
- Com_xa_commit
- Com_xa_end
- Com_xa_prepare
- Com_xa_recover
- Com_xa_rollback
- Com_xa_start
- Compression
- Connections
- Created_tmp_disk_tables
- Created_tmp_files
- Created_tmp_tables
- Delayed_errors
- Delayed_insert_threads
- Delayed_writes
- Flush_commands

- Handler_commit
- Handler_delete
- Handler_discover
- Handler_prepare
- Handler_read_first
- Handler_read_key
- Handler_read_last
- Handler_read_next
- Handler_read_prev
- Handler_read_rnd
- Handler_read_rnd_next
- Handler_rollback
- Handler_savepoint
- Handler_savepoint_rollback
- Handler_update
- Handler_write
- Innodb_buffer_pool_pages_data
- Innodb_buffer_pool_bytes_data
- Innodb_buffer_pool_pages_dirty
- Innodb_buffer_pool_bytes_dirty
- Innodb_buffer_pool_pages_flushed
- Innodb_buffer_pool_pages_free
- Innodb_buffer_pool_pages_misc
- Innodb_buffer_pool_pages_total
- Innodb_buffer_pool_read_ahead_rnd
- Innodb_buffer_pool_read_ahead
- Innodb_buffer_pool_read_ahead_evicted
- Innodb_buffer_pool_read_requests
- Innodb_buffer_pool_reads
- Innodb_buffer_pool_wait_free
- Innodb_buffer_pool_write_requests
- Innodb_data_fsyncs
- Innodb_data_pending_fsyncs
- Innodb_data_pending_reads
- Innodb_data_pending_writes
- Innodb_data_read
- Innodb_data_reads
- Innodb_data_writes
- Innodb_data_written
- Innodb_dblwr_pages_written
- Innodb_dblwr_writes
- Innodb_have_atomic_builtins
- Innodb_log_waits

- Innodb_log_write_requests
- Innodb_log_writes
- Innodb_os_log_fsyncs
- Innodb_os_log_pending_fsyncs
- Innodb_os_log_pending_writes
- Innodb_os_log_written
- Innodb_page_size
- Innodb_pages_created
- Innodb_pages_read
- Innodb_pages_written
- Innodb_row_lock_current_waits
- Innodb_row_lock_time
- Innodb_row_lock_time_avg
- Innodb_row_lock_time_max
- Innodb_row_lock_waits
- Innodb_rows_deleted
- Innodb_rows_inserted
- Innodb_rows_read
- Innodb_rows_updated
- Innodb_truncated_status_writes
- Key_blocks_not_flushed
- Key_blocks_unused
- Key_blocks_used
- Key_read_requests
- Key_reads
- Key_write_requests
- Key_writes
- Last_query_cost
- Max_used_connections
- Not_flushed_delayed_rows
- Open_files
- Open_streams
- Open_table_definitions
- Open_tables
- Opened_files
- Opened_table_definitions
- Opened_tables
- Performance_schema_cond_classes_lost
- Performance_schema_cond_instances_lost
- Performance_schema_file_classes_lost
- Performance_schema_file_handles_lost
- Performance_schema_file_instances_lost
- Performance_schema_locker_lost

- Performance_schema_mutex_classes_lost
- Performance_schema_mutex_instances_lost
- Performance_schema_rwlock_classes_lost
- Performance_schema_rwlock_instances_lost
- Performance_schema_table_handles_lost
- Performance_schema_table_instances_lost
- Performance_schema_thread_classes_lost
- Performance_schema_thread_instances_lost
- Prepared_stmt_count
- Qcache_free_blocks
- Qcache_free_memory
- Qcache_hits
- Qcache_inserts
- Qcache_lowmem_prunes
- Qcache_not_cached
- Qcache_queries_in_cache
- Qcache_total_blocks
- Queries
- Questions
- Rpl_status
- Select_full_join
- Select_full_range_join
- Select_range
- Select_range_check
- Select_scan
- Slave_heartbeat_period
- Slave_open_temp_tables
- Slave_received_heartbeats
- Slave_retried_transactions
- Slave_running
- Slow_launch_threads
- Slow_queries
- Sort_merge_passes
- Sort_range
- Sort_rows
- Sort_scan
- Ssl_accept_renegotiates
- Ssl_accepts
- Ssl_callback_cache_hits
- Ssl_cipher
- Ssl_cipher_list
- Ssl_client_connects
- Ssl_connect_renegotiates

- Ssl_ctx_verify_depth
- Ssl_ctx_verify_mode
- Ssl_default_timeout
- Ssl_finished_accepts
- Ssl_finished_connects
- Ssl_session_cache_hits
- Ssl_session_cache_misses
- Ssl_session_cache_mode
- Ssl_session_cache_overflows
- Ssl_session_cache_size
- Ssl_session_cache_timeouts
- Ssl_sessions_reused
- Ssl_used_session_cache_entries
- Ssl_verify_depth
- Ssl_verify_mode
- Ssl_version
- Table_locks_immediate
- Table_locks_waited
- Tc_log_max_pages_used
- Tc_log_page_size
- Tc_log_page_waits
- Threads_cached
- Threads_connected
- Threads_created
- Threads_running
- Uptime
- Uptime_since_flush_status

Nagios

Nagios is an open source monitoring solution that many companies leverage before moving to a SaaS solution. The FortiMonitor Agent for Linux can read the execution exit codes of Nagios check plugins which will then allow you to configure thresholds for Nagios plugins. The Agent also discovers all of the Nagios plugins on the server and reports them as individual metrics that can be added to FortiMonitor (One for the overall status returned by the plugin, and several metrics for any of the performance metrics that are exported). See Plugin status and performance data for more information on how FortiMonitor handles Nagios plugin output data.



Command line arguments for Nagios are currently not supported.

Agent configuration

This requires the latest version of the Linux Agent.

By default, the Agent will look for your Nagios plugins in **/usr/share/fm-agent/nagios.** If you are dropping your plugins in this directory, no other Agent configuration is necessary. If you would like the FortiMonitor Agent to look for your plugins in a different directory, you can provide a directory path in the FortiMonitor agent config file. To do so, edit the FortiMonitor agent config file, which is located at **/etc/fm-agent/fm_agent.cfg**, and add a block like the one below.

```
[nagios]
plugins_location = /path/to/your/plugins
```



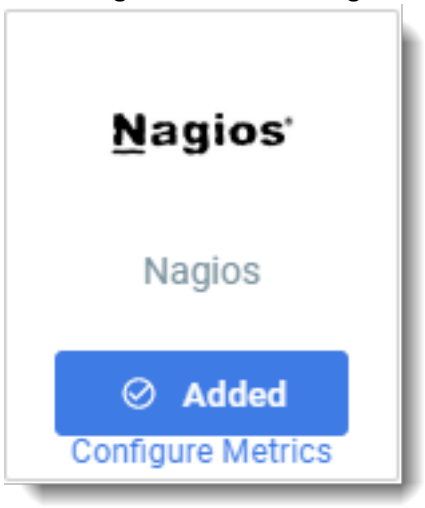
You need to ensure that the **fortimonitor-agent** user has permission to execute the plugin, including any potential admin privileges needed to access the plugin data.

Once the configuration block has been added, rebuild the agent metadata (via the control panel or with the **python /usr/bin/fm-agent/fm_agent.py --rebuild-metadata** command)

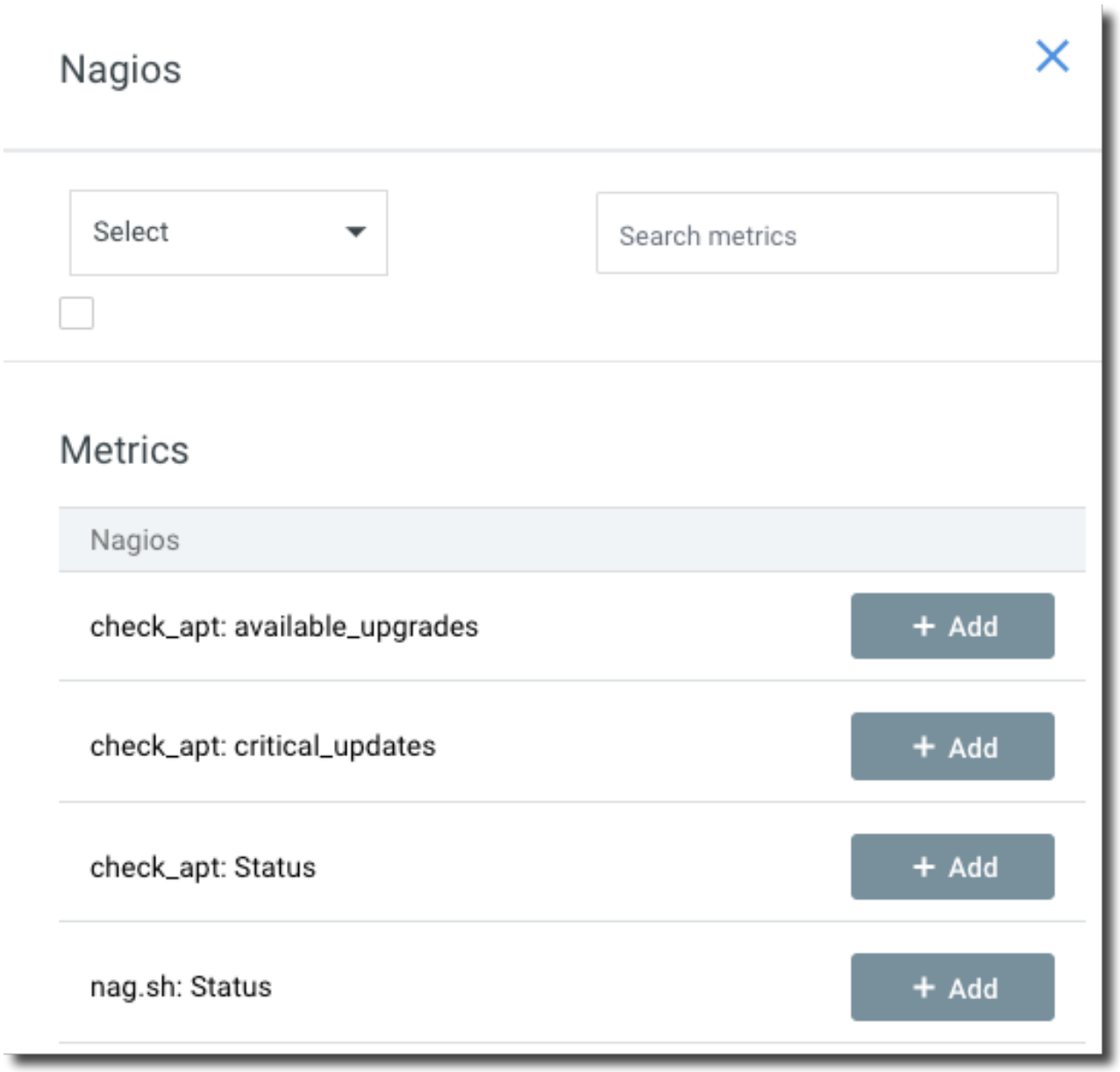
Control panel configuration

To start monitoring Nagios plugins:

1. Select **Add Monitoring** from the **Monitoring Config** tab of your server's **Instance Details** page.
   A catalog with all available services and applications will be displayed.
   Select **Nagios** then click **Configure Metrics**.



2. A configuration drawer with a list of discovered plugins will slide out.



3. Select a Nagios plugin from the list then click **Add**.
   The metric configuration drawer will slide out.
4. Configure the metric's settings, thresholds, and CounterMeasures.
5. Select **Save**.

In the **Thresholds & CounterMeasures** tab of the metric configuration drawer, you can configure incidents to be created based on these returned values. For example, create an incident if the metric value is > 1.

Nagios exit codes

Nagios uses a check plugin to monitor to identify the status of monitored services. When the plugin is run, Nagios can distinguish the status of the service by reading the exit code. These exit codes are then used by FortiMonitor to determine the status of the service. See the following table for more information.

Nagios exit code	Nagios value	FortiMonitor severity	Host state
0	OK		Up
1	WARNING	Warning	Up or down/unreachable
2	CRITICAL	Critical	Down/unreachable
3	UNKNOWN	Critical	Down/unreachable

Plugin status and performance data

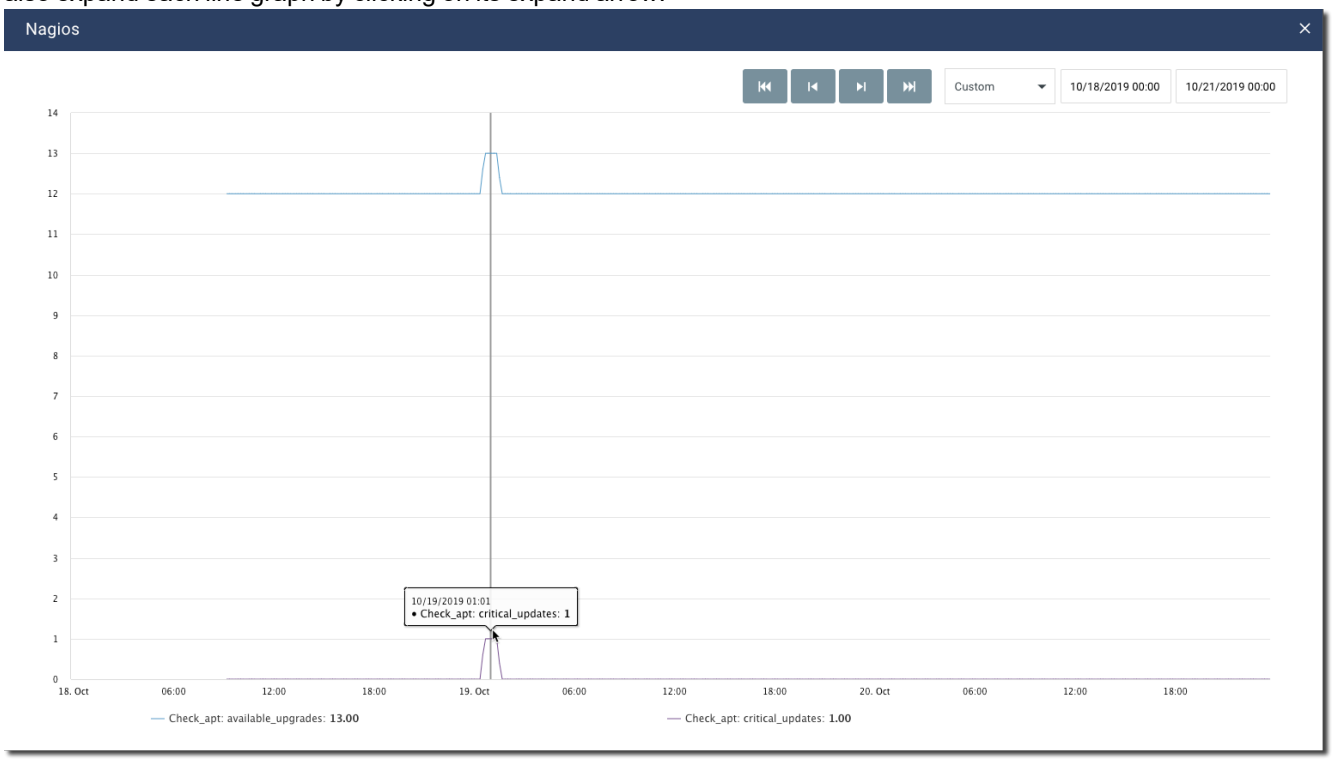
When a plugin is executed, the plugin returns a text that includes the status of the service and if available, the metric performance data. The first part of the text is a human-readable text string that describes the status of the service. The pipe (|), is followed by a text containing one or more performance metrics.

```
PING ok - Packet loss = 0%, RTA = 0.80 ms | percent_packet_loss=0, rta=0.80
```

To learn more about the format of a Nagios plugin output, see this article.

Performance monitoring

The **Performance** section provides a visual representation of the status of the Nagios plugins. Each metric is represented by a line graph that can be filtered based on metric type and scoped using a specific time range. You can also expand each line graph by clicking on its expand arrow.



Troubleshooting

All activities related to Nagios plugins are available in the **Activity** tab of the server's **Instance Details** page. Check this tab if you are having issues with your Nagios integration.

You can also check the FortiMonitor Agent log by entering the following command into a command window:

```
less /var/log/fm-agent/fm-agent.log
```

Nginx

The Nginx plugin lets you monitor a number of metrics on your reverse proxy server. If you can not find this plugin in the **/usr/lib/fm-agent/plugins** folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

Nginx Configuration

In order to monitor Nginx, you will need to edit the default site file. This file will be found in the **/etc/nginx/sites-enabled/** folder. Only Ubuntu/Debian flavors have these folders out of the box. By default, CentOS does not create these folders. If you are using **nginx-conf** to set the configuration for each of your websites, enable **Nginx Status Page** on each one by updating the code inside the server { } block as follows:

```
location /nginx_status {
stub_status on;
allow 127.0.0.1;
deny all;
access_log off;
}
```

Sample Nginx configuration file

```
server {
        listen       80 default_server;
        listen       [::]:80 default_server;
        server_name  _;
        root         /usr/share/nginx/html;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;
        # include /etc/nginx/sites-enabled/*.conf;
        location / {
        }

        error_page 404 /404.html;
            location = /40x.html {
```

```
        }

        error_page 500 502 503 504 /50x.html;
            location = /50x.html {
        }
       location /nginx_status {
       stub_status on;
       allow 127.0.0.1;
       deny all;
       access_log off;
       }

}
```

Log File Access

To monitor 2xx / 3xx / 4xx request count, the FortiMonitor Agent user needs to be able to read the server's Nginx access log file. It is typically found at **/var/log/nginx/access.log**, though you can consult your nginx.conf file for confirmation. It's typically located at **etc/nginx/nginx.conf**, though it may also be at **/usr/local/nginx/conf/nginx.conf** or **/usr/local/etc/nginx/nginx.conf**.

Agent Configuration

You can optionally override a set of parameters related to your Nginx installation.

- console_url: Base location of your Nginx status endpoint - typically **http://127.0.0.1**, which will be composed into **http://127.0.0.1/nginx_status**
- log_location: Path to your Nginx access.log - defaults to **/var/log/nginx/access.log**

```
[nginx]
console_url = http://127.0.0.1
log_location = /var/log/nginx/access.log
```

Control Panel Configuration

If you have not already, read the Add FortiMonitor Agent checks article to learn how to set up monitoring for any Agent plugin. To use the Nginx plugin, first select **Nginx** from the monitoring catalog.

There are a number of metrics available to monitor.

Metric	Unit
Requests per second	requests per second
Accepted connections	connections

Handled connections	connections
Active connections	connections
Request handles	requests
Read request body	
Read request header	
2xx's	requests
4xx's	requests
5xx's	requests

NodeJS

The NodeJS Agent plugin allows you to monitor a number of metrics on your NodeJS application. The plugin's metrics can help your team detect failures like heap total and high resolution time, to see if your application is performant. If you can not find this plugin in the /usr/lib/fm-agent/plugins folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

If you have not already, read the Add FortiMonitor Agent checks article to learn how to set up monitoring for any Agent plugin.

Control Panel Configuration

To use the NodeJS plugin, first select NodeJS from the monitoring catalog.

A drawer will slide out showing the 5 available metrics.

- Cluster workers
- High resolution time
- Resident set size
- Total heap
- Used heap

If you would like to inspect the NodeJS plugin in more detail you can find the python script for it in the /usr/lib/fm-agent/plugins folder.

If you have followed the steps above and the new metric and plugin does not show properly, you may need to manually rebuild the agent's metadata. You can do this by following the guide here.

Note: NodeJS plugin is currently only available for Linux.

NTP (Linux)

Agent Configuration

If you would like to change the NTP server you are using for reference, then you can add the following section to your Agent config file (**/etc/fm-agent/fm-agent.cfg**).

```
[ntp]
ntp_host = pool.ntp.org
ntp_port = 123
```

Just replace the host name and port with the appropriate values for the NTP server that you are using.

Control Panel Configuration

The NTP plugin allows you to monitor the disagreement between your machine's time, and the NTP time.

To add an NTP Agent resource to your server first see Add FortiMonitor Agent checks if you have not already.

First select "Linux" catalog and choose NTP from the list.

The plugin will only have one option; "Difference between machine and NTP time." This resource returns the number of seconds, please take this into account if you choose to change your alert thresholds.

Metric	Unit
Difference between machine and NTP time	seconds

If you would like to inspect the NTP plugin in more detail you can find the python script for it in **/usr/lib/fm-agent/plugins/ntp.py**.

NTP (Windows)

The NTP plugin allows you to monitor the disagreement between your machine's time, and the NTP time.

Agent Configuration

If you would like to change the NTP server you are using for reference, then you can change the NTP section to your Agent config file (**C:/Program Files/FortiMonitorAgent/Agent.config**).

Just replace the host name and port with the appropriate values for the NTP server that you are using.

```
<ntp>
<add key="ntp_host" value="pool.ntp.org" />
<add key="ntp_port" value="123" />
</ntp>
```

Control Panel Configuration

To add an NTP agent resource to your server first see Add FortiMonitor Agent checks if you have not already.

Select NTP from the Windows plugin.

The “Metric” drop-down will only have one option; “Difference between machine and NTP time.” This resource returns the number of seconds, please take this into account if you choose to change your “Alerts Thresholds.”

Metric	Unit
Difference between machine and NTP	seconds

If you would like to change the NTP server you are using for reference, then you can change the NTP section to your Agent config file (**C:/Program Files/FortiMonitorAgent/Agent.config**).

Oracle WebLogic (Linux)

The Oracle WebLogic plugin for Linux allows you to monitor a number of performance and availability metrics for your WebLogic application server. If you can not find this plugin in the **/usr/lib/fm-agent/plugins** folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

Oracle WebLogic Configuration

Once you install the agent you will need to add your WebLogic username, password, and WebLogic home to the **/etc/fm-agent/fm_agent.cfg** file as shown below:

```
[weblogic]
username = weblogic
password = password123
wl_home = /opt/oracle/Oracle/Middleware/Oracle_Home/wlserver
```

Control Panel Configuration

First, select **Oracle WebLogic** from the catalog.

There are a number of available metrics that can be summed into 5 groups.

- EJB Pool
- EJB Transaction
- Executive Queue
- Servlet
- Web App Component

Metric	Units
Average Time of All Invocations the Servlet has Executed since Created	seconds
Bean Instances Currently being Used from the Free Pool	beans
Current Available Bean Instances in the Free Pool	beans

Duration of Single Longest Invocation the Servlet has Executed since Created	seconds
Duration of Single Shortest Invocation the Servlet has Executed since Created	seconds
EJB Pool: The current number of available bean instances in the free pool	beans
EJB Pool: The number of bean instances currently being used from the free pool	beans
EJB Pool: The number of Threads currently waiting for an available bean instance from the free pool	threads
EJB Pool: The total number of available bean instances in the free pool	beans
EJB Pool: The total number of bean instances currently in use from the free pool	beans
EJB Pool: The total number of Threads currently waiting for an available bean instance from the free pool	threads
EJB Pool: The total number of Threads that have timed out waiting for an available bean instance from the free pool	threads
EJB Pool: The total number of times a bean instance from this pool was destroyed	
EJB Pool: The total number of times a bean instance from this pool was destroyed due to a non-application Exception being thrown from it	
EJB Pool: The total number of times a bean instance from this pool was destroyed due to a non-application Exception being thrown from it	
EJB Pool: The total number of times a failed attempt was made to get an instance from the free pool	beans
EJB Pool: The total number of times an attempt was made to get an instance from the free pool	
EJB Transaction: The total number of transactions that have been committed for this EJB	transactions
EJB Transaction: The total number of transactions that have been rolled back for this EJB	transactions
EJB Transaction: The total number of transactions that have timed out for this EJB	transactions
Execute Threads Assigned to the Queue	threads
Executive Queue: The number of idle threads assigned to the queue	thread
Executive Queue: The number of requests that have been processed by the queue	requests
Executive Queue: The number of waiting requests in the queue	requests
Executive Queue: The time since the longest waiting request was placed in the queue	ms

Executive Queue: The total number of execute threads assigned to the queue	threads
Highest Total Number of Open Sessions	sessions
Idle Threads Assigned to the Queue	threads
Maximum Capacity of this Servlet for Single Thread Model Servlets	threads
Number of Bean Instances Currently in use from the Free Pool	beans
Open Sessions in Module	sessions
Requests Processed by the Queue	requests
Requests that have been Processed	requests
Servlet: The amount of time the single longest invocation of the servlet has executed since created	
Servlet: The amount of time the single shortest invocation of the servlet has executed since created	
Servlet: The average amount of time all invocations of the servlet have executed since created	
Servlet: The maximum capacity of this servlet for single thread model servlets	threads
Servlet: The total amount of time all invocations of the servlet have executed since created	
Servlet: The total count of the number of times this servlet has been reloaded	
Servlet: The total count of the times this servlet has been invoked	
The number of connections currently in use by applications	connections
The number of requests that have been processed	requests
The number of threads that are considered to be stuck on the basis of any stuck thread constraints	threads
The number of waiting requests in the queue	requests
Threads Considered Stuck on the Basis of any Stuck Thread Constraints	threads
Threads Currently Waiting for an Available Bean Instance from the Free Pool	threads
Time all Invocations of the Servlet have Executed since Created	seconds
Time since the Longest Waiting Request was Placed in the Queue	seconds
Times this Servlet has been Invoked	seconds
Times this Servlet has been Reloaded	reloads
Total Attempts Made to get an Instance from the Free Pool	attempts
Total Attempts to Access a Bean from the Cache	attempts
Total Attempts to Access a Bean from the Cache Failed	attempts
Total Attempts to Access a Bean from the Cache Succeeded	attempts

Total Available Bean Instances in the Free Pool	beans
Total bean instances from this Pool that were Destroyed Due to a Non-Application Exception being Thrown from it	beans
Total Beans from this EJB Home Currently in the EJB Cache	beans
Total Beans from this EJB Home that have been Activated	beans
Total Beans from this EJB Home that have been Passivated	beans
Total Failed Attempts to get an Instance from the Free Pool	attempts
Total Number of Sessions Opened	sessions
Total Threads Timed Out Waiting for an Available Bean Instance from the Free Pool	threads
Total Threads Waiting for an Available Bean Instance from the Free Pool	threads
Total Transactions Committed for this EJB	trans
Total Transactions Rolled Back for this EJB	trans
Total Transactions Timed Out for this EJB	trans
Waiting Requests in the Queue	requests
Web App Component: The current total number of open sessions in this module	sessions
Web App Component: The high watermark of the total number of open sessions in this server	sessions
Web App Component: The total number of sessions opened	sessions
WebLogic Connections Currently in use by Applications	connections
Work Manager: The total number of attempts to access a bean from the cache	
Work Manager: The total number of beans from this EJB Home currently in the EJB cache	beans
Work Manager: The total number of beans from this EJB Home that have been activated	beans
Work Manager: The total number of beans from this EJB Home that have been passivated	beans
Work Manager: The total number of times an attempt to access a bean from the cache failed	
Work Manager: The total number of times an attempt to access a bean from the cache succeeded	

If you would like to inspect the Oracle plugin in more detail you can find the python script for it in the /usr/lib/fm-agent/plugins folder.

Oracle WebLogic 12c

Once you install the agent you will need to add your **weblogic12c** configuration block to the **/etc/fm-agent/fm_agent.cfg**file as shown below:

```
[weblogic12c]
classpath = (REQUIRED) A colon-separated Java CLASSPATH specification that points to the
     wlclient.jar and wljmxclient.jar files. These are installed with WebLogic and are
     generally found at $WL_HOME/wlserver/server/lib. If permissions issues prevent the
     agent user from accessing them in their default location, they can be copied to
     another location available to the agent such as $AGENT_HOME/share and referenced there
     in the classpath variable.
username = (OPTIONAL) If both username and password are present, used to authenticate the
     JMX API call
password = (OPTIONAL) If both username and password are present, used to authenticate the
     JMX API call
protocol = (OPTIONAL) The protocol used to access the JMX API. If not provided, defaults to
     iiop.
hostname = (OPTIONAL) The hostname used for the JMX connection. If not provided, defaults to
     localhost.
port = (OPTIONAL) The port to use for the JMX connection. If not provided, defaults to 7001.
jndi_name = (OPTIONAL) The Java Naming and Directory Interface name used to fine the
     WebLogic management beans. If not provided, defaults to
     weblogic.management.mbeanservers.runtime
jmx_url = (OPTIONAL) The full JMX URL used to access the WebLogic JMX API.
```

If not provided, this URL is built up from the previous configuration items.

Oracle Weblogic (Windows)

You can use the JMX plugin for Windows to monitor several performance and availability metrics on your Oracle WebLogic application server.

Enable JMX for WebLogic

1. To enable JMX for WebLogic monitoring, add the following lines to **C:\Oracle\Middleware\Oracle_Home\user_projects\domains\base_domain\bin\setDomainEnv.cmd**.

```
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.port=8888
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.ssl=false
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.access.file=%JRE_
HOME%\lib\management\jmxremote.access
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.password.file=%JRE_
HOME%\lib\management\jmxremote.password
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.authenticate=true
```

2. Create the **jmxremote.access** and **jmxremote.password** files in **C:\Program files\Java\jdk1.8.0_241\jre\lib\management**.

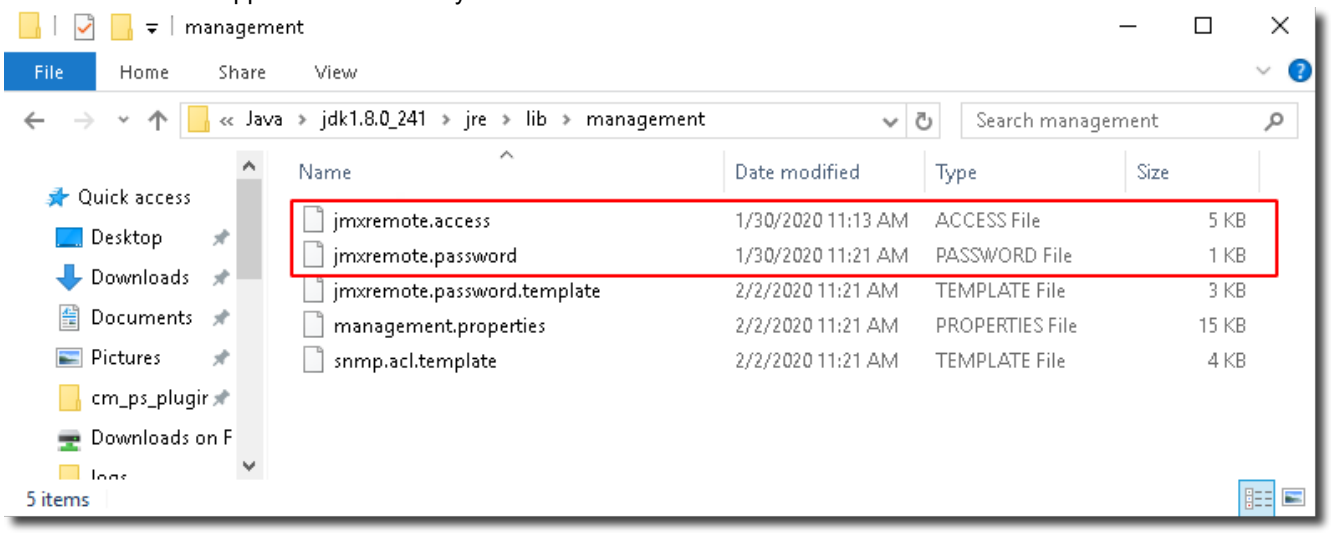
jmxremote.access – The file should be created as follows.

```
# The "monitor" role has readonly access.
# The "admin" role has readwrite access.
monitor readonly
admin readwrite
```

jmx.password – The file should be created as shown in the example below. Set permission to **read-only**.

```
# specify actual password instead of the text password
monitor password
admin password
```

3. Both files should appear in the directory as shown below:

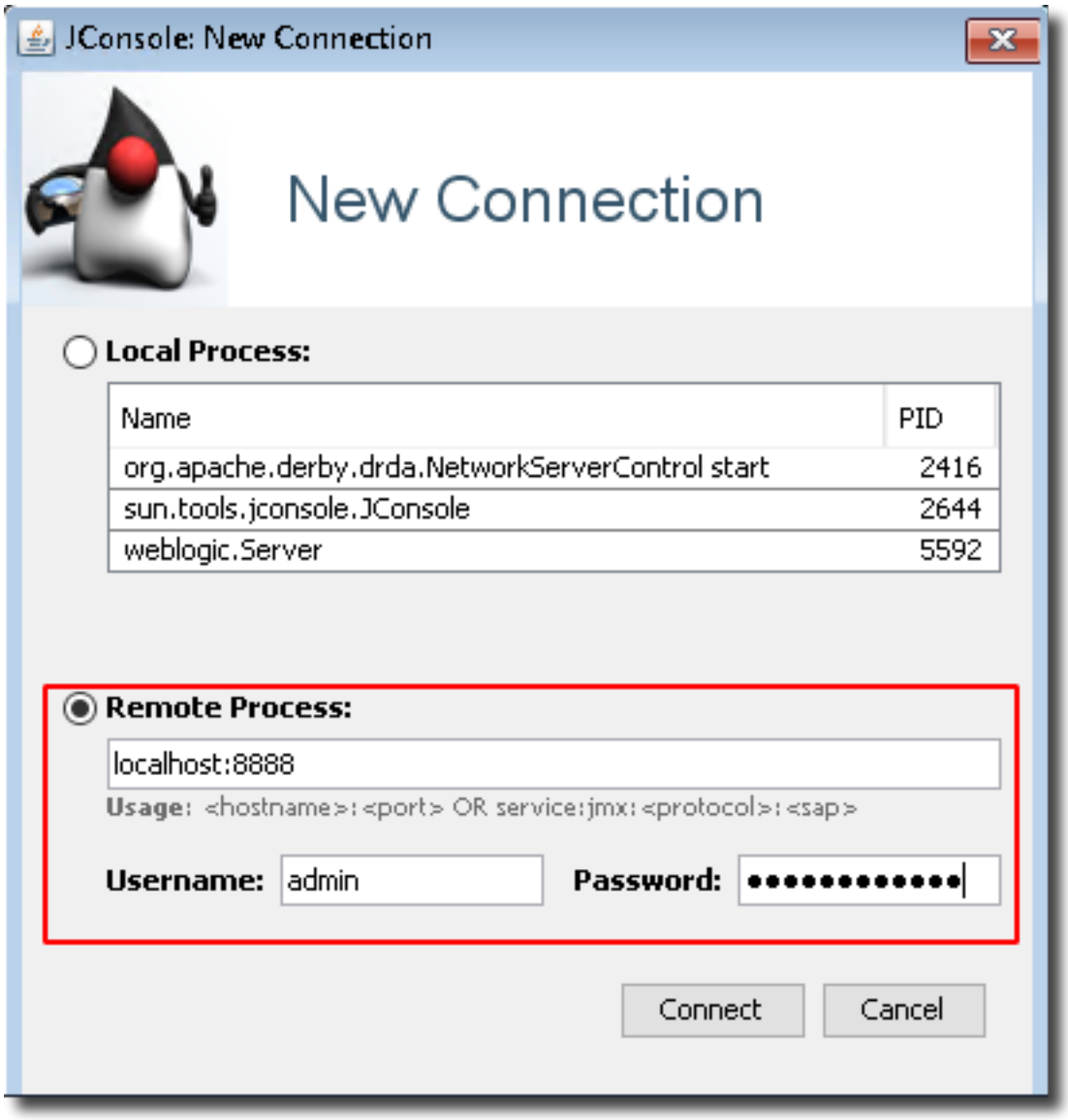


4. Open PowerShell and run **startWebLogic.cmd**.

a. The path is **C:\Oracle\Middleware\Oracle_Home\user_projects\domains\base_domain**.

b. Review the command output for errors. If an error is encountered, resolve the error before continuing.

5. Check MBeans with JConsole.

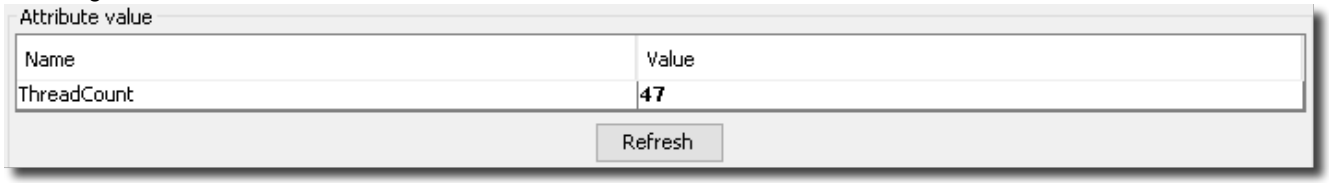


a. Open JConsole from the command prompt.

b. Select **Remote Process** then enter the **<hostname>:<port>**. Use the port configured while enabling JMX in **setDomainEnv.cmd**.

c. Use the **Username** and **Password** set on the **jmxremote.password** file.

6. Click **Connect**. Once connected, click the MBeans tab and verify that the data values match the values in the following:



Agent configuration

After installing the Windows Agent, edit the Agent configuration file to add your JMX credentials. The steps are detailed in JMX (Windows) Agent configuration.

Control Panel configuration

The next step is to add and configure the JMX plugin.

1. Add JMX from the Monitoring Catalog.
2. Add the **Custom JMX Metric**.
3. Enter the following information:
   FieldValuemBean stringEnter java.lang:type=Threading/ThreadCountMetric Name OverrideEnter Thread Count
4. Click **Save**.
5. Verify that the FortiMonitor Agent log has the following output.

```
[2020-02-02 22:43:48,807] [MetricLogger] [INFO] [123] - <JMX> First Output variable: 02/02/2020
22:43:48 -0600 org.archive.jmx.Client ThreadCount: 47
[2020-02-02 210002:43:48,807] [MetricLogger] [INFO] [123] - <JMX> Second Output variable:
ThreadCount: 47
[2020-02-02 22:43:48,807] [MetricLogger] [INFO] [123] - JMX Plugin: JMX Process completed
[2020-02-02 22:43:48,807] [MetricLogger] [INFO] [123] - <JMX> Checking if key is empty
[2020-02-02 22:43:48,807] [MetricLogger] [INFO] [123] - <JMX> Key is empty - value: 47
[2020-02-02 22:43:48,807] [MetricLogger] [INFO] [123] - <JMX> Key is empty - value: 47
```

Oracle (Linux)

The Oracle plugin allows you to monitor a number of metrics that provide insight into the performance, capacity, and health of your database server. If you can not find this plugin in the **/usr/lib/fm-agent/plugins** folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

Oracle Configuration

The user that is specified above for the agent needs specific permissions in order to access the metrics from Oracle. The user should be created in Oracle with the following grants:

```
CREATE USER <USERNAME> IDENTIFIED BY "<PASSWORD>";
GRANT CREATE SESSION TO <USERNAME>;
GRANT SELECT ON dba_data_files TO <USERNAME>;
GRANT SELECT ON dba_segments TO <USERNAME>;
GRANT SELECT ON dba_free_space to <USERNAME>;
GRANT SELECT ON v_$resource_limit to <USERNAME>;
```

Agent Configuration

Once the agent is installed you will need to add the following information to the agent config file (/etc/fm-agent/fm_agent.cfg):

```
[oracle]
oracle_home = /u01/app/oracle/product/11.2.0/dbhome_1
oracle_sid = database_name
username = username
password = password
tns_listener_ip = 1.2.3.4
```

You should replace these variables with the appropriate values. Of these variables, all are required except for tns_listener_ip, which defaults to 127.0.0.1 if it is left out. oracle_home should be set to the same value as the ORACLE_HOME environment variable that is normally set in your shell. The oracle_sid can be set as a comma separated list if you would like to monitor multiple databases.

Control Panel Configuration

After making these changes give the agent a few minutes to connect with FortiMonitor's servers then you should see new server resources available to add for your server in the control panel.

If you have not already, read the Add FortiMonitor Agent checks article to learn how to set up monitoring for any agent plugin. First select "Oracle" from the monitoring catalog.

There are a number of metrics to choose from, and are listed below

Metric	Unit
Current number of Maximum Shared Servers	
Current number of transactions	
Free space of tablespace in MB	MB
Maximum number of enqueue locks	
Maximum number of enqueue resources	
Maximum number of Global Enqueue Service locks	
Maximum number of Global Enqueue Service processes	
Maximum number of Global Enqueue Service resources	
Maximum number of Maximum Shared Servers	
Maximum number of processes	
Maximum number of sessions	
Maximum number of transactions	
Number of current enqueue locks	

Number of current enqueue resources	
Number of current Global Enqueue Service locks	
Number of current Global Enqueue Service processes	
Number of current processes	
Number of current sessions	
Number of Global Enqueue Service resources	
Percent of tablespace free	%
Size of tablespace in MB	MB
TNS listener is active and reachable	boolean

Package Upgrades

The Package Upgrade Agent plugin allows you to monitor the number of days since your server was last updated and how many packages (either security on non-security related) are waiting to be installed. However it does require additional configuration if you want to be able to use the “Security-related packages waiting to be updated” and “Non-security-related packages waiting to be updated” metrics. To set up support for these metrics, the following is needed in the /etc/sudoers folder.

On Red Hat/CentOS:

```
Defaults:panopta-agent !requiretty
fm-agent ALL=(ALL) NOPASSWD: /usr/bin/yum, /usr/bin/zgrep
```

On Ubuntu/Debian:

```
fm-agent ALL=(ALL) NOPASSWD: /usr/bin/apt-get
```



The package_upgrade plugin is disabled by default in the panopta_agent.cfg file as seen here: plugin_blacklist = package_upgrade

To enable this package_ugrade plugin you will need to remove package_upgrade from the plugin_blacklist .

See the plugin_blacklist documentation for more details about the plugin_blacklist.

To add a package upgrade agent resource to your server first see Add FortiMonitor Agent checks if you have not already.

First select Linux from the monitoring catalog.

The metric drop-down will have either one or three options, depending on whether you chose to add the additional configuration.

- Check for Package Installation
- Days since the last package update was run
- Non-security-related package waiting to be updated
- Security-related packages waiting to be updated

If you choose to change your alert thresholds for this resource, please remember that this resource returns a value in days.

If you would like to inspect the Package Upgrades plugin in more detail you can find the python script for it in /usr/lib/fm-agent/plugins/package_upgrade.py.

Patches

The Patches plugin allows you to monitor the number of outstanding updates needed to be installed on your server.

To add a Patches resource type to your server first see Add FortiMonitor Agent checks if you have not already.

Control Panel Configuration

First select open the Windows metrics section, and choose "patches"

There are 4 possible metrics to monitor.

Metric	Units
Critical	Patches
Important	Updates
Last Update Days	Days
Optional	Updates

The Critical metric will return the number of critical patches. The Optional metric will return the number of optional patches yet to be installed, the Important metric returns the number of important patches yet to be installed, and the Last Update Days metric returns the number of days since the last update.

The specific number that this resource returns is defined by the above options, please take this into account if you choose to change your “Alert Threshold” for this resource.

You can monitor multiple Patches agent resources on one server.

PHP-FPM

FPM Configuration

In order for the PHP FPM plugin to work, you will need to enable the status module within your web server. Instructions will differ based on the operating system and webserver. An example of how it is done in Nginx can be found here.

Agent Configuration

You'll need to add PHP FPM configuration block into Agent's configuration file, which can be found at **/etc/fm-agent/fm_agent.cfg**. If you have multiple PHP FPM instances running on the server, use comma-separated values as shown below:

```
[phpfpm]
console_url = http://127.0.0.1/status, http://127.0.0.1/status
```

Control Panel Configuration

Like other application monitoring, PHP FPM monitoring is added to a server via the Monitoring Catalog. If you don't see PHP FPM as a discovered resource, you may need to Rebuild the FortiMonitor Agent metadata.

If you haven't yet read about how to add an Agent metric, please read our article Add FortiMonitor Agent checks.

Once PHP FPM is added, you can begin to monitor any of the metrics below.

Metric	Unit
Active Processes	Processes
Children Reached	Children
Idle Processes	Processes
Listen Queue	
Listen Queue len	Processes
Max Active Processes	Processes
Max Listen Queue	
Slow Requests	Requests
Start Since	
Start Time	
Total Processes	Processes

PHP OPCache

To configure the PHP OPCache metrics, Python 2.7 or above is required in addition to some additional setup steps which are outlined below.

OPCache Configuration

To Install:

- Copy the file from **/usr/lib/fm-agent/extras/opcache.php** into a location that can be served by your webserver.
- Important: Make sure to lock down that file so only the localhost can access it.
- Edit your FortiMonitor agent config file to include the path to the OPCache script (see config section below).

Agent Configuration

You'll need to add OPCache configuration block into Agent's configuration file, which can be found at /etc/fm-agent/fm_agent.cfg.

```
[opcache]
opcache_url = http://localhost/opcache-url-path/opcache.php
```

Control Panel Configuration

Like other application monitoring, OPCache monitoring is added to a server via the Monitoring Catalog. If you don't see OPCache as a discovered resource, you may need to Rebuild the FortiMonitor Agent metadata.

If you haven't yet read Add FortiMonitor Agent checks, please do that first.

Once OPCache is added, you can begin to monitor any of the metrics below.

Metric	Unit
Blacklist Misses	
Buffer Size	MB
Cache Hit Rate	%
Current % Wasted Cache	%
Hash Restarts	
Hits	
Manual Restarts	
Max Cached Keys	
Memory Free	MB
Memory Used	MB
Memory Wasted	MB
Misses	
Number of Cached Keys	
Number of Cached Scripts	
Number of Strings	
OOM Restarts	

PostgreSQL

The PostgreSQL plugin allows you to monitor a number of metrics about the PostgreSQL database running on your server. If you can not find this plugin in the **/usr/lib/fm-agent/plugins** folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

Please note that at this time the PostgreSQL plugin can only monitor one database and only supports **password** or **md5** authentication.

PostgreSQL Configuration

You will need to add a fm-agent user to PostgreSQL. You can do this by typing in your psql shell:

```
CREATE USER "fm-agent" WITH PASSWORD 'XXXXX';
```

You should also give this user access to the database you would like to monitor.

Agent Configuration

In addition you will need to add your PostgreSQL username and password to the **/etc/fm-agent/fm_agent.cfg** file as shown below:

```
[postgresql]
hostname = localhost
port = 5432
username = fm-agent
password = XXXXX
database = SomeDatabase
unix_sock = /tmp/.s.PGSQL.5432 (optional for unix socket based connections)
```

Make sure that the password in the configuration file matches that of the fm-agent user in PostgreSQL.



Unix socket connections are also supported. If you choose to use this option, the hostname parameter will be ignored.

Control Panel Configuration

After making this change, give the agent a few minutes to sync with our FortiMonitor's server then you should see new server resources available to add for the server in the control panel.

To add a PostgreSQL agent resource to your server first see Add FortiMonitor Agent checks if you have not already.

First select "PostgreSQL" from the monitoring catalog.

The metric drop-down will have the following options:

- Blocks read from buffer cache
- Blocks read from disk
- Buffer cache hit rate (%)
- DELETES
- INSERTS
- Number of active backends
- SELECTS (from index)
- SELECTS (from scans)
- Total queries
- Total transactions
- Transactions committed
- Transactions rolled back
- UPDATES
- Total database checkpoints
- Minutes between database checkpoints
- Total connections
- Active connections
- Idle connections
- Idle in transaction connections
- Connections waiting for lock
- Max transaction age
- Temp files generated
- Temp bytes generated

Processes (Linux)

The Processes plugin is an updated version of our “Process Name” plugin. You need to have the latest version of the Agent to use this plugin.

Control Panel Configuration

First, select Linux from the monitoring catalog and scroll to Processes.

There are a number of metrics to choose to monitor.

Metric	Unit
CPU % of processes - full command line	%
CPU % of processes - name	%
MB of memory used by processes - full command line	MB

Memory % of processes - full command line	%
Memory % of processes - name	%
Number of processes - full command line	processes
Number of processes - name	processes
Number of processes running	processes
Process is running	boolean
Process is running - full command line	boolean

You can monitor multiple Processes agent resources on one server if you would like, so you can keep track of all of your essential processes in this manner.

If you would like to inspect the process name plugin in more detail you can find the python script for it in **/usr/lib/fm-agent/plugins/process.py**.

Processes (Windows)

The Processes plugin allows you to monitor what processes are running on your server.

Control Panel Configuration

First select **Windows** and scroll to **Processes**.

There will be three options: Number of processes by name, Number of processes running and Process is running. For “Process is running” and “Number of processes by name,” you can enter in whatever process name you would like to check.

Metric	Unit
Number of processes by name	processes
Number of processes running	processes
Process is running	boolean

Important Note: The process name must be the **friendly** name, which is the process without the .exe extension.

The **Process is running** check will return a 1 if that process is running and 0 if it is down. Please do not forget this is a boolean check if you choose to change your **Alert Thresholds** for this resource.

You can monitor multiple Processes agent resources on one server if you would like, so you can keep track of all of your essential processes in this manner.

Prometheus

If you are already using Prometheus to monitor your instances, you can use the **Prometheus integration** to incorporate the metrics into your FortiMonitor infrastructure. Prometheus supports a number of exporters that enable FortiMonitor to

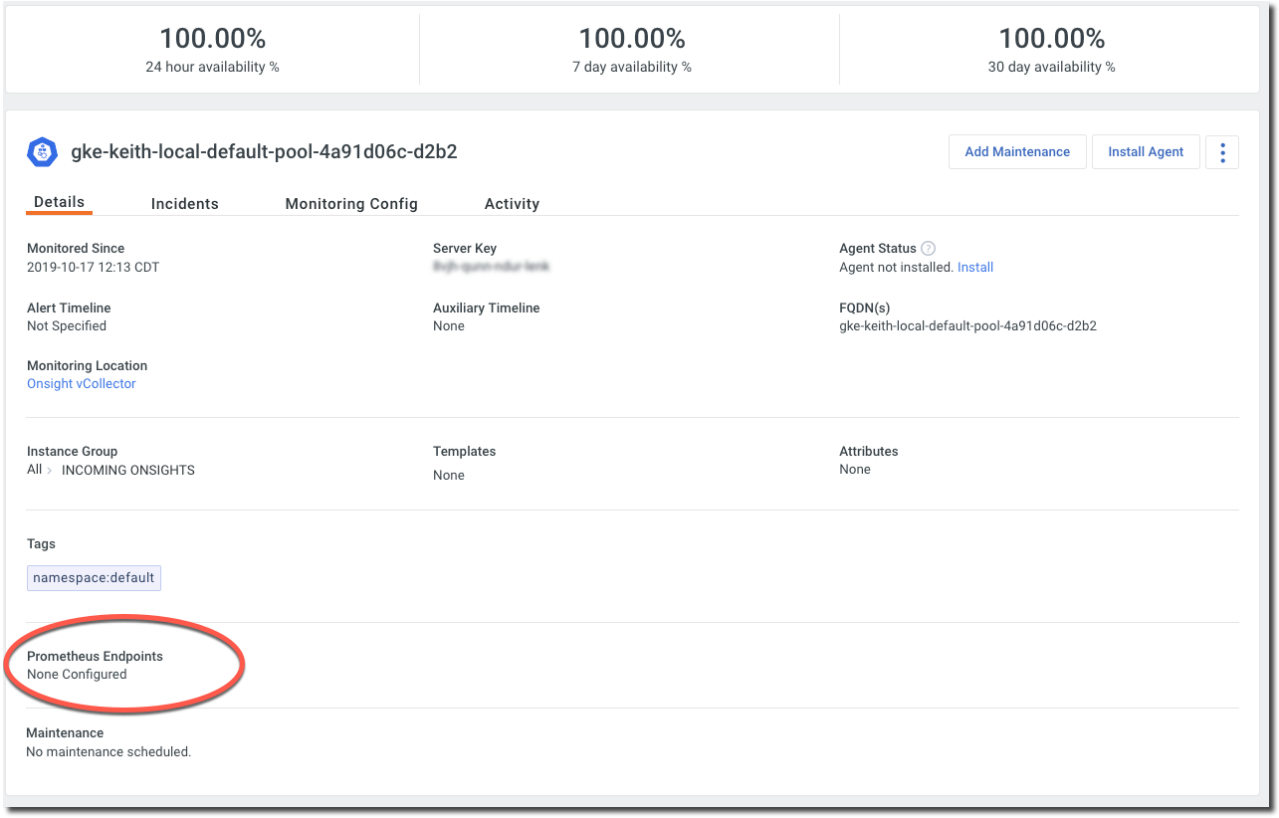
scrape metric data from exposed endpoints. These metrics are then visualized in configurable graphs to help you track each metric. Setting up an Alert Timeline also allows you to configure alerting and take action when an incident occurs.

Contact our support team if you want the Prometheus integration feature enabled.

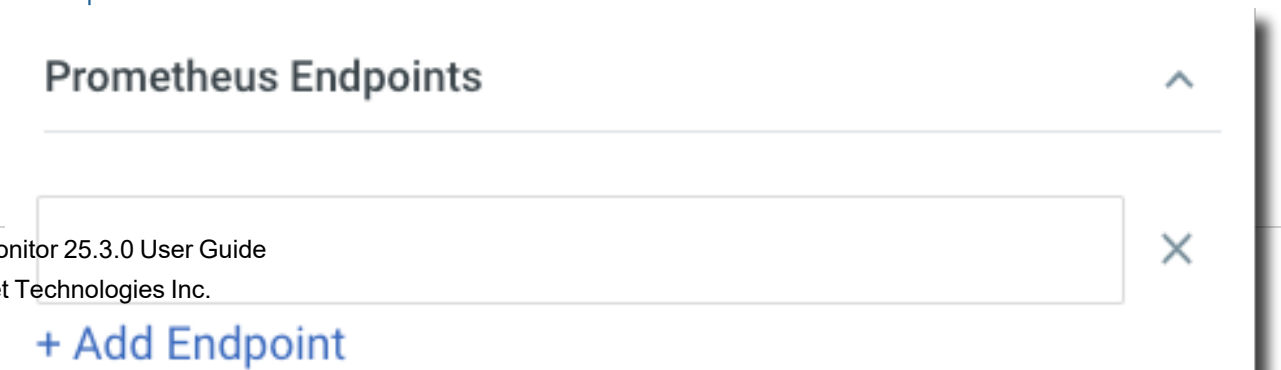
Control panel configuration

To start monitoring your Prometheus endpoints, perform the following steps:

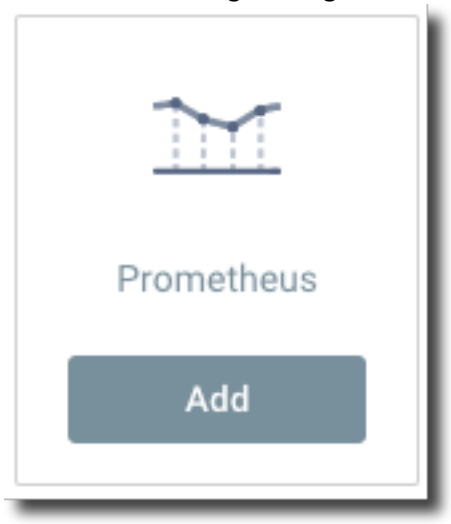
1. From the **Instance Details** page of the instance that you want to monitor, click **Prometheus Endpoints**.
Note: To allow FortiMonitor to scrape off data from Prometheus endpoints, the primary monitoring location of the instance must be set to an OnSight vCollector.



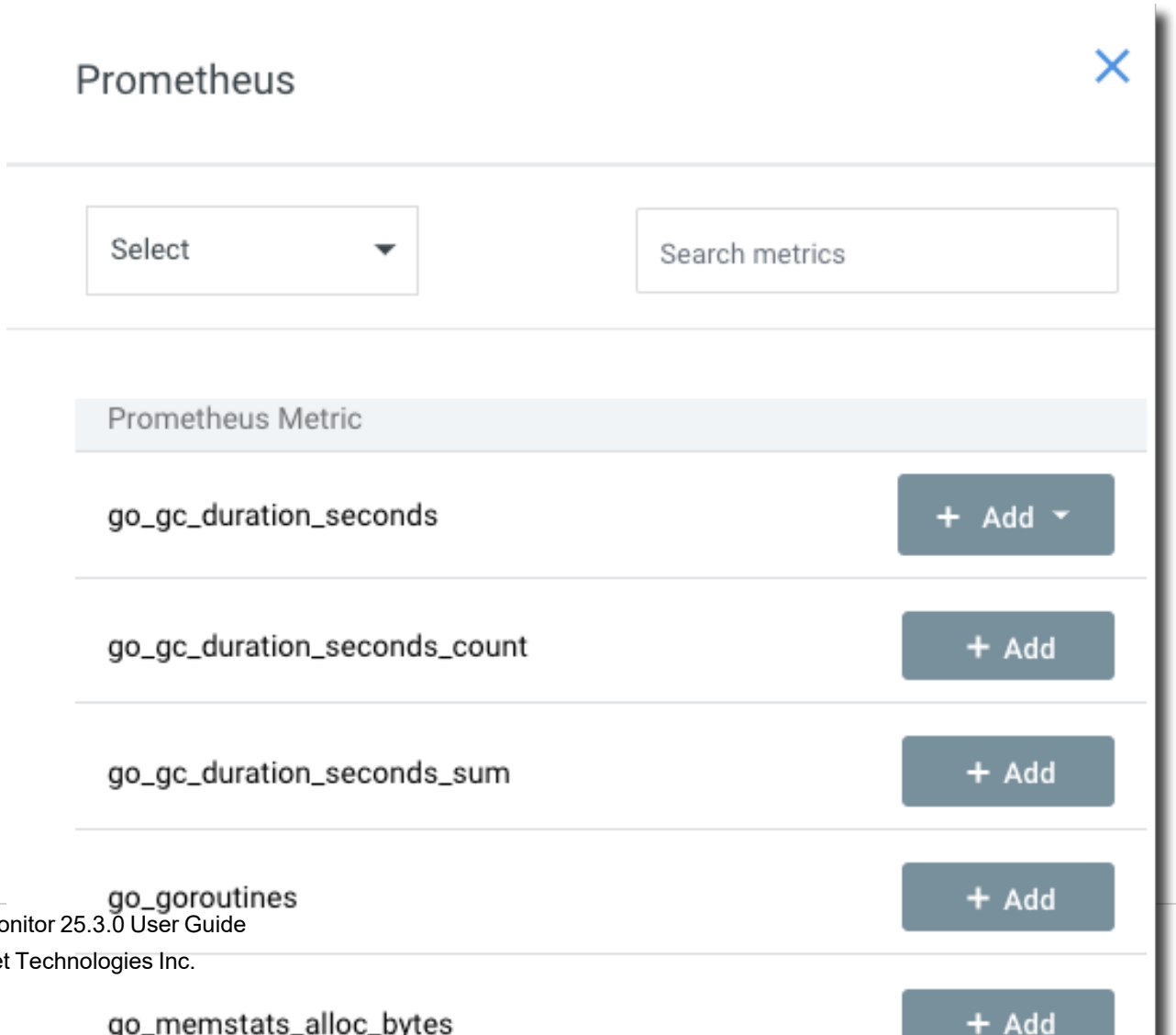
2. On the configuration drawer, select **Add Endpoint** then provide the endpoint. For example, **prometheus-rabbitmq-exporter-7fb45675b-nvqbk:9419/metrics**.
Note: 9419 is the default port for the RabbitMQ exporter. If you are unsure about which port to use, see Prometheus default port allocations.



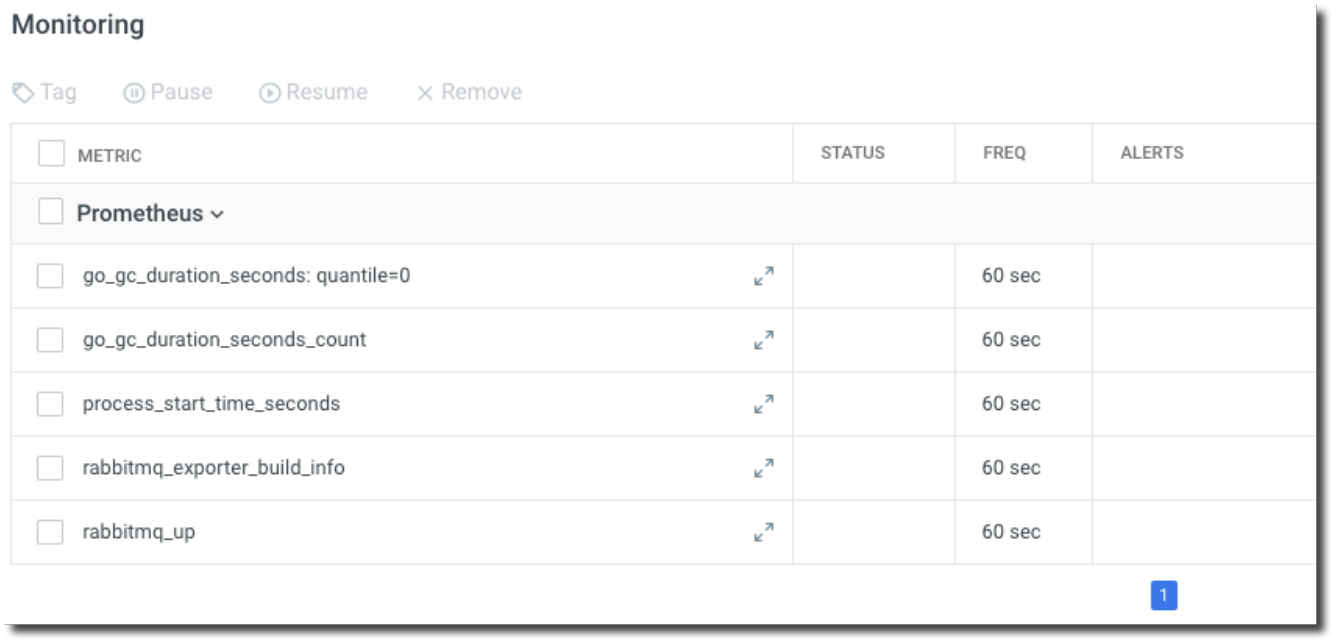
3. Select **Save Changes**.
4. Go to the **Monitoring Config** tab then click **Add Monitoring**.
   From the **monitoring catalog**, add **Prometheus**.



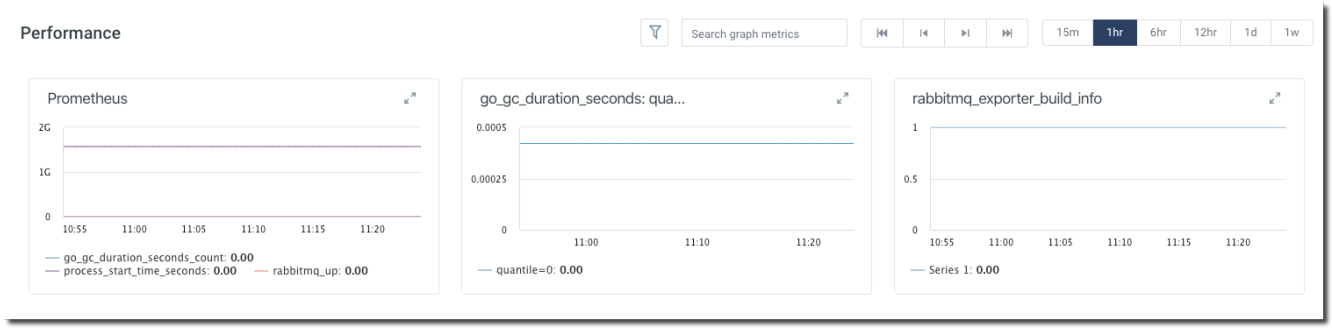
5. The metric drawer that contains all of the supported Prometheus metrics will slide out. Choose a metric then select **Add**.
   Note: After adding the endpoint, give the OnSight a few minutes to discover all available metrics and populate the metric drawer.



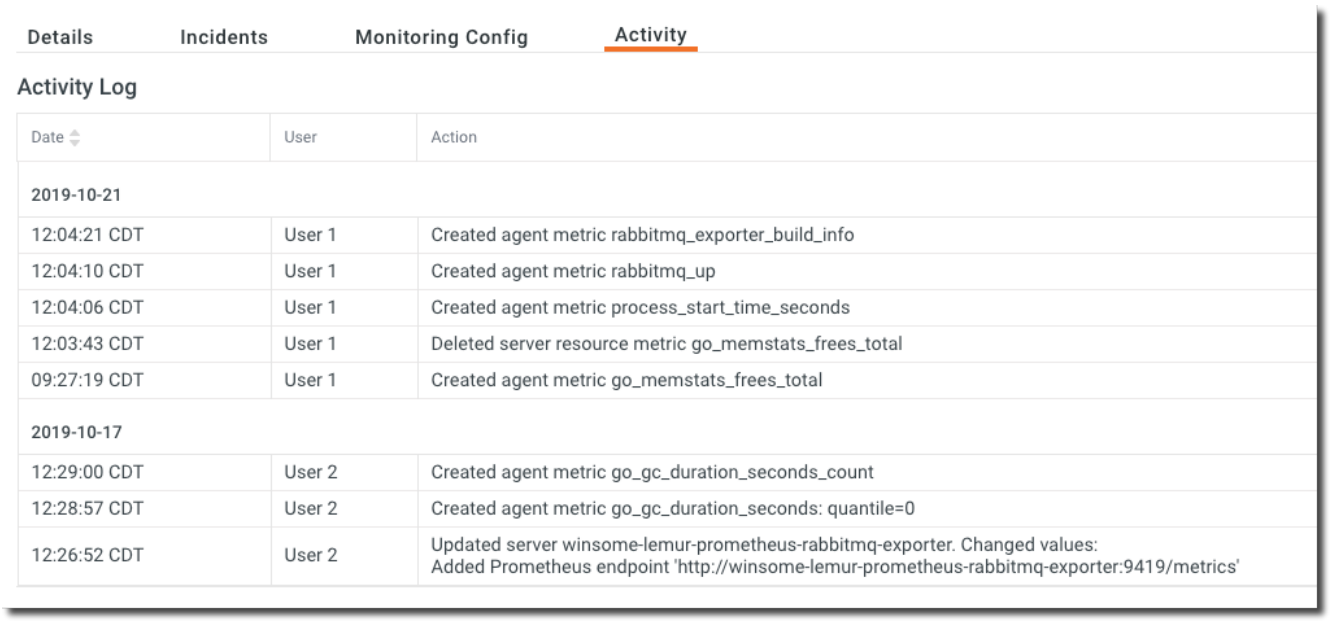
The **Monitoring Config** tab of the **Instance Details page** will show all of the metrics that you have configured for this instance.



The **Performance** section provides a visual representation of the status of the instance. Each metric is represented by a line graph that can be filtered based on metric type and scoped using a specific time range.



The **Activity** tab provides a log of all actions performed on the instance. The initiator of the action, the date, and the action performed is detailed in the **Activity Log** table.



RabbitMQ

The RabbitMQ plugin allows you to monitor several metrics on your RabbitMQ server. If you would like to install RabbitMQ, go to https://www.rabbitmq.com/download.html for instructions.

If you cannot find this plugin in the **/usr/lib/fm-agent/plugins** folder, then your version of the agent is out of date and you should update it to gain access to this feature.

Agent Configuration

Once the agent is installed, add the following information to the agent configuration file:

/etc/fm-agent/fm_agent.cfg

```
[rabbitmq]
management_interface_host = https://localhost
management_interface_port = 15672
username = rabbitmq_username
password = rabbitmq_password
```

Control Panel Configuration

To add a RabbitMQ Agent resource to your server first see Add FortiMonitor Agent checks if you have not already.

To use the plugin, select **RabbitMQ** from the monitoring catalog. The available metrics are listed below.

Metric	Unit
# of Objects	count

Cluster metrics	
Cluster message delivery rate	msgs/sec
Cluster message redelivery rate	msgs/sec
Cluster messages delivered to consumers recently	count
Cluster messages publish rate	msgs/sec
Cluster messages published recently	count
Cluster messages redelivered to consumers recently	count
Cluster number of messages ready for delivery	count
Cluster number of unacknowledged messages	count
Node metrics	
Node disk free amount	kBs
Node disk free amount rate	kBs/sec
Node file descriptors used	count
Node file descriptors used rate	count/sec
Node memory used rate	kBs/sec
Node number of Erlang processes in use	count
Node processes used rate	count/sec
Node rate of read count	count/sec
Node rate of read kBs	kBs/sec
Node rate of written bytes	kBs/sec
Node sockets available	count
Node sockets used	count
Node sockets used rate	count/sec
Node total amount of memory used	kBs
Node total number of bytes read from disk by the persister	kBs
Node total number of bytes written to disk by the persister	kBs
Node total number of read operations by the persister	count
Queue metrics	
Queue Length	messages
Queue Memory Usage	kb
Queue message delivery rate	count/sec
Queue message publishing rate	count/sec

Queue message redelivered recently	count
Queue messages delivered recently	count
Queue messages published recently	count
Queue messages redeliver rate	count/sec
Queue number of messages ready for delivery	count
Queue number of unacknowledged messages	count

If you would like to inspect the RabbitMQ plugin in more detail, you can find the python script for it in the **/usr/lib/fm-agent/plugins** folder.

Redis

The Redis plugin allows you to monitor a number of metrics on your key-value data store. If used effectively, this plugin can help you prevent outages before they happen. For example, you can use this plugin to alert your IT team when Redis' memory usage becomes dangerously high.

If you cannot find this plugin in the **/usr/lib/fm-agent/plugins** folder, then your version of the agent is out of date and you should update it to gain access to this feature.

If you haven't yet read the article Add FortiMonitor Agent checks, you should start there.

Agent Configuration

If your Redis instance is secured with password protection, you will need to add the password to the agent config file:

- **/etc/fm-agent/fm_agent.cfg**

If your Redis instance does not require a password, this section in the config file is not required.

```
[redis]
password = xxxxx
```

Control Panel Configuration

To use the Redis plugin, first select **Redis** from the monitoring catalog.

There are a number of metrics to monitor available.

- Clients: blocked clients
- Clients: connected clients
- Data: length of list
- Memory: used memory
- Memory: used memory peak
- Memory: used memory rss
- Persistence: background save in progress
- Persistence: changes since last save

- Replication: connected slaves
- Replication: role (master=1, slave=0)
- Server: uptime in seconds
- Stats: evicted keys
- Stats: expired keys
- Stats: keyspace hits
- Stats: keyspace misses
- Stats: pub/sub channels
- Stats: pub/sub patterns
- Stats: total commands processed

If you would like to inspect the Redis plugin in more detail, you can find the python script for it in the **/usr/lib/fm-agent/plugins**folder.

Sendmail

The **Sendmail** plugin allows you to monitor the queue depth of your Sendmail MTA server.

Control Panel Configuration

Select the **Sendmail** plugin from the monitoring catalog. To add this resource, please see Add FortiMonitor Agent checks.

Metric	Unit
Sendmail queue depth	messages

Additional permissions

Additional file system permissions may be needed if you are encountering problems when collecting data from Sendmail metrics. Resolve this issue by granting permission to the **fm-agent** user to run **Sendmail -bp**.

1. Run the following command as **root** to edit **/etc/sudoers**.

```
visudo /etc/sudoers
```

2. Add the following line:

```
fm-agent ALL=NOPASSWD:/usr/sbin/sendmail
```

3. Save and exit the file.

4. Running the **Sendmail** command may also require **sudo** privileges. To configure the command to use **sudo**, open **/etc/fm-agent/fm_agent.cfg** then add the following:

```
[sendmail]
use_sudo = 1
```

TCP

From the monitoring catalog, you can configure TCP port latency and port status as network checks between your Agent and the server you'd like to monitor.

To add this resource, please see Add FortiMonitor Agent checks.

Metric	Unit
TCP Port Latency	ms
TCP Port Status	Boolean

If you have any further questions, please contact our support team.

Tomcat (JMX)

The Tomcat plugin lets you monitor a number of metrics on your web server. If you cannot find this plugin in the **/usr/lib/fm-agent/plugins** folder, update your Agent.

Enable JMX on Tomcat

Before configuring the Agent, make sure that JMX is enabled on the host where your Tomcat web server is deployed.

1. Create **setenv.sh** file in the Tomcat bin directory. For example, if your bin directory is **/opt/tomcat/apache-tomcat-9.0.37/bin/**, run the following command:

```
export CATALINA_OPTS="-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=8008 -
Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=true -
Dcom.sun.management.jmxremote.access.file=/opt/tomcat/apache-tomcat-9.0.37/conf/jmxremote.access
-Dcom.sun.management.jmxremote.password.file=/opt/tomcat/apache-tomcat-
9.0.37/conf/jmxremote.password"
```

The command also allows you to set the port that you want to use for JMX and the path where the **jmx.access** and **jmx.password** files must be created.

Be sure that the Tomcat user owns the new **setenv.sh** file.

2. Create the **jmx.access** and **jmx.password** files.
jmx.access
`<USERNAME> readonly`

jmx.password
`<USERNAME> <PASSWORD>`

You can set the username and password as desired. These will be set in the FortiMonitor configuration file in the following section.

Also, make sure that the tomcat user owns these files and that the password file has permission 600.

```
-rw-r--r--. 1 tomcat tomcat 18 Jul 31 12:52 jmxremote.access
```

```
-rw-------. 1 tomcat tomcat 16 Jul 31 12:52 jmxremote.password
```

3. Restart the Tomcat service.

a. Navigate to ***tomcat_home*/bin**.

b. Run the following command to stop Apache Tomcat:

```
./shutdown.sh
```

c. Run the following command to restart Apache Tomcat:

```
./startup.sh
```

Agent configuration

Once the agent is installed, you will need to edit the Agent config file (**/etc/fm-agent/fm_agent.cfg**) and set the following parameters:

- **username** and **password** – The username and password that you used in the **jmx.password** file.
- **port** – The port where Tomcat is listening to. You can enter any number of ports in the **port** parameter for each Tomcat instance that you have installed in the server. Each port entered should be separated by a comma.
- **host** – The host where your Tomcat instances are deployed. If you have multiple Tomcat instances, make sure that you enter the hostname, separated by a comma, for each Tomcat instance you want to monitor. See the following example for more details.
- **jvm_path** – The path to your Java installation. Run `locate libjvm.so` to determine your JVM path.

The following Agent configuration file shows the configuration for multiple Tomcats running on a single server.

```
[tomcat_jmx]
username = <USERNAME>
password = <PASSWORD>
port = 8008, 9009
host = localhost, localhost
jvm_path = </PATH/TO/JVM>
```

Control Panel configuration

If you have not already, read the Add FortiMonitor Agent checks article to learn how to set up monitoring for any agent plugin. To use the Tomcat plugin, first select **Tomcat (JMX)** from the monitoring catalog.

There are a number of metrics available to monitor.

Metric	Unit
Action/Classloadingtime	
Action/Errorcount	

Action/Loadtime	
Action/Maxtime	
Action/Processingtime	
Action/Requestcount	
Average total memory	seconds
Count of errors	errors
Count of expired sessions	sessions
Count of requests	requests
Current count of busy threads	threads
Current count of threads	threads
Default/Classloadingtime	
Default/Errorcount	
Default/Loadtime	
Default/Maxtime	
Default/Processingtime	
Default/Requestcount	
DynamicJAXWSServlet/Classloadingtime	
DynamicJAXWSServlet/Errorcount	
DynamicJAXWSServlet/Loadtime	
DynamicJAXWSServlet/Maxtime	
DynamicJAXWSServlet/Processingtime	
DynamicJAXWSServlet/Requestcount	
GC/Classloadingtime	
GC/Errorcount	
GC/Loadtime	
GC/Maxtime	
GC/Processingtime	
GC/Requestcount	
HTMLManager/Classloadingtime	
HTMLManager/Errorcount	
HTMLManager/Loadtime	
HTMLManager/Maxtime	

HTMLManager/Processingtime	
HTMLManager/Requestcount	
J4p-agent/Classloadingtime	
J4p-agent/Errorcount	
J4p-agent/Loadtime	
J4p-agent/Maxtime	
J4p-agent/Processingtime	
J4p-agent/Requestcount	
JMXProxy/Classloadingtime	
JMXProxy/Errorcount	
JMXProxy/Loadtime	
MXProxy/Maxtime	
JMXProxy/Processingtime	
JMXProxy/Requestcount	
Jsp/Classloadingtime	
Jsp/Errorcount	
Jsp/Loadtime	
Jsp/Maxtime	
Jsp/Processingtime	
Jsp/Requestcount	
JVM maximum memory	megabytes
JVM total memory	megabytes
Longest sessions alive time	seconds
Manager/Classloadingtime	
Manager/Errorcount	
Manager/Loadtime	
Manager/Maxtime	
Manager/Processingtime	
Manager/Requestcount	
Maximum number of active sessions	sessions
Maximum number of threads	threads
Maximum processing time	milliseconds

Mem/Classloadingtime	
Mem/Errorcount	
Mem/Loadtime	
Mem/Maxtime	
Mem/Processingtime	
Mem/Requestcount	
Memory used	MB
Number of loaded JSPs	JSPs
Number of reloaded JSPs	JSPs
Params/Classloadingtime	
Params/Errorcount	
Params/Loadtime	
Params/Maxtime	
Params/Processingtime	
Params/Requestcount	
Rejected sessions creations	sessions
RSDispatcher/Classloadingtime	
RSDispatcher/Errorcount	
RSDispatcher/Loadtime	
RSDispatcher/Maxtime	
RSDispatcher/Processingtime	
RSDispatcher/Requestcount	
ShutdownRestart/Classloadingtime	
ShutdownRestart/Errorcount	
ShutdownRestart/Loadtime	
ShutdownRestart/Maxtime	
ShutdownRestart/Processingtime	
ShutdownRestart/Requestcount	
Status/Classloadingtime	
Status/Errorcount	
Status/Loadtime	
Status/Maxtime	

Status/Processingtime	
Status/Requestcount	
Time to start up	milliseconds
Total active sessions	sessions
Total bytes received	bytes
Total bytes sent	bytes
Total number of active sessions since startup	
Total number of rejected sessions since startup	
Total number of requests served since startup	
Total processing time	seconds
Total processing time (in milliseconds) since startup	
Total sessions count	sessions
Total/Averagesessionalivetime	
Total/Bytesreceived	
Total/Bytessent	
Total/Currentthreadbusy	
Total/Currentthreadcount	
Total/Errorcount	
Total/Expiredsessions	
Total/Freememory	
Total/JSPsloaded	
Total/JSPsreloaded	
Total/Longestsessionalivetime	
Total/Maxactivesessions	
Total/Maxmemory	
Total/Maxprocessingtime	
Total/Maxthreads	
Total/Requestcount	
Total/Totalmemory	
WSServlet/Classloadingtime	
WSServlet/Errorcount	
WSServlet/Loadtime	

WSServlet/Maxtime
WSServlet/Processingtime
WSServlet/Requestcount

If you would like to inspect the Tomcat plugin in more detail you can find the python script for it in the **/usr/lib/fm-agent/plugins** folder.

Tomcat (Legacy)

The Tomcat plugin lets you monitor a number of metrics on your web server. If you can not find this plugin in the **/usr/lib/fm-agent/plugins** folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

Agent Configuration

Once the agent is installed, you will need to add the following information to the agent config file (**/etc/fm-agent/fm_agent.cfg**):

```
[tomcat]
username = <USERNAME>
password = <PASSWORD>
console_url = http://<IP>:<PORT>
```

The console_url is the URL that you use to access your Tomcat web server. Replace <IP> with your server's IP address, and <PORT> with the port tomcat is listening to (this will be 8080 by default).

If you do not know the username and password required to sign in to your Tomcat URL, then you can find them in the conf/tomcat-users.xml file. The user you choose must be of the manager-gui role.

Control Panel Configuration

To use the Tomcat plugin, first select "Tomcat" from the monitoring catalog.

There are a number of metrics available to monitor.

Metric	Unit
Action/Classloadingtime	
Action/Errorcount	
Action/Loadtime	
Action/Maxtime	
Action/Processingtime	
Action/Requestcount	

Average total memory	seconds
Count of errors	errors
Count of expired sessions	sessions
Count of requests	requests
Current count of busy threads	threads
Current count of threads	threads
Default/Classloadingtime	
Default/Errorcount	
Default/Loadtime	
Default/Maxtime	
Default/Processingtime	
Default/Requestcount	
DynamicJAXWSServlet/Classloadingtime	
DynamicJAXWSServlet/Errorcount	
DynamicJAXWSServlet/Loadtime	
DynamicJAXWSServlet/Maxtime	
DynamicJAXWSServlet/Processingtime	
DynamicJAXWSServlet/Requestcount	
GC/Classloadingtime	
GC/Errorcount	
GC/Loadtime	
GC/Maxtime	
GC/Processingtime	
GC/Requestcount	
HTMLManager/Classloadingtime	
HTMLManager/Errorcount	
HTMLManager/Loadtime	
HTMLManager/Maxtime	
HTMLManager/Processingtime	
HTMLManager/Requestcount	
J4p-agent/Classloadingtime	
J4p-agent/Errorcount	

J4p-agent/Loadtime	
J4p-agent/Maxtime	
J4p-agent/Processingtime	
J4p-agent/Requestcount	
JMXProxy/Classloadingtime	
JMXProxy/Errorcount	
JMXProxy/Loadtime	
MXProxy/Maxtime	
JMXProxy/Processingtime	
JMXProxy/Requestcount	
Jsp/Classloadingtime	
Jsp/Errorcount	
Jsp/Loadtime	
Jsp/Maxtime	
Jsp/Processingtime	
Jsp/Requestcount	
JVM maximum memory	megabytes
JVM total memory	megabytes
Longest sessions alive time	seconds
Manager/Classloadingtime	
Manager/Errorcount	
Manager/Loadtime	
Manager/Maxtime	
Manager/Processingtime	
Manager/Requestcount	
Maximum number of active sessions	sessions
Maximum number of threads	threads
Maximum processing time	milliseconds
Mem/Classloadingtime	
Mem/Errorcount	
Mem/Loadtime	
Mem/Maxtime	

Mem/Processingtime	
Mem/Requestcount	
Memory used	MB
Number of loaded JSPs	JSPs
Number of reloaded JSPs	JSPs
Params/Classloadingtime	
Params/Errorcount	
Params/Loadtime	
Params/Maxtime	
Params/Processingtime	
Params/Requestcount	
Rejected sessions creations	sessions
RSDispatcher/Classloadingtime	
RSDispatcher/Errorcount	
RSDispatcher/Loadtime	
RSDispatcher/Maxtime	
RSDispatcher/Processingtime	
RSDispatcher/Requestcount	
ShutdownRestart/Classloadingtime	
ShutdownRestart/Errorcount	
ShutdownRestart/Loadtime	
ShutdownRestart/Maxtime	
ShutdownRestart/Processingtime	
ShutdownRestart/Requestcount	
Status/Classloadingtime	
Status/Errorcount	
Status/Loadtime	
Status/Maxtime	
Status/Processingtime	
Status/Requestcount	
Time to start up	milliseconds
Total active sessions	sessions

Total bytes received	bytes
Total bytes sent	bytes
Total number of active sessions since startup	
Total number of rejected sessions since startup	
Total number of requests served since startup	
Total processing time	seconds
Total processing time (in milliseconds) since startup	
Total sessions count	sessions
Total/Averagesessionalivetime	
Total/Bytesreceived	
Total/Bytessent	
Total/Currentthreadbusy	
Total/Currentthreadcount	
Total/Errorcount	
Total/Expiredsessions	
Total/Freememory	
Total/JSPsloaded	
Total/JSPsreloaded	
Total/Longestsessionalivetime	
Total/Maxactivesessions	
Total/Maxmemory	
Total/Maxprocessingtime	
Total/Maxthreads	
Total/Requestcount	
Total/Totalmemory	
WSServlet/Classloadingtime	
WSServlet/Errorcount	
WSServlet/Loadtime	
WSServlet/Maxtime	
WSServlet/Processingtime	
WSServlet/Requestcount	

If you would like to inspect the Tomcat plugin in more detail you can find the python script for it in the **/usr/lib/fm-agent/plugins** folder.

Tomcat (Windows)

The Tomcat plugin lets you monitor a number of metrics on your Windows Tomcat server.

Agent Configuration

Once the agent is installed, you will need to add the following information to the Agent configuration file (**C:\Program Files\FortiMonitorAgent\Agent.config**):

```
<tomcat>
<add key="username" value="admin" />
<add key="password" value="admin" />
<add key="console_url" value="http://127.0.0.1:9090/manager/status/all" />
</tomcat>
```

The console_url is the URL that you use to access your Tomcat web server. Replace <IP> with your server's IP address, and <PORT> with the port tomcat is listening to (this will be 8080 by default).

If you do not know the username and password required to sign into your Tomcat URL, then you can find them in the conf/tomcat-users.xml file. The user you choose must be of the manager-gui role.

Control Panel Configuration

If you have not already, read the Add FortiMonitor Agent checks article to learn how to set up monitoring for any agent plugin. To use the Tomcat plugin, first select "Tomcat" from the monitoring catalog.

There are a number of metrics available to monitor.

Metric	Unit
Action/Classloadingtime	
Action/Errorcount	
Action/Loadtime	
Action/Maxtime	
Action/Processingtime	
Action/Requestcount	
Average total memory	seconds
Count of errors	errors
Count of expired sessions	sessions
Count of requests	requests

Current count of busy threads	threads
Current count of threads	threads
Default/Classloadingtime	
Default/Errorcount	
Default/Loadtime	
Default/Maxtime	
Default/Processingtime	
Default/Requestcount	
DynamicJAXWSServlet/Classloadingtime	
DynamicJAXWSServlet/Errorcount	
DynamicJAXWSServlet/Loadtime	
DynamicJAXWSServlet/Maxtime	
DynamicJAXWSServlet/Processingtime	
DynamicJAXWSServlet/Requestcount	
GC/Classloadingtime	
GC/Errorcount	
GC/Loadtime	
GC/Maxtime	
GC/Processingtime	
GC/Requestcount	
HTMLManager/Classloadingtime	
HTMLManager/Errorcount	
HTMLManager/Loadtime	
HTMLManager/Maxtime	
HTMLManager/Processingtime	
HTMLManager/Requestcount	
J4p-agent/Classloadingtime	
J4p-agent/Errorcount	
J4p-agent/Loadtime	
J4p-agent/Maxtime	
J4p-agent/Processingtime	

J4p-agent/Requestcount	
JMXProxy/Classloadingtime	
JMXProxy/Errorcount	
JMXProxy/Loadtime	
JMXProxy/Maxtime	
JMXProxy/Processingtime	
JMXProxy/Requestcount	
Jsp/Classloadingtime	
Jsp/Errorcount	
Jsp/Loadtime	
Jsp/Maxtime	
Jsp/Processingtime	
Jsp/Requestcount	
JVM maximum memory	megabytes
JVM total memory	megabytes
Longest sessions alive time	seconds
Manager/Classloadingtime	
Manager/Errorcount	
Manager/Loadtime	
Manager/Maxtime	
Manager/Processingtime	
Manager/Requestcount	
Maximum number of active sessions	sessions
Maximum number of threads	threads
Maximum processing time	milliseconds
Mem/Classloadingtime	
Mem/Errorcount	
Mem/Loadtime	
Mem/Maxtime	
Mem/Processingtime	
Mem/Requestcount	

Memory used	MB
Number of loaded JSPs	JSPs
Number of reloaded JSPs	JSPs
Params/Classloadingtime	
Params/Errorcount	
Params/Loadtime	
Params/Maxtime	
Params/Processingtime	
Params/Requestcount	
Rejected sessions creations	sessions
RSDispatcher/Classloadingtime	
RSDispatcher/Errorcount	
RSDispatcher/Loadtime	
RSDispatcher/Maxtime	
RSDispatcher/Processingtime	
RSDispatcher/Requestcount	
ShutdownRestart/Classloadingtime	
ShutdownRestart/Errorcount	
ShutdownRestart/Loadtime	
ShutdownRestart/Maxtime	
ShutdownRestart/Processingtime	
ShutdownRestart/Requestcount	
Status/Classloadingtime	
Status/Errorcount	
Status/Loadtime	
Status/Maxtime	
Status/Processingtime	
Status/Requestcount	
Time to start up	milliseconds
Total active sessions	sessions
Total bytes received	bytes

Total bytes sent	bytes
Total number of active sessions since startup	
Total number of rejected sessions since startup	
Total number of requests served since startup	
Total processing time	seconds
Total processing time (in milliseconds) since startup	
Total sessions count	sessions
Total/Averagesessionalivetime	
Total/Bytesreceived	
Total/Bytessent	
Total/Currentthreadbusy	
Total/Currentthreadcount	
Total/Errorcount	
Total/Expiredsessions	
Total/Freememory	
Total/JSPsloaded	
Total/JSPsreloaded	
Total/Longestsessionalivetime	
Total/Maxactivesessions	
Total/Maxmemory	
Total/Maxprocessingtime	
Total/Maxthreads	
Total/Requestcount	
Total/Totalmemory	
WSServlet/Classloadingtime	
WSServlet/Errorcount	
WSServlet/Loadtime	
WSServlet/Maxtime	
WSServlet/Processingtime	
WSServlet/Requestcount	

UDP

Control Panel Configuration

From the monitoring catalog, you can configure UDP port status and UDP connections as network checks between your Agent and the server you'd like to monitor.

To add this resource, please see Add FortiMonitor Agent checks.

Metric	Unit
UDP Connections	Count
UDPv6 Connections	Count
UDP Port Status	Boolean

If you have any further questions, please contact our support team.

UWSGI

You can use the UWSGI plugin to report on the health of your Python application server. This includes information surrounding worker status/throughput, exception counts and overall performance. If you can not find this plugin in the /usr/lib/fm-agent/plugins folder, then your version of the Agent is out of date and you should update it to gain access to this feature.

Agent Configuration

In order to use this plugin, you will need to add the host server address and port of your UWSGI stats page. This can be specified when starting up an application with uwsgi using the extra option "–stats 127.0.0.1:1313 –stats-http". Replacing the IP address and port with whatever you wished to bind the stats page to. Once you have done this, add the address and port to the **/etc/fm-agent/fm_agent.cfg** file as shown below:

```
[uwsgi]
server = 127.0.0.1
port = 1313
```

Control Panel Configuration

To add a UWSGI agent resource to your server, read Add FortiMonitor Agent checks. To use the UWSGI plugin, first select UWSGI from the **App Server** option in the **Add Monitoring** dropdown.

The Options Allow you to select the UWSGI worker you wish to monitor. The Metric drop-down will have nine options:

Metric	Units
Average request time	ms
Exceptions	exceptions
How long worker is working	workers

How many data was transmitted by worker	data
How many requests worker did since worker (re)spawn	requests
Requests per second	requests
Worker is busy or free to use(BUSY=1, FREE=0)	
Worker RSS (Resident Set Size)	
Worker VSZ (Virtual Memory Size)	

If you would like to inspect the UWSGI plugin in more detail you can find the python script for it in the **/usr/lib/fm-agent/plugins** folder.

If you have followed the steps above and the new metric and plugin do not show properly, you may need to manually rebuild the agent's metadata. You can do this by following the guide here.

Varnish

The Varnish plugin does not require any special configuration. You need to have the latest version of the Agent to use this plugin.

If you haven't yet read Add FortiMonitor Agent checks, start there.

Control Panel Configuration

First select **Varnish** from the monitoring catalog.

There are a number of available metrics to monitor.

If you would like to inspect the process name plugin in more detail you can find the python script for it in /usr/lib/fm-agent/plugins/varnish.py.

Metric	Unit
Backend conn. features	failures
Cache hits	hits
Cache hits for pass	hits
Cache misses	misses
Client connections accepted	requests
Client requests received	requests
ESI parse errors (unlock)	errors
Ndropped work requests	requests
Nexpired objects	objects
NLRU nuked objects	objects
Nstruct object	objects

Nworker threads	threads
Nworker threads created	threads
Nworker threads limited	threads
Nworker threads not created	threads

Windows Log Monitor

The Windows Log Monitor plugin allows you to monitor your windows logs based on the matching and frequency of certain specified occurrences. Event Log monitoring is based on C# regex syntax, you can read more about the syntax required to monitor event logs.

If you can not add this plugin in the Windows **Add** metric list, then your version of the Agent is out of date and you should update it to gain access to this feature.

If you haven't read Add FortiMonitor Agent checks, start there.

Control Panel Configuration

You can add a Windows Log Monitor check by clicking **Add** under the Windows metric section. Look for **Windows Event Logs**.

Metric	Unit
Windows Event Logs	Occurrences

The check itself will hinge off a number of variables, which are listed with their options below:

Log type

- System
- Application
- Custom
- All

To add custom logs or logs that are neither System or Application, see Custom logs.

Log Severity

- Critical
- Warning
- Error
- Information

Log Source (optional)

- All Sources
- Application Popup
- Customer Experience Improvement
- EventLog
- Microsoft-Windows-CAPI2

- Microsoft-Windows-DHCPv6-Client
- Microsoft-Windows-DNS-Client
- Microsoft-Windows-Dchp-Client
- Microsoft-Windows-Diagnostics-Networking
- Microsoft-Windows-Eventlog
- Microsoft-Windows-FilterManager
- Microsoft-Windows-GroupPolicy
- Microsoft-Windows-Iphlpsvc
- Microsoft-Windows-Kernel-General
- Microsoft-Windows-Kernel-Power
- Microsoft-Windows-Kernel-Processor-Power
- Microsoft-Windows-RestartManager
- Microsoft-Windows-Security-Auditing
- Microsoft-Windows-Time-Service
- Microsoft-Windows-UserPnp
- Microsoft-Windows-WindowsUpdateClient
- Microsoft-Windows-Winlogon
- MsiInstaller
- NetBT
- PowerShell
- Server
- Service Control Manager
- TermDD
- USER32
- WinRM
- Windows Error Reporting
- xennet6

Log Message Regex Filter (Optional)

Set the **Time Frame** (range from 1 minute-60 minutes in 10-minute increments)

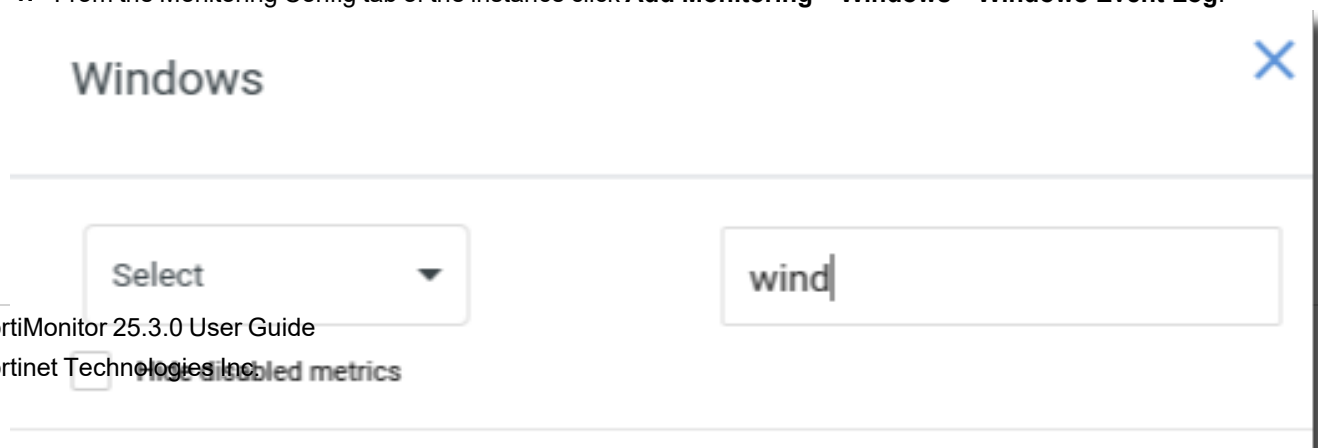
Set the **frequency** of the monitor (how often will the metric be collected?)

You can also set the Alert Timeline the alerts will adhere to, and tag the Windows Log Monitor checks.

Custom logs

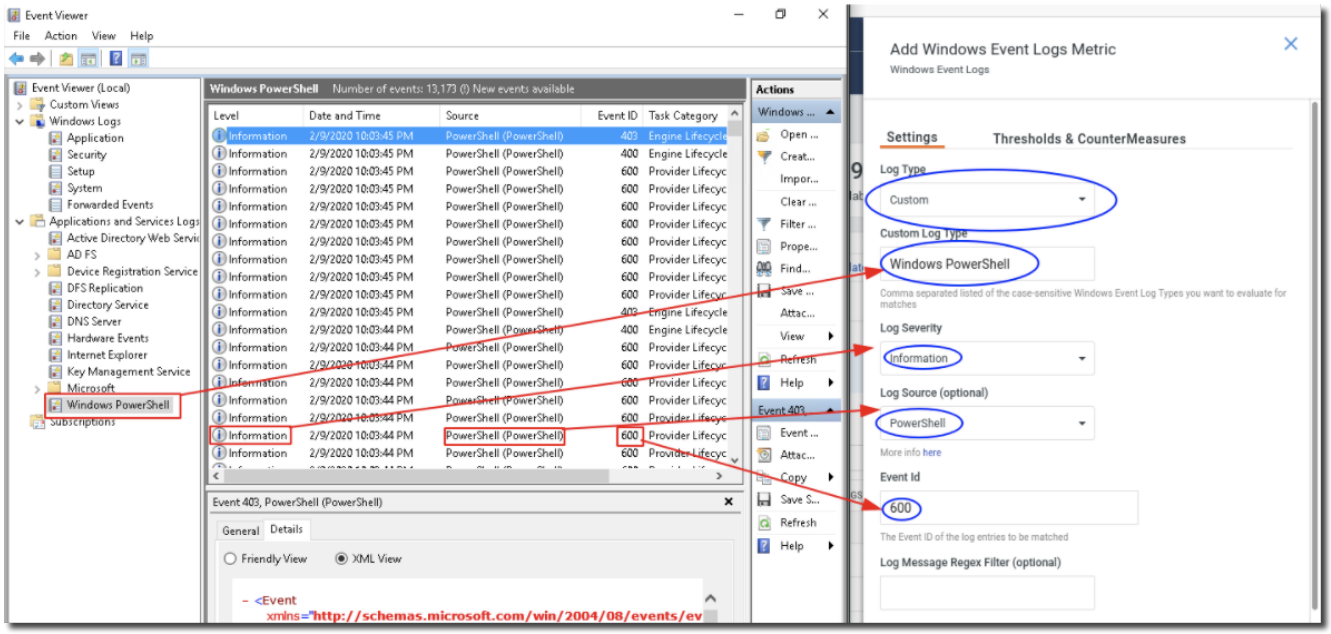
To add custom logs, perform the following steps:

1. From the Monitoring Config tab of the instance click **Add Monitoring** > **Windows** > **Windows Event Log**.



2. For the **Log Type**, select **Custom**.

3. Fill in the other fields as shown in the following example. In this example, the Windows Event Viewer is used as a reference.



4. Set the Time Frame and Frequency based on your needs and the average period between events.

If you have any questions about how to properly configure this check, please contact our support team.

Windows Service

The Windows Service agent plugin type allows you to monitor several different metrics on a specified windows service running on your server.

If you haven't yet read Add FortiMonitor Agent checks, please start there.

Control Panel Configuration

First select “Windows” from the monitoring catalog. Scroll to the Service section.

There are 3 options for this metric.

Metric	Unit
% Processor Time	%
Memory in Use	kb
Running status	boolean

The specific number that this resource returns is defined by the above options, please take this into account if you choose to change your “Alert Threshold” for this resource.

You can monitor multiple Windows Service agent resources on one server if you would like, especially if you have a number of essential services running on your server.

SharePoint

The Sharepoint plugin is used to monitor your Sharepoint application. If you haven't yet read the Add FortiMonitor Agent checks, please start there.

Control Panel Configuration

To begin monitoring, choose the Sharepoint plugin from the monitoring catalog.

There are a number of metrics to choose from.

Metric	Unit
Foundation	
Active Threads	Threads
Current Page Requests	Requests
Executing Time/Page Request	ms/request
Incoming Page Request Rate	
Reject Page Request Rate	
Responded Page Request Rate	
Publishing Cache	
Publishing cache flushes/sec	flushes/sec
Publishing cache hits/sec	hits/sec
Publishing Cache Hit Ratio	misses/sec
Publishing Cache Misses Ratio	misses/sec
Total number of cache compactions	compactions

Uptime

The Uptime plugin allows you to monitor when your system was last rebooted.

If you haven't yet read our Add FortiMonitor Agent checks article, start there.

Control Panel Configuration

The metric is part of the Linux plugin, under the Other category.

Metric	Unit
Time Since Last Boot	seconds

You can set a threshold to alert you if too much time has passed since the last boot.

If you have any questions, please contact our support team.

Integrate with Domino

Integrating Domino with FortiMonitor

Make sure that you have the appropriate permissions on the directory used to write the domino statistics. The FortiMonitor agent needs to be able to read and remove files from the directory you designated.

Install the Agent

To integrate Domino with FortiMonitor, you must first install the FortiMonitor Agent. The steps are detailed in Install the FortiMonitor Agent.

If you are installing the Linux Agent, make sure to append --custom-metrics at the end of the standalone installer command.

Create the directories where Domino will post the output

Domino will output statistics files that the FortiMonitor Agent will pick up. These are stored in a sub-folder under the Domino installation directory. You need to create these folders before modifying the Domino **Notes.ini** file.

For example, from the directory **C:\Program Files\IBM\Domino\data** you will need to create: **stats\FortiMonitor\custom_metrics**.

Modify Notes.ini

After installing the Agent and creating the folder where Domino will write output statistics, you can now edit your Domino server's **notes.ini** file and add the following content.

For Linux

Add the following to the **Notes.ini** file.

```
STATPUB_ENABLE=1

STATPUB_URI=file:///../../usr/share/fm-agent/incoming/dominostats_$Timestamp$

STATPUB_HEADERS=Content-type: application/json

STATPUB_DATA_HEAD={"agent_payload_version": 1.0, "plugin_textkey": "com.pnp-
hcl.dominostats", "plugin_category_name": "Domino Statistics", "metrics": [$Newline$

STATPUB_METRIC_FORMAT={"textkey": "$Name$", "label": "$Name$", "value": $Value$, "unit": "
(units)", "timestamp": "$Timestamp$"}

STATPUB_DELTA_METRIC_FORMAT={"textkey": "Rate.$Name$", "label": "$Name$ Rate", "value":
$Value$, "unit": "(units per minute)", "timestamp": "$Timestamp$"}

STATPUB_METRIC_DELIMITER=,$Newline$

STATPUB_DATA_TAIL=$Newline$]}
```

For Windows

1. Add the following to the **Notes.ini** file.
   Note: The STATPUB_URI needs to be a relative path in relation to your Domino install. The installation below

assumes Domino is installed in C:\.

```
STATPUB_ENABLE=1

STATPUB_URI=file:///stats\FortiMonitor\custom_metrics\domino_stats_$Timestamp$

STATPUB_HEADERS=Content-type:application/json

STATPUB_DATA_HEAD={"agent_payload_version": 1.0, "plugin_textkey": "com.pnp-
hcl.dominostats", "plugin_category_name": "Domino Statistics", "metrics": [$Newline$

STATPUB_METRIC_FORMAT={"textkey": "$Name$", "label": "$Name$", "value": $Value$, "unit": "
(units)", "timestamp": "$Timestamp$"}

STATPUB_DELTA_METRIC_FORMAT={"textkey": "Rate.$Name$", "label": "$Name$ Rate", "value":
$Value$, "unit": "(units per minute)", "timestamp": "$Timestamp$"}

STATPUB_METRIC_DELIMITER=,$Newline$

STATPUB_DATA_TAIL=$Newline$]}
```

2. Edit the FortiMonitor agent config file. You can find it in **C:\Program Files\FortiMonitorAgent**.

3. Add the Domino output path to the **<service>** field:

```
<add key="MetricIncomingDirectory" value="C:\Program
Files\IBM\Domino\data\stats\FortiMonitor\custom_metrics" />
```

For example:

```
<agent>
       <service>
              <add key="AggregatorUrl" value="https://rx.us01.fortimonitor.com/v20.11" />
              <add key="ServerKey" value="xxxx-yyyy-zzzz-0000" />
              <add key="MaintenanceDuration" value="" />
              <add key="ShowPublicStatus" value="" />
              <add key="EnableCounterMeasures" value="true" />
              <add key="CounterMeasuresRemotePlugins" value="" />
              <add key="CounterMeasuresRefreshPlugins" value="" />
              <add key="MetricIncomingDirectory" value="C:\Program
Files\IBM\Domino\data\stats\FortiMonitor\custom_metrics" />
              <!--
                     <add key="DiscoveryCheckInterval" value="300" />
                            <add key="MetricResultsQueueMax" value="7000" />
                     <add key="EnableAlertCheckDelay" value="False" />
                     <add key="PluginBlacklist" value="" />
              -->
       </service>
```

4. Restart the FortiMonitor Windows Agent service after changing this setting.

Troubleshooting

Unable to see Domino metric data

If you do not see any Domino metric data in your control panel:

1. Check that you have modified the **notes.ini** file correctly. See Modify Notes.ini.
2. Restart the Domino service after saving the file.
3. Make sure the FortiMonitor Agent is able to read the files and write to the directory.
4. Ensure that the folders where Domino writes the output have been created and the path is correctly set. See Create the directories where Domino will post the output.

Failed to POST metric data error message

If you see the following error:

[001927:000182-00007F86E4D13700] 03-01-2020 08:00:16.46 StatPubFlushPostData> Failed to POST metric data using URI: 'file:///../../usr/share/fm-agent/incoming/1578034816'; Cannot write or create file (file or disk is read-only)

Make sure that you have the appropriate permissions on the directory used to write the domino statistics. The FortiMonitor agent needs to be able to read and remove files from the directory you designated.

If you see the following error:

001927:000182-00007F86E4D13700] 03-01-2020 07:58:16.45 StatPubFlushPostData> Failed to POST metric data using URI: 'file:///../../usr/share/fm-agent/incoming'; Is a directory

Make sure that the URI ends with a file.

Unable to create the STATPUB_URI file

If you are having trouble getting the STATPUB_URI file written:

1. Add **STATPUB_DEBUG=1** to your **notes.ini** for more diagnostic information.
2. Restart the Domino service after saving the file.
3. Check the Domino log files for errors.

Monitor Beanstalk with the FortiMonitor Agent

1. If you haven't already, in the root of your Beanstalk project, create a directory named `.ebextensions`.
2. Next, you'll need a to create a config file, such as `01_fm_agent.config`.
   Note: the 01 is used to help with execution order, so feel free to drop it if ordering isn't important. The only important part is the .config extension.
3. Inside your new config file, add the following snippet, replacing the customer key placeholder with your own.

```
container_commands:

install_panopta_agent:
curl -s https://packages.panopta.com/install/linux_agent_install.py | python3 /dev/stdin --
customer-key YOUR_CUSTOMER_KEY
```

Endpoint Agent

An Endpoint Agent (DEM Agent) is a local monitoring utility that is deployed directly on a Windows or MacOS instance. The Agent allows you to monitor your Windows machine's health, performance, and DEM (Digital Experience Monitoring) metrics to ensure that end-user experience is optimized. The available metrics are detailed in the following sections.

The Endpoint Agent can also be used to trigger CounterMeasures.

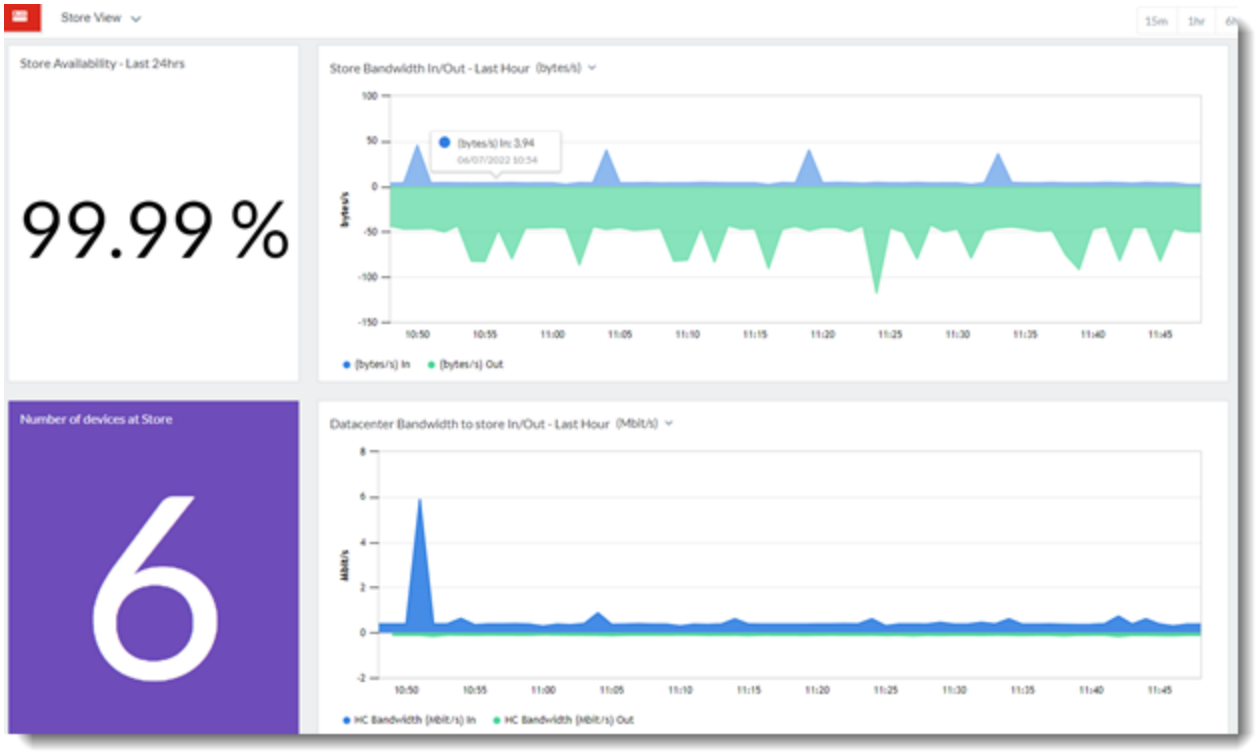
To add a Windows Endpoint Agent, see Add a Windows Endpoint FortiMonitor Agent.

To add a MacOS Endpoint Agent, see Add a MacOS Endpoint Agent.

Usage scenarios

Remote site availability and performance

You can use Endpoint DEM to make sure that a remote location, like a store or restaurant, has good network availability, and more importantly that the users at the store can get to the key applications that they rely on. You can start at a high level with a view of all the stores, then use FortiMonitor's custom dashboards to view the individual stores.



Collected server information

The Agent also supports collection of WiFi and wired network connection details. (signal strength, SSID, wired or wireless)

Available metrics

In addition to standard health metrics and Windows PerfMon metrics, the following metrics are also supported:

- Network quality checks
  - Jitter
  - Latency
  - MOS
  - Packet loss
- Synthetic checks
  - HTTP/HTTPS transactions (including SSL/TLS)
  - Traceroutes
- Download speed
- WiFi Rx (Reception) Rate
- WiFi Tx (Transmission) Rate
- WiFi Signal Strength
- Windows Endpoint Agent health metrics

Add a Windows Endpoint FortiMonitor Agent

This article describes the steps on how to enable Digital Experience Monitoring (DEM) using the Windows Endpoint Agent.

Resource requirements

These are the minimum resources required to add a Windows Endpoint FortiMonitor Agent.
For a maximum of 20 DEM checks running from any one Agent:

- Memory: 55 MB
- Hard disk space: 50 MB

Add a new Windows Endpoint FortiMonitor Agent

To install a Windows Endpoint Agent, perform the following steps:

1. From the navigation menu, click **Add**.
2. Select **Windows Endpoint FortiMonitor Agent**.
3. Select either **Quick Install Instructions** or **Manual Install Instructions**.
   a. Quick install instructions:
   Run the following command then follow the on-screen instructions:
   ```
   Invoke-WebRequest https://repo.fortimonitor.com/install/win/fortimonitor_agent_windows.ps1 -OutFile fortimonitor_agent_windows.ps1; .\fortimonitor_agent_windows.ps1 -customer_key <CUSTOMER_KEY> -countermeasures -dem
   ```

   Where:
   CUSTOMER_KEY - Your customer key. To obtain your customer key, visit the FortiMonitor Control Panel and click your avatar then select My Account.
   -countermeasures - Add this flag to enable CounterMeasures.
   -dem - Add this flag to enable DEM.
   b. Manual install instructions:
      i. Follow the on-screen instructions.
      ii. When creating the manifest file, make sure to add the **digital_experience = true** parameter. See the following example:

      ```
      [agent]
      customer_key = dwrm-bzbh-k8mj-dggw
      enable_countermeasures = true
      digital_experience = true
      ```

      The manifest file can placed anywhere. The on-screen instructions assume it is at **C:\FortiMonitorAgent.manifest**. For more information on how to use the manifest file, see Use the Agent manifest file.

Enable DEM on an existing Agent

To enable DEM on an existing Agent, do the following:

1. Edit the **Agent.cfg** file located in **C:\Program Files\FortiMonitor\Agent.cfg** if you are using the FortiMonitor Agent and add **<add key="DigitalExperience" value="true" />**. The relevant section of the configuration is shown below:
   ```
   <agent>
   <service>
   <add key="DigitalExperience" value="true" />
   ```
2. Save and close the file.

Note: If you are using the Windows version of the Agent, you will have to restart the service from within the services menu before seeing any configuration changes take place.

MacOS Endpoint Agent

The MacOS Endpoint Agent is a local monitoring utility that is deployed directly on a MacOS instance. The Agent allows you to monitor your MacOS machine's health, performance, and DEM (Digital Experience Monitoring) metrics to ensure that end-user experience is optimized.

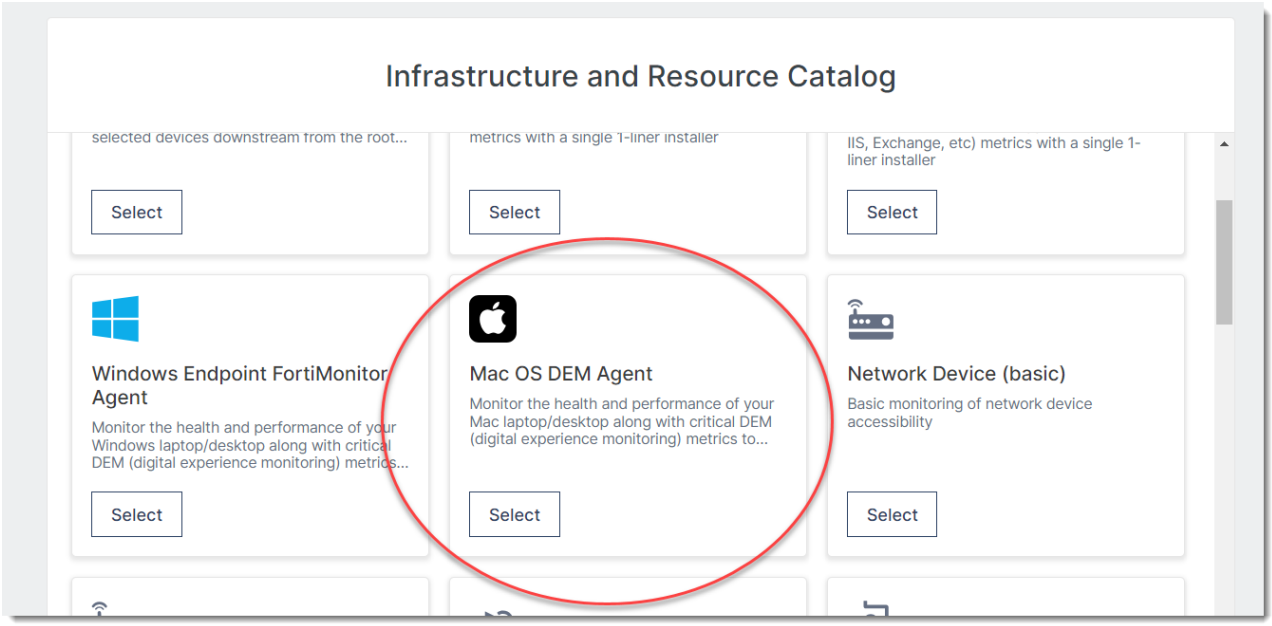
Supported MacOS versions

- Intel and Apple CPUs
- MacOS 11-14

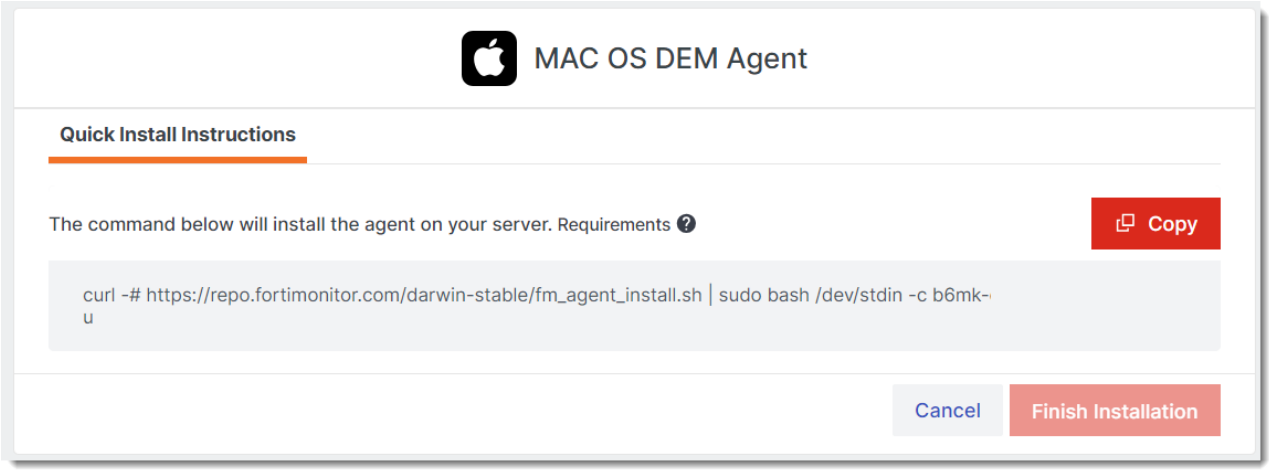
Add a MacOS Endpoint Agent

To install a MacOS Endpoint Agent, perform the following steps:

1. From the navigation menu, click **Add**.
2. Select **MacOS DEM Agent**.



3. Copy the **Quick Install Instructions** command.



4. Open a terminal window then run the following command. Make sure you have administrator privileges (sudo).
   `curl -# https://repo.fortimonitor.com/darwin-stable/fm_agent_install.sh | sudo bash /dev/stdin -c <CUSTOMER_KEY> -e -d -u`
   Where:
   <CUSTOMER_KEY> is your customer key.
   -e sets the digital experience section in the manifest file.
   -d sets the disable server match key in the manifest file
   -u sets the auto-update flag in the manifest file.
5. Wait for the installation to complete.
6. Go back to FortiMonitor then click **Finish Installation**.

CounterMeasures

CounterMeasures are automated responses to specific events that the macOS Agent detects on your infrastructure. Automated actions are provided but you can also create your own custom CounterMeasures. Supported CounterMeasures are largely the same with Linux CounterMeasures.

Created custom CounterMeasures files can be dropped in the following location:

`/usr/local/FortiMonitor/agent/share/fm-agent/countermeasures`

Uninstall the MacOS Endpoint Agent

To uninstall the agent, open a terminal window then run the following command:

`sudo bash /usr/local/fortimonitor/uninstall.sh`

CounterMeasures

CounterMeasures are automated responses (actions) to incidents that are triggered in your infrastructure. They can be simple, such as attaching a `top` output to your incident; or, they can be more complex, like making an API call or restarting a service when a threshold is crossed. At its core, CounterMeasures is an extensible platform that allows you to increase the level of automation within your operations workflow - either by using the out of the box FortiMonitor CounterMeasures or by writing your own.

CounterMeasures can be run from an Agent (Linux/ Windows) or an OnSight vCollector.

A simple example

The CounterMeasures workflow is quite simple and is driven by the FortiMonitor monitoring agent.

1. Enable CounterMeasures (Linux | Windows) in your agent config file or manifest file.
2. Add a CounterMeasure to a metric threshold, such as `Disk % used.`
3. If the threshold is crossed, the CounterMeasure will be run.

Any output provided by the CounterMeasure will be attached to the incident record and will be available in the FortiMonitor control panel, in the incident log, and Slack.

Next steps

- Learn more about enabling CounterMeasures on your instances (Linux | Windows)
- Check out the actions that come with CounterMeasures out of the box (Linux | Windows)
- Dive into writing your custom CounterMeasure actions (Linux | Windows)

Linux CounterMeasures (FortiMonitor Agent)

As a quick reminder, CounterMeasures are automated responses to specific events that the FortiMonitor Agent detects on your infrastructure. We provide a handful of automated actions out-of-the-box and also allow you to write your own. Regardless of the route you choose, you will be leveraging the FortiMonitor Agent as the conduit for the process.

Enabling CounterMeasures

Agent Requirements

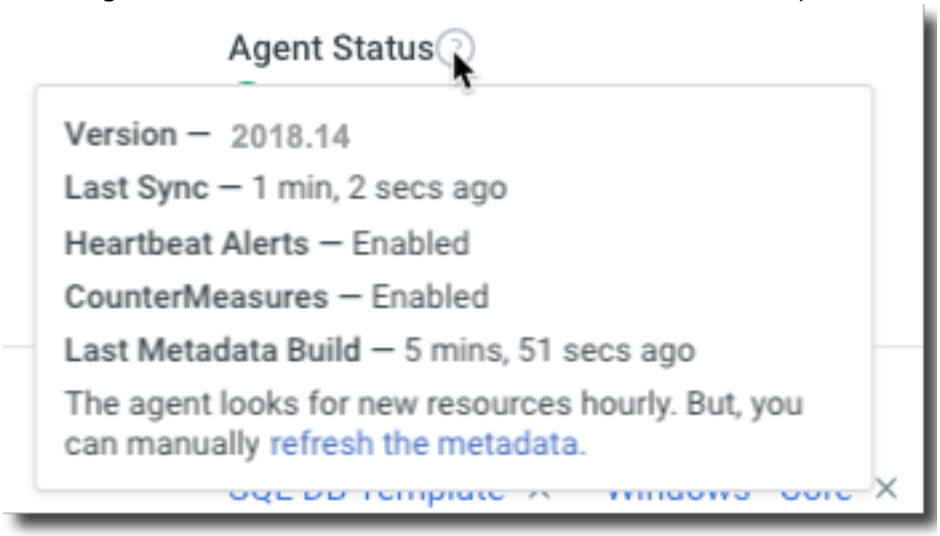
The latest version of the Linux Agent is required. If you are using an older version of the Agent, see this article to upgrade your Agent.

Fresh Install

If you're performing a new agent installation, you can enable CounterMeasures simply by adding `--countermeasures` onto the agent 1-line installer.

```
curl -s https://repo.fortimonitor.com/install/linux/fm_agent_install.sh | bash /dev/stdin -c
YOUR_CUSTOMER_KEY --countermeasures
```

The agent will now be installed and CounterMeasures will be ready to go on your instance. You can confirm that CounterMeasures have successfully been enabled by going to the **Instance Details** page. Click the question mark icon beside **Agent Status** and locate the **CounterMeasures - Enabled** datapoint in the **Agent Status** information block.



Existing Agent Install

Open your agent config file, located in `/etc/fm-agent/fm_agent.cfg`. Add the `enable_countermeasures` flag as a **new line** to the file.

```
enable_countermeasures=true
```

Next, run `python3 /usr/bin/fm-agent/fm_agent.py --rebuild-metadata`. This command resends metadata about your Agent installation up to FortiMonitor. You can also do this from the control panel.

The next time the agent runs (every 60 seconds), it will pick up the new flag and will be ready to go.

Configuring a CounterMeasure

CounterMeasures are tied to metric thresholds, which means they're configured together as well. Let's walk through how to add a CounterMeasures to capture `top` when CPU usage elevates.

1. In the control panel, navigate to an instance with CounterMeasures enabled.
2. Scroll down to the monitoring configuration table and locate the `Linux: CPU` section
3. Edit the **CPU: Usage percentage - Total** metric
4. In the slide out pane, expand the Thresholds & CounterMeasures section
5. If you don't already have a threshold set, go ahead and set one.

6. At the bottom of your configured threshold, click `+ Add CounterMeasure.`
7. Configure your CounterMeasure options:
   - **Delay:** this is how long to wait *after* the threshold is triggered before running the CounterMeasure. For now, select 0 minutes, which means it will fire immediately.
   - **Action:** this is the specific CounterMeasure that will be run. Select `top`, which is provided out of the box.
   - **Approval:** require approval before the CounterMeasure is run. You can read more about approval below.
   - **Send to Slack:** if you've previously sent an alert to Slack related to this incident, we'll also send any CounterMeasure output to this #channel.
8. Click **Save**.

Now you're good to go. When the threshold is crossed, the countermeasure will be triggered. We'll run `top`, collect the output, and display it in the log for the generated incident.

CounterMeasures and external checks

External monitoring checks (HTTP, HTTPs, Ping, etc) can also be used to trigger CounterMeasures. The agent still must be running on the instance, but when the external check threshold is crossed, it will trigger a CounterMeasure (e.g., restart Apache) on the instance.

Viewing CounterMeasure output

The output (if any) from your CounterMeasure action is available from a number of locations.

CounterMeasure Details Panel

Throughout the ControlPanel, you can open the CounterMeasures detail panel by clicking the CounterMeasures indicator icon, which is the beaker.



The beaker icon can be found in a number of locations - virtually anywhere an incident may be listed.

- Dashboards (Active Incident and Incident History widgets)
- Incident History Page
- Recent Incidents Tab on the Instance Details page
- Active Incidents Modal

Clicking the icon will open the CounterMeasures detail panel. It contains, among other things, any output the CounterMeasure returned after running.

Incident Details Page

CounterMeasures output is also available from the Incident details page. In the top navigation, select `Incidents`. From the table, select the incident you're interested in (you'll also noticed the beaker icon in the row). From the Incident details page, select `Outage Log`. In the log, you will find details about which CounterMeasures have been executed and what output the returned.

CounterMeasure States

CounterMeasures have a simple, linear lifecycle. Each lifecycle state is covered below

- **Pending:** state of an action that is scheduled to execute in the future and does not require approval or has already been approved. The beaker icon will appear grey.
- **Pending Approval:** state of an action that is scheduled to execute in the future but will not do so until approved. The beaker icon will glow orange.
- **In Process:** state of an action that has been synced down to the agent to execute. The beaker icon will appear grey.
- **Complete:** state of an action that has executed successfully. The beaker icon will appear green.
- **Error:** state of an action that attempted to execute but returned an error while executing. The beaker icon will appear red.
- **Skipped:** state of an action that was at one time pending but whose underlying incident was resolved before executing. The beaker icon will appear grey.

Approval

Certain situations may arise where you don't want a CounterMeasure to execute unless someone gives the final go-ahead. This is supported via the `Approval` option.

When configuring your CounterMeasure, check the `Require Approval` option. Anywhere the incident is displayed and the beaker indicator is present, the beaker will glow orange, indicating approval is required for at least one CounterMeasure on the incident. Clicking it will show a modal where you can approve the appropriate CounterMeasures. CounterMeasures that require approval will stay queued until approved; they will not block other CounterMeasure and will not run until their configured timing (e.g., you can approve them early and they'll still run at the proper time).

Using CounterMeasures in Templates

CounterMeasures work in templates the same way they would on a regular instance, except for one difference - in templates, the list of available actions is a union of all the CounterMeasures used across your account. On instances, the only actions available are the ones that have been reported by the instance.

If you attempt to apply a template with a CounterMeasure to an instance that has not reported that CounterMeasure as present, it will not be added to the monitoring configuration for that metric.

Utilizing Remote Actions

Centrally managing your custom CounterMeasure plugins is the best way to ensure consistency across your infrastructure monitoring. To help with this, you can add the `countermeasures_remote_plugins` parameter to your agent manifest file that points to an archive of agent plugins. When the agent is installed, it will download and extract the directory in the proper location. Zip, tar, and python file types are supported.

If you would like updates to your CounterMeasures plugins to automatically be applied, you can optionally set `countermeasures_refresh_plugins` to the number of hours for the refresh interval. This will help with keeping all your servers up-to-date.

Fortimonitoragent.manifest

```
customer_key = XXX-XXX-XXX-XXX
enable_countermeasures=true
countermeasures_remote_plugins = https://s3.amazonaws.com/some-s3-bucket/custom-plugins.zip
countermeasures_refresh_plugins = 6
```

Post agent installation

If you'd like to download a set of remote plugins after the installation process has been completed, you can run the following command:

```
python3 /usr/bin/fm-agent/countermeasure.py install_plugins --url URL
```

Then, rebuild your agent metadata.

```
python3 /usr/bin/fm-agent/fm_agent.py --rebuild-metadata
```

Once the metadata rebuild is complete, your plugins will be available for use in the control panel.

Standard Linux CounterMeasure actions (FortiMonitor Agent)

Out of the box, the FortiMonitor agent comes with a handful of standard CounterMeasure actions to use. You can view them using the following command:

```
python3 /usr/bin/fm-agent/countermeasure.py list_plugins
```

```
Available Countermeasures
=========================
Name                    Author                    Description
-------------------------------------------------------------------------------
Reboot Server           testing@fortinet.com       Reboot the server
dmesg                   testing@fortinet.com       Gather the latest lines from dmesg
netstat                 testing@fortinet.com       Gather most recent netstat output
top                     testing@fortinet.com       Gather most recent top output
vmstat                  testing@fortinet.com       Gather vmstat output
```

All of these will run without requiring further configuration, except for `Reboot Server`. Instructions on configuring the reboot server are detailed in the following section.

Configuring Reboot Server privileges

CounterMeasure actions are executed by the **fm-agent** user, which is created at the time of agent installation. The **fm-agent** user itself does not have elevated privileges and does not require them to perform it's normal monitoring tasks. However, one out-of-the-box CounterMeasure action requires elevated permissions - `reboot server`. If you attempt to run this CounterMeasure before you've configured permissions, it will fail.

Ubuntu

- Open `/etc/passwd`. At the end of the `fm-agent` line, remove `/usr/sbin/nologin` and replace it with `/bin/bash`
- Save the file

Make sure the following steps are taken using the visudo command, which validates file integrity when saving.

- Open `/etc/sudoers`. Under `User privilege specification`, add the following under the existing declaration:
  fm-agent ALL=(ALL) NOPASSWD: /sbin/shutdown
  fm-agent ALL=(ALL) NOPASSWD: /usr/sbin/
- Save the file

On a stock Ubuntu image, the `sudoers` file would now look like this:

```
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_
path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
fm-agent ALL=(ALL) NOPASSWD: /sbin/shutdown
fm-agent  ALL=(ALL) NOPASSWD: /usr/sbin/
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "#include" directives:
#includedir /etc/sudoers.d
```

Custom Linux CounterMeasure actions (FortiMonitor Agent)

You can create custom CounterMeasure actions for Linux using JSON or Python. In most cases, a CounterMeasure created using JSON will be sufficient. For more advanced or complex use cases, you have the option to use Python.

Here are some basic CounterMeasure actions that you can create using JSON:

- Information gathering commands (top, netstat, reading a log file, etc.)
- Restarting a service or server
- Deleting files

For advanced use cases such as the following, use Python:

- Authentication
- Interacting with APIs
- Managing state over multiple steps

Create CounterMeasure actions using JSON

To create custom Linux CounterMeasures using JSON, create a new JSON file in the `/usr/share/fm-agent/countermeasures` directory. The command provided in the `command` key-value will be executed when the CounterMeasure is triggered. The command's output (if any) will be returned and available in the FortiMonitor Control Panel. The following example shows a JSON CounterMeasure that returns the output of a **netstat** command.

```
{
"name": "netstat",
"textkey": "info.netstat",
"description": "Gather most recent netstat output",
"max_frequency": 60,
"max_runtime": null,
"author": "testing@fortinet.com",
"wall_announce_delay": null,
"command": "netstat -ant"
}
```

For more details, see Implementation Reference. Keep in mind that the command will be run as the fm-agent user. Any elevated privileges will need to be configured as described in the sudo privileges section.

Create CounterMeasure actions using Python

To create your own CounterMeasure:

1. Create a new python file in the `/usr/share/fm-agent/countermeasures` directory and ensure your implementation subclasses the `CountermeasurePlugin` class.
   A full reference is available in the Implementation Reference section. Largely, implementing the **run()** method and providing a few instance variables is all that is required. You can see a basic example in CounterMeasure Action metadata example.

The name of your custom class needs to end with **Countermeasure**.

2. Run the following command, where **my_script.py** is the name of your CounterMeasure script:
`python3 -m py_compile my_script.py`

Ensure that there are no syntax errors. If your CounterMeasure script has syntax errors, it will not be added to the control panel.

3. Rebuild your Agent metadata with the following command:
`python3 /usr/bin/fm-agent/fm_agent.py --rebuild-metadata`

Your CounterMeasure will then be available in the FortiMonitor control panel.

Logging

If you need to log information in your CounterMeasure plugin, you can use `self.log.info("your message here")`, which will log to `/var/log/fm-agent/countermeasure.log`

Leverage incident data

When the agent is notified that it should run a local CounterMeasure action, it also receives metadata about the underlying incident triggering the CounterMeasure. This JSON object is available to you in your code via the **metadata** property. For example, the below code returns the incident metadata right back to FortiMonitor.

CounterMeasure Action metadata example

```
from CountermeasurePlugin import CountermeasurePlugin
class TestMetaCountermeasure(CountermeasurePlugin):
name = "Test Metadata"
author = "FortiMonitor User"
textkey = "test.metadata"
description = "Returns passed metadata"
def run(self):
output = str(self.metadata)
self.save_text_output(output)
self.save_return_code(0)
```

This is helpful as it allows you to take action based on certain criteria, such as which application or metrics triggered the incident. The payload schema is detailed in the following table.

Key	Description
Outage	
id	The ID number of the associated incident.
alert_label	Alert label of the incident/anomaly.
timestamp	UTC timestamp of when the incident/clear occurred.
severity	The severity of the outage/anomaly, either "critical" or "warning".
reasons	The reasons for the network service incidents or the details for anomalies.
Server	

id	The ID number of the server experiencing the incident/clear.
server_key	The server key for the server.
fqdn	The fully qualified domain name of the server experiencing the incident/clear.
name	Name of the server experiencing the incident/clear.
tags	The tags for the server.
partner_server_id	The partner server id for the server.
Customer attributes	
metrics	Services experiencing the incident/clear or resources experiencing the anomaly/clear.
metric_tags	The tags for all of the metrics involved in the outage.
resource	For resource anomalies: resources experiencing the anomaly/clear.
services	For service incident: services experiencing the incident/clear.

Limit plugin execution time

You may optionally set the maximum time a CounterMeasure plugin may run using the `max_runtime` property. You may supply any number of seconds, represented as an integer. If it exceeds the allotted time, the CounterMeasure driver will attempt to kill it, though it is not guaranteed

View plugins on an instance

To view all the plugins available to use on an instance, use the following command:

```
python /usr/bin/fm-agent/countermeasure.py list_plugins
```

If you're not seeing an expected plugin, ensure that it is in `/usr/share/fm-agent/countermeasures`.

sudo privileges

CounterMeasure plugins are executed by the **fm-agent** user, which is created at the time of agent installation. The **fm-agent** user itself does not have elevated privileges and does not require them to perform it's normal tasks. There may be times, however, when creating custom CounterMeasure plugins that you need the **fm-agent** user to have elevated privileges - this requires passwordless access to be configured for the **fm-agent** user (an example is below). As well, one out-of-the-box CounterMeasure plugin requires elevated permissions - **reboot**. If you attempt to run this CounterMeasure before you've configured permissions, it will fail.

Verifying sudo privileges

The Agent ships with a CLI helper method to allows you to validate sudo privileges for your CounterMeasure plugins. It will help you identify plugins that the **fm-agent** user does not have the proper privileges to run.

First, we need to enable switching to the `fm-agent` user.

- Open `/etc/passwd`. At the end of the `fm-agent` line, remove `/usr/sbin/nologin` and replace it with `/bin/bash`
- Save the file.

Now we can switch to the `fm-agent` user:

```
su fm-agent
```

Then, we can run the following command, which will highlight the plugins that require sudo privileges and whether or not those privileges are configured correctly.

```
python3 /usr/bin/fm-agent/countermeasure.py validate_sudo
```

If you've properly configured sudoers, you'll see an output like below. Otherwise, you'll see **Missing Permissions**.

```
Verifying sudo requirements for reboot
reboot:shutdown                Pass
```

Implementation reference

Instance Variable	Type	Description	Required
name	String	Human-readable name for the Countermeasure, will be displayed in the control panel and alerts	Yes
author	String	Identifier of the author (recommended to be your email address)	Yes
textkey	String	Unique identifier for the countermeasure, should be lowercase letters, numbers, underscores, and periods. No spaces allowed	Yes
description	String	Description of the countermeasure, for display at command line and in the FortiMonitor control panel	Yes

wall_announce_delay	Int	How long to pause execution of the countermeasure after announcing it as a wall message. Set to None to disable wall announcements for this countermeasure	No
max_frequency	Int	The shortest allowed time between two executions of this plugin, in seconds. If less than that time has elapsed, the second execution won't be performed. Leave set to None to disable frequency checks	No
max_runtime	Int	The longest amount of time that the plugin should be allowed to run. The Countermeasures driver will attempt to kill the execution when it exceeds this time, although due to the condition that may not be guaranteed	No
sudo_requirements	[Command Lines]	List of full command lines this plugin requires sudo access for, which is used for validating sudo configurations	No

Method	Parameters	Description	Required
run	none	Execute the countermeasure action	Yes

validate	none	Method to perform validation on the plugin's setup. This is called by the command-line tool's "validate-plugins" command. Mainly used by helper subclasses that intend to have some additional properties overridden. Should return nothing if the plugin is valid, or a string describing validation issues if there are problems.	No
prepare	none	Method to be run before execution, for any initial setup or validation that the countermeasure action needs to perform	No
save_text_output	output (String)	Save countermeasure output as plain text for later publishing up to the FortiMonitor cloud	No
save_html_output	output (String)	Save countermeasure output as formatted HTML for later publishing up to the FortiMonitor cloud	No
save_return_code	return_code	Save the return code from the countermeasure execution	No

Windows CounterMeasures (FortiMonitor Agent)

As a quick reminder, CounterMeasures are automated responses to specific events that the FortiMonitor agent detects on your infrastructure. We provide a handful of automated actions out-of-the-box and also allow you to write your own. Regardless of the route you choose, you will be leveraging the FortiMonitor agent as the conduit for the process.

Enabling CounterMeasures

Agent Requirements

The latest version of the Windows Agent is required. If you are using an older version of the Agent, see this article to upgrade your Agent.

Fresh Install

If you're performing a new agent installation, you can enable CounterMeasures in the FortiMonitor Agent manifest file and then install the agent.

C:\FortimonitorAgent.manifest

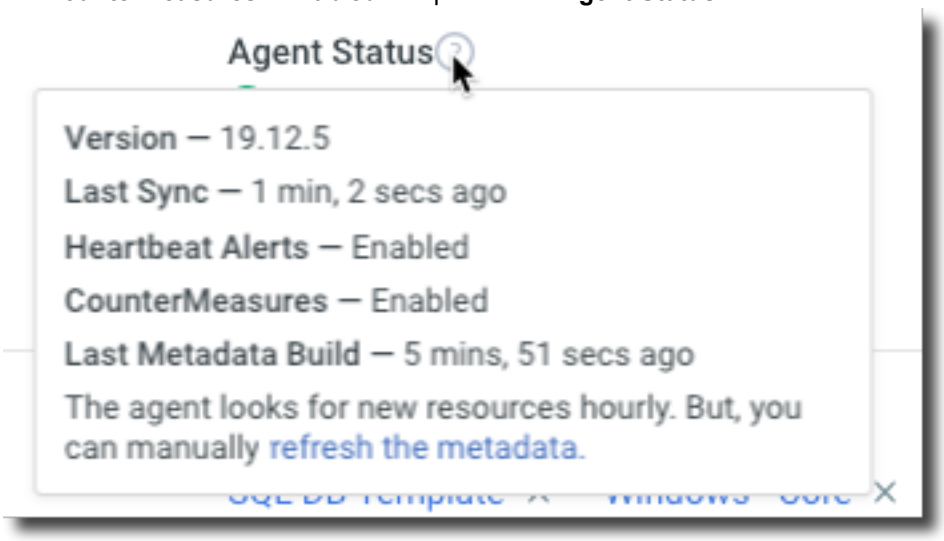
```
customer_key=XXX-XXXX-XXXX-XXXX
enable_countermeasures=true
```

Let's run through a quick example:

1. Create an agent manifest file like the one above - with your customer key - at `C:\FortimonitorAgent.manifest`
2. Run the 1-line agent install command, replacing `CUSTOMER_KEY` with your own.

```
Invoke-WebRequest https://repo.fortimonitor.com/install/win/fortimonitor_agent_windows.ps1 -
OutFile fortimonitor_agent_windows.ps1; .\fortimonitor_agent_windows.ps1 -customer_key
<CUSTOMER_KEY> -server_key NONE
```

CounterMeasures is now ready to go on your instance. You can confirm that CounterMeasures have successfully been enabled by going to the **Instance Details** page. Click the question mark icon beside **Agent Status** and locate the **CounterMeasures - Enabled** datapoint in the **Agent Status** information block.



Existing Agent Install

1. Open your agent config file, located at `C:\Program Files (x86)\FortimonitorAgent\Agent.config`.
2. After opening the agent config file, locate the entry shown below (if you cannot find this entry, create it under the **<service>** header)

```
<add key="EnableCounterMeasures" value="" />
```

3. Update this entry with the new value shown below.

```
<add key="EnableCounterMeasures" value="true" />
```

4. Save the file.

5. Restart the Agent service within the Services manager.

Configuring a CounterMeasure

CounterMeasures are tied to metric thresholds, which means they're configured together as well. Let's walk through how to add a CounterMeasures to capture **top** when CPU usage elevates.

1. In the control panel, navigate to an instance with CounterMeasures enabled.
2. Scroll down to the monitoring configuration table and locate the **Windows: CPU** section
3. Edit the `CPU: Percent used - _Total` metric
4. In the slide-out pane, expand the Thresholds & CounterMeasures section
5. If you don't already have a threshold set, go ahead and set one.
6. At the bottom of your configured threshold, click `+ Add CounterMeasure.`
1. Configure your two CounterMeasure options
   - **Delay:** this is how long to wait *after* the threshold is triggered before running the CounterMeasure. For now, select 0 minutes, which means it will fire immediately.
   - **Action:** this is the specific CounterMeasure that will be run. Select **top**, which is provided out of the box.
   - **Approval:** require approval before the CounterMeasure is run. You can read more about approval below.
   - **Send to Slack:** if you've previously sent an alert to Slack related to this incident, we'll also send any CounterMeasure output to this #channel.
2. Hit **Save**.

Now you're good to go. When the threshold is crossed, the countermeasure will be triggered. We'll run **top**, collect the output, and display it in the log for the generated incident.

CounterMeasures and external checks

External monitoring checks (HTTP, HTTPs, Ping, etc) can also be used to trigger CounterMeasures. The agent still must be running on the instance, but when the external check threshold is crossed, it will trigger a CounterMeasure (e.g., restart Apache) on the instance.

Viewing CounterMeasure output

The output (if any) from your CounterMeasure action is available from a number of locations.

CounterMeasure Details Panel

Throughout the Control Panel, you can open the CounterMeasures detail panel by clicking the CounterMeasures indicator icon, which is the beaker.



The beaker icon can be found in a number of locations - virtually anywhere an incident may be listed.

- Dashboards (Active Incident and Incident History widgets)
- Incident History Page
- Recent Incidents Tab on the Instance Details page
- Active Incidents Modal

Clicking the icon will open the CounterMeasures detail panel. It contains, among other things, any output the CounterMeasure returned after running.

Incident Details Page

CounterMeasures output is also available from the Incident details page. In the top navigation, select `Incidents`. From the table, select the incident you're interested in (you'll also noticed the beaker icon in the row). From the Incident details page, select `Outage Log`. In the log, you will find details about which CounterMeasures have been executed and what output the returned.

CounterMeasure States

CounterMeasures have a simple, linear lifecycle. Each lifecycle state is covered below

- **Pending:** state of an action that is scheduled to execute in the future and does not require approval or has already been approved. The beaker icon will appear grey.
- **Pending Approval:** state of an action that is scheduled to execute in the future but will not do so until approved. The beaker icon will glow orange.
- **In Process:** state of an action that has been synced down to the agent to execute. The beaker icon will appear grey.
- **Complete:** state of an action that has executed successfully. The beaker icon will appear green.
- **Error:** state of an action that attempted to execute but returned an error while executing. The beaker icon will appear red.
- **Skipped:** state of an action that was at one time pending but whose underlying incident was resolved before executing. The beaker icon will appear grey.

Approval

Certain situations may arise where you don't want a CounterMeasure to execute unless someone gives the final go-ahead. This is supported via the `Approval` option.

When configuring your CounterMeasure, check the `Require Approval` option. Anywhere the incident is displayed and the beaker indicator is present, the beaker will glow orange, indicating approval is required for at least one CounterMeasure on the incident. Clicking it will show a modal where you can approve the appropriate CounterMeasures. CounterMeasures that require approval will stay queued until approved; they will not block other CounterMeasure and will not run until their configured timing (e.g., you can approve them early and they'll still run at the proper time).

Using CounterMeasures in Templates

CounterMeasures work in templates the same way they would on a regular instance, except for one difference - in templates, the list of available actions is a union of all the CounterMeasures used across your account. On instances, the only actions available are the ones that have been reported by the instance.

If you attempt to apply a template with a CounterMeasure to an instance that has not reported that CounterMeasure as present, it will not be added to the monitoring configuration for that metric.

Standard Windows CounterMeasure Actions

Out of the box, the Windows Agent comes with a handful of standard CounterMeasure actions to use. Also, keep in mind that it's easy to write your own.

Action	Description
Top	Analogous to top on Linux, displays information about CPU and memory utilization by process
Application Error Logs	Grabs the last 50 Application Error logs
Get Service Status	Retrieves the status of all running services
Netstat	Displays network connections for the TCP, routing tables, and a number of network interface and network protocol statistics
Reboot	Reboots the server
Restart IIS	Restarts the IIS web server service
System Error Logs	Grabs the last 50 System Error logs
Vmstat	Analogous to vmstat on Linux, grabs the following virtual statistics: processes, memory, paging, block IO, traps, and CPU activity

Custom Windows CounterMeasures actions (FortiMonitor Agent)

You can create custom CounterMeasure actions for Windows using JSON or PowerShell. In most cases, a CounterMeasure created using JSON will be sufficient. For more advanced or complex use cases, you have the option to use PowerShell.

Here are some basic CounterMeasure actions that you can create using JSON:

- Information gathering commands (Test-NetConnection –TraceRoute, Get-NetTCPConnection, etc.)
- Restarting a service or server
- Deleting files

Create CounterMeasure actions using JSON

To create custom Windows CounterMeasures using JSON, create a new JSON file in the **C:\Program Files (x86)\FortimonitorAgent\cm_ps_plugins** directory. The command provided in the `command` key-value will be executed when the CounterMeasure is triggered. The command's output (if any) will be returned and available in the FortiMonitor Control Panel. The following example shows a JSON CounterMeasure that returns the output of a **Get-ChildItem** command.

```
{
"name": "Detailed Dir List",
"textkey": "info.detailed_dir_list",
"description": "Returns a detailed list of the current directory",
"max_frequency": 60,
"max_runtime": null,
"author": "testing@fortinet.com",
"command": "Get-ChildItem -Path C:\\Test"
}
```

For more details, see Plugin-Configuration Properties. Keep in mind that the command will be run as the **fm-agent** user.

Create CounterMeasure actions using PowerShell

To create your own CounterMeasure:

1. Create a new .ps1 file in the **C:\Program Files (x86)\FortimonitorAgent\cm_ps_plugins** directory. At a minimum, your CounterMeasure needs to implement two functions **- Plugin-Configuration**, which provides config and metadata about your CounterMeasure and Execute, which is the driver for your CounterMeasure.
2. Check for syntax errors by running the script in the PowerShell ISE. If the syntax is correct, the execution should not generate any output or error message. If your CounterMeasure has syntax errors, it will not be added to the control panel.
3. Rebuild your agent metadata via the instance's details page in the FortiMonitor control panel. Your custom plugin will then be available in the FortiMonitor control panel.

CustomCounterMeasure.ps1

```
function Plugin-Configuration {
	$properties = @{}
	#Required: a name for your custom CounterMeasure
	$properties.add("name", "My Custom CounterMeasure")
	#Required: a description of your custom CounterMeasure
	$properties.add("description", "Returns the current dir list")
	#Required: a distinct textkey for your custom CounterMeasure
	$properties.add("textkey", "Fortimonitor.ListDirectoryCM")
	#Required: the name of the CounterMeasure Author
```

```
        $properties.add("author", "Fortimonitor.HeadDeveloper")
        #Optional: The maximum frequency at which a CounterMeasure can run
        $properties.add("max_frequency", "60")
        # Required
        return $properties
}
function Execute {
        # The Metadata param will receive information about the incident the CM is responding
to
        param([hashtable]$metadata)
        # Dictionary to to hold the returned CM payload - you can name this anything
        $returndata = @{}
        # Dictionary to hold each chunk of information you want to return - you can name this
anything
        $output = @{}
        # Array to hold your 1..n dictionaries of output - you can name this anything
        $output_list = New-Object System.Collections.ArrayList
        # Required: either success or error. key must be return_code
        $returndata.add("return_code", "success")
        # Required: output type the data you are returning. either text or html. key must be
format
        $output.add("format", "text")
        # Required: output content. key must be format
        $output.add("output", "$(Get-ChildItem | Format-Table -Property Name, Length | Out-
String)")
        # add your output chunks to your array
        $output_list.add($output);
        # add your array to your top-level dictionary. key must be output
        $returndata.add("output", $output_list)
        # return your countermeasure data
        return $returndata
}
```

Leveraging Incident Data

When the agent is notified that it should run a local CounterMeasure action, it receives metadata about the incident triggering the CounterMeasure. This metadata object is available to you in your code via the metadata parameter. For example, the below code just returns the incident **metadata** right back to FortiMonitor.

CounterMeasure return metadata

```
function Plugin-Configuration {
        $properties = @{}
        $properties.add("name", "Return mmeta")
        $properties.add("description", "Returns the invoking incident's md")
        $properties.add("textkey", "Fortimonitor.MetadataCM")
        $properties.add("author", "Fortimonitor.HeadDeveloper")

return $properties
}
function Execute {
        # The Metadata param will receive information about the incident the CM is responding to
        param([hashtable]$metadata)
                $returndata = @{}
                $output = @{}
                $output_list = New-Object System.Collections.ArrayList

                $returndata.add("return_code", "success")
                $output.add("format", "text")
                $output.add("output", metadata)
                $output_list.add($output);

                $returndata.add("output", $output_list)
return $returndata
}
```

This is helpful as it allows you to take action based on certain criteria, such as which application or metrics triggered the incident. The payload scheme is included below. The hashtable properties are listed in the table below:

CounterMeasures incident metadata

Items in the incident metadata can be accessed from the **$metadata** variable.

Key	Description
$metadata.outage_id	The ID number of the associated incident.
$metadata.outage_alert_label	Alert label of the incident/anomaly.

$metadata.outage_timestamp	UTC timestamp of when the incident/clear occurred.
$metadata.outage_severity	The severity of the outage/anomaly, either "critical" or "warning".
$metadata.outage_reasons	The reason for the outage.
$metadata.server_id	The ID number of the server experiencing the incident/clear.
$metadata.server_server_key	The server key for the server.
$metadata.server_fqdn	The fully qualified domain name of the server experiencing the incident/clear.
$metadata.server_name	Name of the server experiencing the incident/clear.
$metadata.server_tags	The tags for the server.
$metadata.server_partner_server_id	The partner server ID for the server.
$metadata.metrics	Services experiencing the incident/clear or resources experiencing the anomaly/clear.
$metadata.metric_tags	The tags for all of the metrics involved in the outage.
$metadata.resource	For resource anomalies: resources experiencing the anomaly/clear.
$metadata.resource_item_type	The server resource item type.
$metadata.resource_name	The server resource name.
$metadata.resource_resource_option	if your metric utilizes an option, such as the mount point, NIC, or disk drive, it will be present here.
$metadata.services	For service incident: services experiencing the incident/clear.

Plugin-Configuration Properties

Parameters	Type	Description	Required
name	String	A human-readable name for the CounterMeasure, will be displayed in the control panel and alerts	Yes
author	String	Identifier of the author (recommended to be your email address)	Yes

textkey	String	Unique identifier for the CounterMeasure, should be lowercase letters, numbers, underscores, and periods. No spaces allowed	Yes
description	String	Description of the countermeasure, for display at the command line and in the FortiMonitor control panel	Yes
max_frequency	String	The shortest allowed time between two executions of this plugin, in seconds. If less than that time has elapsed, the second execution won't be performed. Leave set to None to disable frequency checks	No

Execute Properties

The **Execute** function returns a dictionary, which contains two things: the CounterMeasure execution status and an array of output items. You can name the returned dictionary whatever you like, but for clarity, we'll use the property name **$returndata**.

Parameter	Type	Description	Required
ReturnData["return_code"]	String	Execution status of the CounterMeasure - either success or error	Yes
ReturnData[outputs_list]	Array of Dictionaries	An array output chunks that you'd like returned for viewing in the FortiMonitor control panel	Yes
outputs_list["format"]	String	Either text or HTML	Yes
outputs_list["output"]	String	Output to be returned	Yes

Plugin Functions

Function	Parameters	Description	Required
Execution	none	Executes the countermeasure action	Yes

Plugin-Configuration	none	Returns CounterMeasure plugin configuration information	Yes

CounterMeasures Security

As is with the rest of FortiMonitor, security was (and is) top of mind when implementing CounterMeasures. A couple of key considerations were made to ensure a secure experience:

- **SSL communication:** all communication between the agent and FortiMonitor is over SSL (this goes for all agent communication)
- **All Outbound Communication:** the agent only requires outbound communication and only receives information in response to requests made to FortiMonitor (this goes for all agent communication)
- **Only local code is executed:** Only code that is located on the instance is executed. Nothing that is retrieved from FortiMonitor is executed

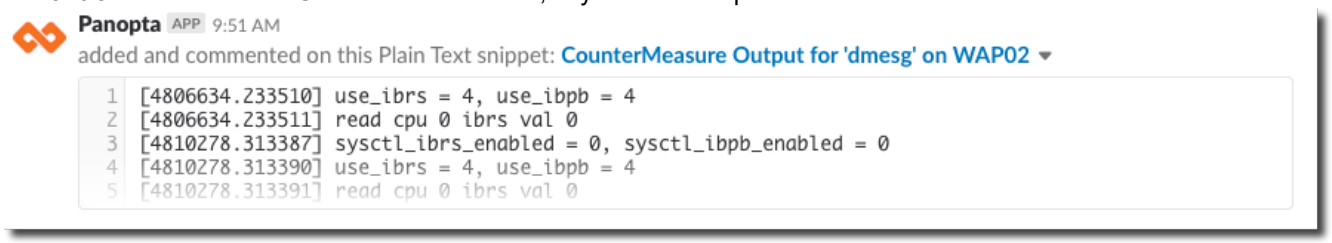
CounterMeasures Slack Integration

CounterMeasures often return valuable information that is useful when debugging incidents. Quick access to the output is key and is why we allow you to send it to Slack.

Setup

Sending CounterMeasure output to Slack relies on Alert Timelines - which makes integrating as easy as adding a new event on your timeline.

- The first step is to make sure you have the most recent Slack integration, which is the FortiMonitor Slack App. Check out the docs if you're using an older version or need a fresh install.
- Add your Slack integration to any Alert Timelines you'd like to receive CounterMeasure output.
- When configuring a CounterMeasure that has output you'd like sent to Slack, check the `Send Output to Slack` box. When the CounterMeasure runs, any returned output will now be sent to Slack.



OnSight vCollector

The OnSight vCollector is a lightweight appliance that functions similarly to our public monitoring nodes, which runs uptime checks and synthetic monitoring against your publicly available servers and web endpoints. However, the OnSight vCollector sits within your private network and monitors resources that are not publicly exposed. The OnSight can run uptime and synthetic checks on your private infrastructure, monitoring things like RDP, database connectivity, and port availability.

The OnSight can also function as a proxy for servers that have the Server Agent installed but have no outbound connectivity. Also, the OnSight can monitor network devices, storage arrays, VMWare clusters, and private network-managed cloud resources such as RDS and DynamoDB.

The OnSight is a collection of Docker containers that provide the necessary services. During the installation of the OnSight, if Docker is not found on the target machine it will automatically be installed.

Note: Only Docker version 23.0 and above are supported.

System requirements

OS requirements

The OnSight vCollector supports the following Linux distributions:

Note: The OnSight only supports operating systems that are actively under standard support from their corresponding vendors.

- Debian 9 and 10
- Ubuntu Server 20.04 LTS, 24.04 LTS, and 24.10
- Redhat Enterprise 7.x, 8.0-8.9, and 9.0-9.6
- Oracle Enterprise Linux 7.0, 9.0-9.6
- Amazon Linux 2 and 2018.03
- Rocky Linux 9.3-9.6

Please note that podman should not be installed on the OS. We only use Docker currently. The installation process will install it if it is not found.

The docker SNAP package requires additional configuration that may be environment dependent. We currently only support the docker.io package installation. If the SNAP version was installed during installation, you should remove it and install the docker.io package using your package manager.

Resource requirements

These are the minimum resources required to install the OnSight.

- **Processor:** 4 cores
- **Memory**: 8 GB
- **Hard disk space**: 50 GB

Note: You can use the **SMALL=1** parameter to install the OnSight to machines that only have 2 cores and 4 GB of memory. This can be used for deployments with less than 1000 metrics. Command example:

```
curl -fsSL https://repo.fortimonitor.com/install/onsight/onsight-install.sh > onsight-install.sh && SMALL=1 bash onsight-install.sh <CUSTOMER KEY>
```

For larger deployments involving a significant number of network devices or advanced synthetic checks, please contact our support team to help determine the resource requirements for your environment.

Network connectivity requirements

The host machine must have outbound access to the following:

Port	Destination	Used for
443	rx.us01.fortimonitor.com (US region) rx.eu01.fortimonitor.com (EU region) aggregator2.panopta.com aggregator2-secondary.panopta.com global.fortimonitor.com	Primary and fail-over communication with the FortiMonitor cloud
443, 80	repo.fortimonitor.com	OnSight software updates
443	hub.docker.com	OnSight software updates
443	download.docker.com get.docker.com *.docker.io *.cloudfront.net *.cloudflare.docker.com	Docker download and installation

The host machine must have inbound access to the following ports allowed:

Port	Protocol	Usage
22	TCP	Used to SSH into host
8443	TCP	Used to configure the OnSight as a proxy for agents
162	UDP	Used for SNMP Trap receipt
444	TCP	Used for NCM console

Docker IP change requirements

The OnSight vCollector uses Docker. The default network space defined by Docker is **172.17.0.0/12** with Docker using **172.17.0.1/16** as its IP address and the first container using the **172.17.0.2/16** IP address.

This can conflict with the your internal network IP space and cause the OnSight to not be able to communicate with the your network devices. Specifically, if you have any hosts in the **172.16.0.0/12** space, as defined by RFC 1918, then you must change your Docker configuration to allow for the OnSight to successfully reach those networks.

To resolve this, update your Docker host to use a different IP range.

Note: The following must be performed **before** installing the OnSight vCollector.

1. Add a file (or modify if it exists): **/etc/docker/daemon.json**

```
{
   "default-address-pools":
   [
   {"base":"192.168.0.0/16","size":24}
   ]
}
```

Where:

- **base** is the network to allocate to docker created and managed networks.
- **size** is the netmask of each block created within the base.

  In the above example this would use 192.168.0.0/16 for all networks and an individual network would have a netmask of /24.

2. After adding or modifying the above file, restart the docker service. The docker service and all the associated iptables NAT rules will get set up accordingly.

Deployment methods

Installation Method	Best for
One-liner installer	You're comfortable operating in a Linux environment You have a specific server image you want to conform to You have an available Linux VM
AWS AMI	You're deploying the OnSight in an AWS environment You're looking for the simplest cloud deployment method
Virtual image	You prefer a "black box" deployment You have the available resources to allocate to a VM You'd like to deploy on the HyperVisor of your choosing

One-liner installer

Using the one-liner installer allows you to install the OnSight vCollector on any qualifying Linux-based instance.

Installation

Install the OnSight by running the following command as **root**:

```
curl -fsSL http://repo.fortimonitor.com/install/onsight/onsight-install.sh > onsight-install.sh
&& \
```

```
bash onsight-install.sh <CUSTOMER-KEY>
```

where **<YOUR CUSTOMER KEY>** is your own customer key. To obtain your customer key, visit the FortiMonitor Control Panel and click your avatar then select **My Account**.

This will add the appropriate repositories to the VM and begin downloading and configuring dependencies. This process should only take 2 - 3 minutes. Once the installation has completed, you will see a message similar to the following:

```
OnSight installed, registering with FortiMonitor
Processing manifest file in /etc/panopta-onsight-manifest
Registering new OnSight
Registered OnSight, configuring vCollector with key xxxx-xxxx-xxxx-xxxx.
Pulling latest vCollector image, version 2020.45.3
Starting OnSight vCollector version 2020.45.3
OnSight can be managed at https://my.panopta.com/report/ApplianceReport?appliance_id=4217
```

The OnSight vCollector can be managed from the Control Panel. Follow the link outputted at the bottom of the installation success message.

OnSight virtual image installation

To deploy an OnSight vCollector as an image, you need to download the VMware image from the FortiMonitor package server and import the image into a designated hypervisor. vSphere 6.5 and above is required for the host to import the image. The system requirements are detailed here.

To install the OnSight vCollector, perform the following steps:

1. Download the OnSight image.
2. After downloading the OnSight image, import it as a virtual machine into your hypervisor. Once your OnSight is imported and booted, the VM will go through the normal Linux start up process, finishing with a login prompt.
   a. Log in with username **fortimonitor** and password **fortimonitor**. You will then be prompted to set a new password.
   Important note: Do not lose this password. Without it, there is no way to access the OnSight for further updates.
   b. Configure the network by running `sudo onsight configure-network` as root. See Network Configuration for more details.
   c. Run updates to pick up any recently released OS patches by running `sudo apt update; sudo apt upgrade`.
   d. Reboot the server by running `shutdown -r now`
   e. Register the OnSight by running `sudo onsight register --customer-key <customer-key>` as root. Your OnSight appliance key will also be displayed.
   where **<customer-key>** is your own customer key. To obtain your customer key, visit the Control Panel and click your avatar then select **My Account**.
3. Proceed through the final steps of the OnSight installation process in the Control Panel, where you can assign things such as the **Alert Timeline** and **Instance Group** to use for your new OnSight.
   The OnSight will begin syncing immediately after deployment.

Network configuration

By default, the OnSight vCollector will attempt to use DHCP except in AWS, which pre-configures networking correctly. Basic network configuration is available by running:

```
sudo onsight configure-network
```

This will walk you through a series of steps that allows you to configure the following:

- Select a network interface for the OnSight
- Use static IP or DHCP
- IP address
- Gateway
- Nameservers

For complex use cases like bridging multiple networks or custom route configurations, you need to manually configure the network. Please contact our support team if you need assistance.

NTP server configuration

The OnSight uses the default Ubuntu NTP servers for time synchronization. If your environment blocks outbound NTP access, you can configure the OnSight to use an internal NTP server using the following:

```
sudo onsight configure-ntp
```

This will request a list of servers to use then update the configuration and restart the NTP daemon to pick up the new servers.

OnSight AMI installation

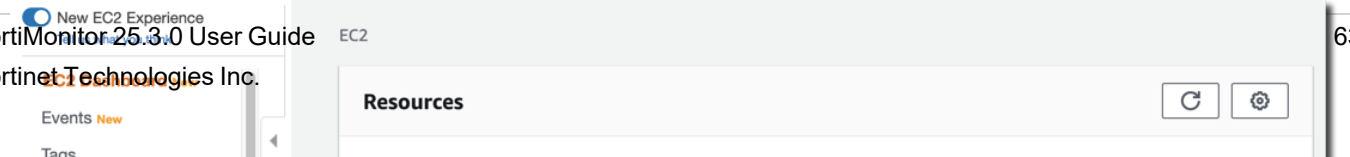
The FortiMonitor OnSight AMI allows you to deploy and run an OnSight from an AWS instance. The following sections list the resource requirements and deployment steps.

Resource requirements

- Minimum of a t3.medium instance - as your monitoring expands, we recommend 1 core + 1 GB Ram per 1,000 checks
- Network access to packages.panopta.com (165.227.86.161 and 162.243.0.67)
- Network connectivity requirements

OnSight installation

1. Log in to your AWS account and switch to the EC2 console by selecting **EC2** from the splash screen or from under the **Services** navigation menu.



2. Switch to the AWS Marketplace by clicking **Launch Instance**, and then selecting the **AWS Marketplace** tab.

3. Search for **FortiMonitor** then select **FortiMonitor OnSight for AWS**.

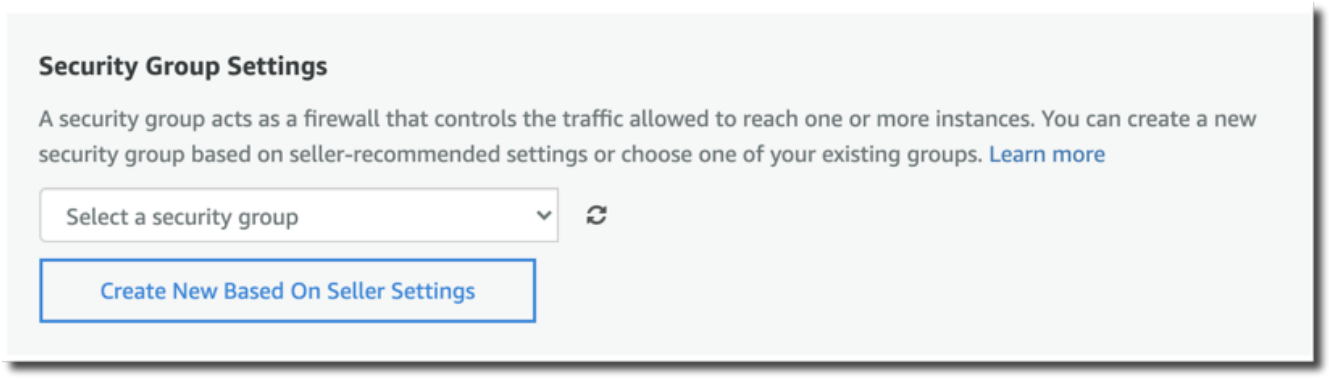
4. Review and accept the subscription terms. The configuration page will open.

5. Click **Continue to Launch**. The following page will allow you to review your configuration and choose how you wish to launch the software.

6. Select your instance size - a **t3.medium** is sufficient to start out.

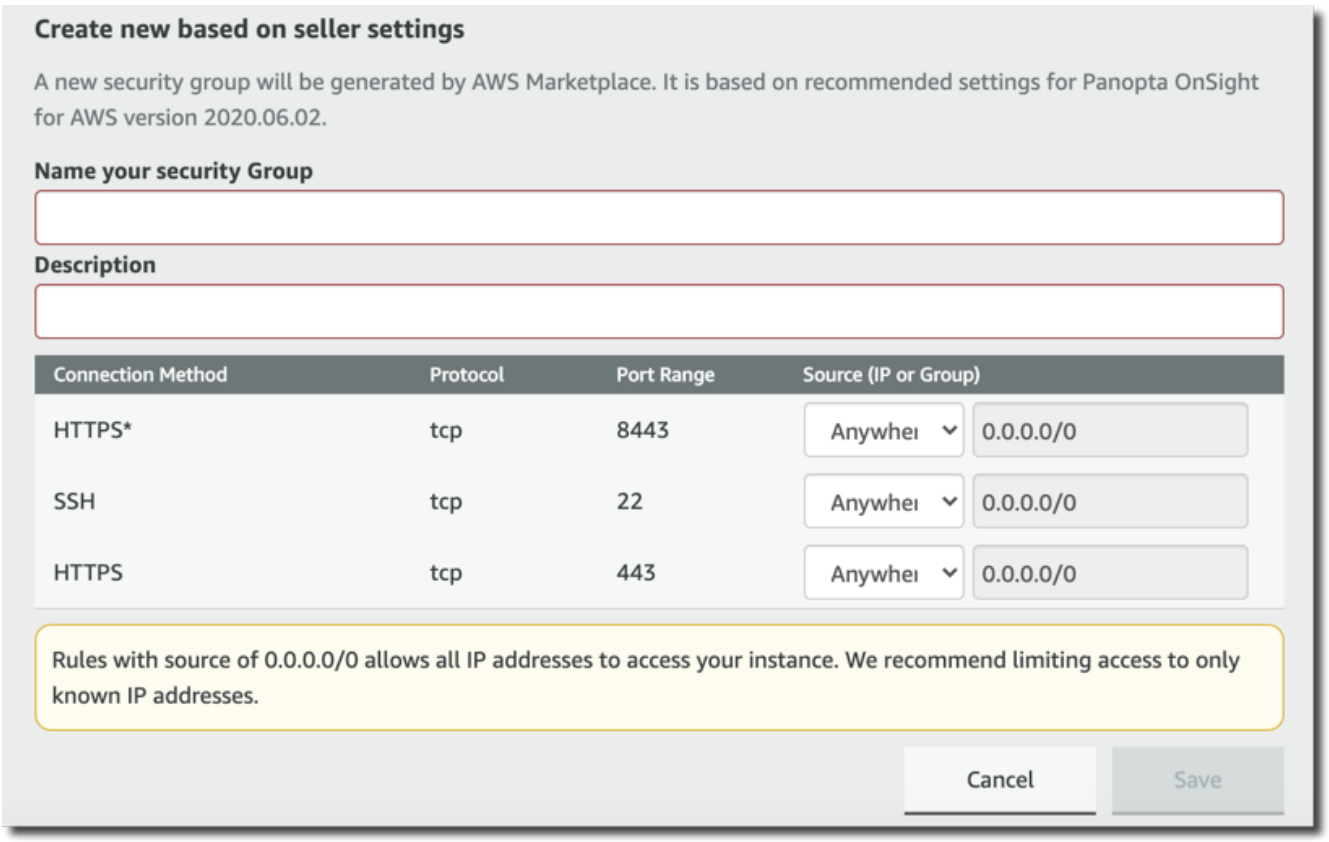
As your monitoring expands, we recommend 1 core + 1 GB Ram per 1,000 checks.

7. Under **Security Group Settings**, click **Create New Based on Seller Settings**.



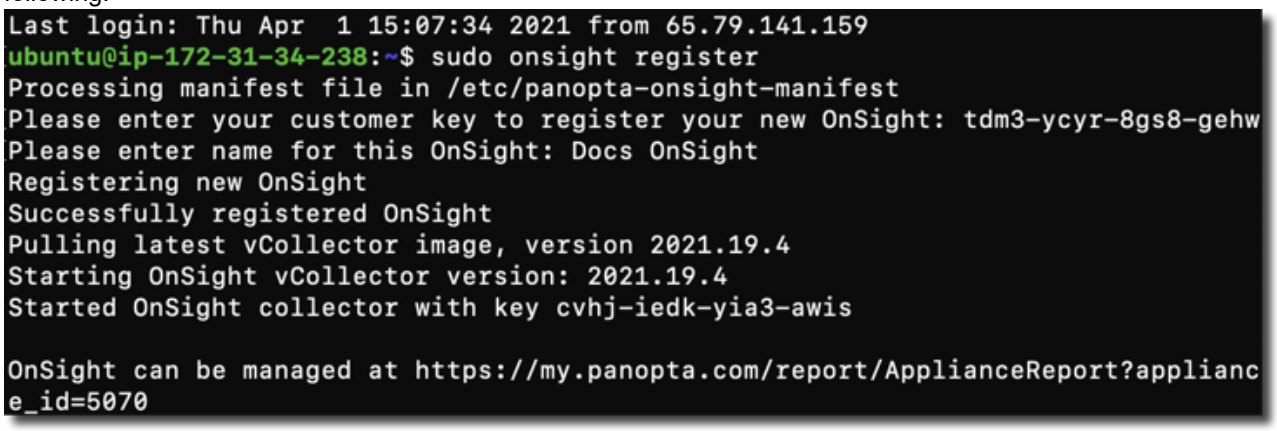
The only ports that need to be open are SSH, HTTPS and port 8443/TCP.

Type	Protocol	Port range	Source
SSH	TCP	22	This IP range should be limited to only clients where you will be SSHing to the OnSight from, such as a Jump Server.
Custom TCP Rule	TCP	8443	This IP range should be limited to the other servers that will be using the OnSight to proxy agent data through (if any).
HTTPS	TCP	443	This IP range should be limited to only clients where you'll be sshing to the OnSight from, such as a Jump Server.



8. Select your key-pair from the dropdown, launch the instance, and then wait for it to boot.

Register the OnSight

1. SSH into your newly created OnSight vCollector instance using the user **ubuntu** and the key-pair you selected in the previous step.
2. Run updates to pick up any recently released OS patches by running `sudo apt update; sudo apt upgrade`.
3. Reboot the server by running `shutdown -r now`
   Register the OnSight by running `sudo onsight register`. You will also need your Customer Key. To obtain your Customer Key, visit the Control Panel and click your avatar then select **My Account**.
   This will add the appropriate repositories to the VM and begin downloading and configuring dependencies. This process should only take 2 - 3 minutes. Once the installation has completed, you will see a message similar to the following:
4. 
```
Last login: Thu Apr  1 15:07:34 2021 from 65.79.141.159
ubuntu@ip-172-31-34-238:~$ sudo onsight register
Processing manifest file in /etc/panopta-onsight-manifest
Please enter your customer key to register your new OnSight: tdm3-ycyr-8gs8-gehw
Please enter name for this OnSight: Docs OnSight
Registering new OnSight
Successfully registered OnSight
Pulling latest vCollector image, version 2021.19.4
Starting OnSight vCollector version: 2021.19.4
Started OnSight collector with key cvhj-iedk-yia3-awis

OnSight can be managed at https://my.panopta.com/report/ApplianceReport?applianc
e_id=5070
```

The OnSight vCollector can be managed from the Control Panel. Follow the link outputted at the bottom of the installation success message.

OnSight for Azure installation

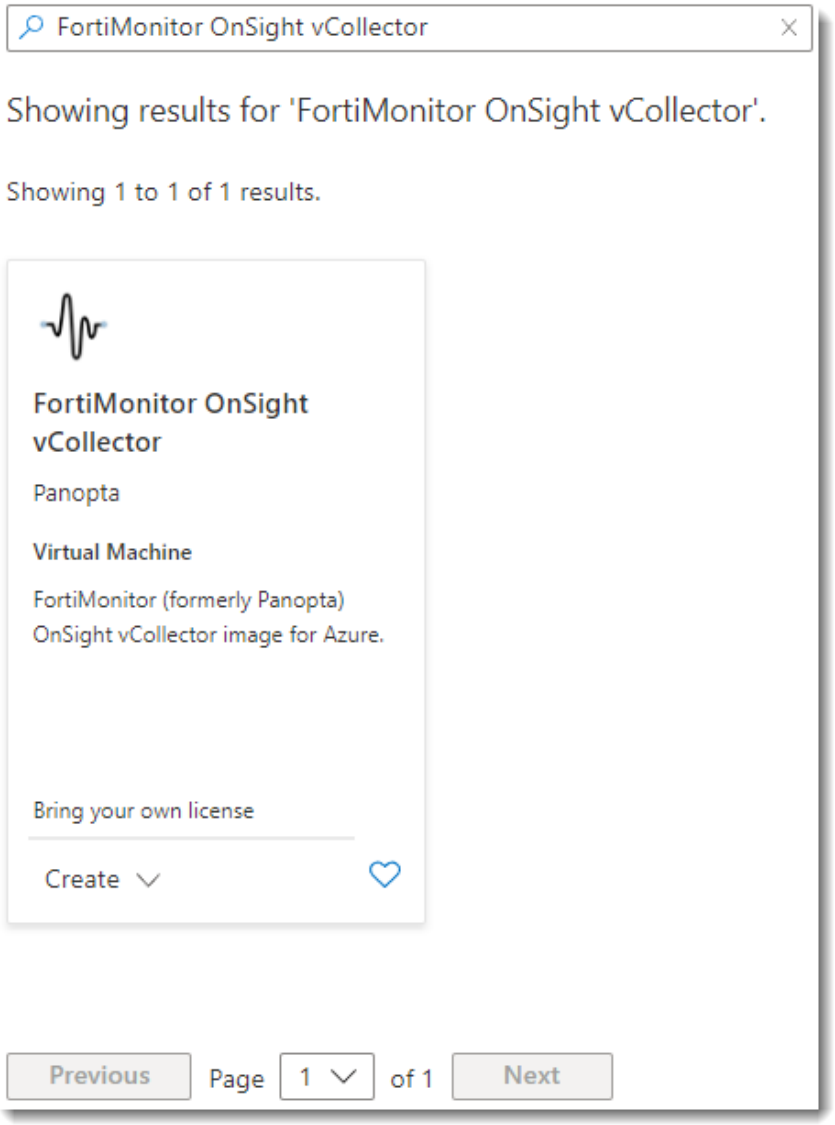
The **FortiMonitor OnSight vCollector for Azure** - available via the Azure Marketplace - allows you to deploy and run an OnSight from an Azure instance. The following sections detail the resource requirements and deployment steps.

Resource requirements

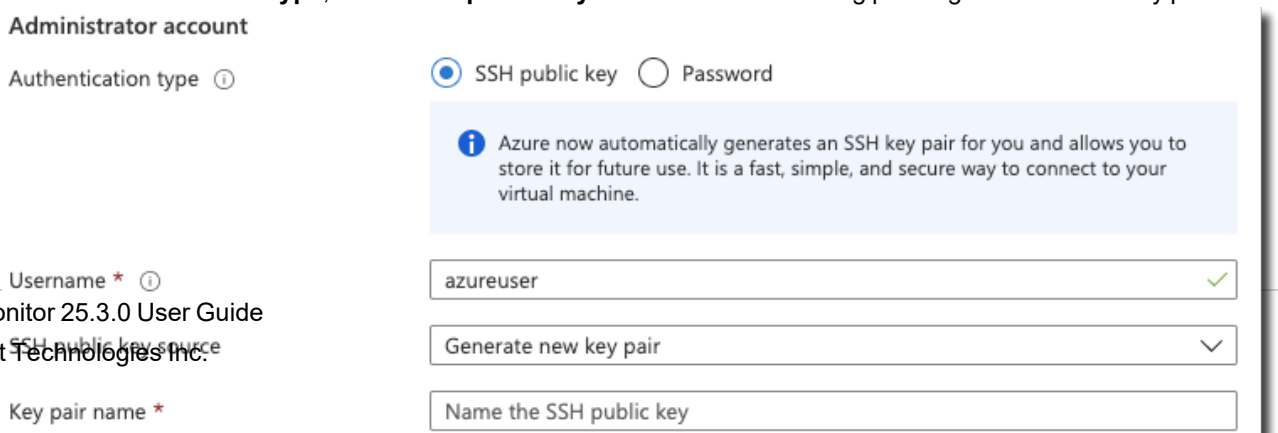
- Resource requirements
- Network connectivity requirements

OnSight installation

1. Within your Azure Portal, navigate to the Azure Marketplace.
   Search for **FortiMonitor OnSight vCollector** then select it from the search results.



2.
3. Click **Create**.
4. From the **Create a virtual machine** page, fill out the fields as needed.
5. Within the **Instance details** category, select a VM size by clicking **Select size**. Note: Make sure to select an appropriate VM size based on the recommended requirements. See Resource requirements.
   For the **Authentication type**, select **SSH public key**. You can use an existing pair or generate a new key pair.



7. Click **Review and Create** and confirm your configuration.
8. Click **Create** to create the OnSight vCollector instance.

Register the OnSight

1. SSH into your newly created OnSight vCollector instance using the key-pair you selected in the previous step.
2. Run updates to pick up any recently released OS patches by running `sudo apt update; sudo apt upgrade`.
3. Reboot the server by running `shutdown -r now`
   Register the OnSight by running `sudo onsight register`. You will also need your Customer Key. To obtain your Customer Key, visit the Control Panel and click your avatar then select **My Account**.
   This will add the appropriate repositories to the VM and begin downloading and configuring dependencies. This process should only take 2 - 3 minutes. Once the installation has completed, you will see a message similar to the following:
4. 

   The OnSight vCollector can be managed from the Control Panel. Follow the link outputted at the bottom of the installation success message.

Automate the FortiMonitor OnSight deployment

Similar to our server Agent, you can use a manifest file to automatically register new OnSight vCollectors with default settings into your FortiMonitor account. This can be added into your existing provisioning and DevOps automation processes.

The manifest file should be **/etc/panopta-onsight-manifest** prior to installation using the OnSight standalone installer or shortly after the VM boots up if you're using the OnSight virtual image installation

OnSight Manifest File Configuration

The format of the file is as follows:

OnSight manifest file

```
[onsight]

name = US East Onsight

customer_key = aaaa-bbbb-cccc-dddd

server_group = Amazon Infrastructure
```

```
appliance_group = 3

alert_timeline = Production Alerts

agent_proxy = enable
```

Customer_Key: This identifies you with your FortiMonitor account. By setting this value, the agent will automatically add the server to the FortiMonitor control panel. You can find this key in the control panel under "My Account" in the "Settings" drop-down.

Server_Group: This controls which group your OnSight gets added to in the control panel. Any server template(s) that you have set to be applied to this group will be applied to this OnSight as well. The value of the server group is identified by an ID. If you go to a server group in the control panel you can see that group's ID number at the end of that pages URL.

Appliance_Group: Optional, in case you choose to put your OnSight Appliances in a cluster or HA setup.

Alert_Timeline: Determines which timeline the OnSight heartbeat alarm will use.

Agent_Proxy: Defines whether or not Server Agents in your account will be able to use the OnSight as a proxy for outbound connectivity.

Disable auto-provisioning

Auto-provisioning can be disabled in the OnSight manifest file by adding **disable_auto_provisioning = true**. For example:

```
[onsight]
customer_key = {YOUR_CUSTOMER_KEY}
aggregator_url = {YOUR_AGGREGATOR_URL}
auto_discovery_rules = lan1,lan2

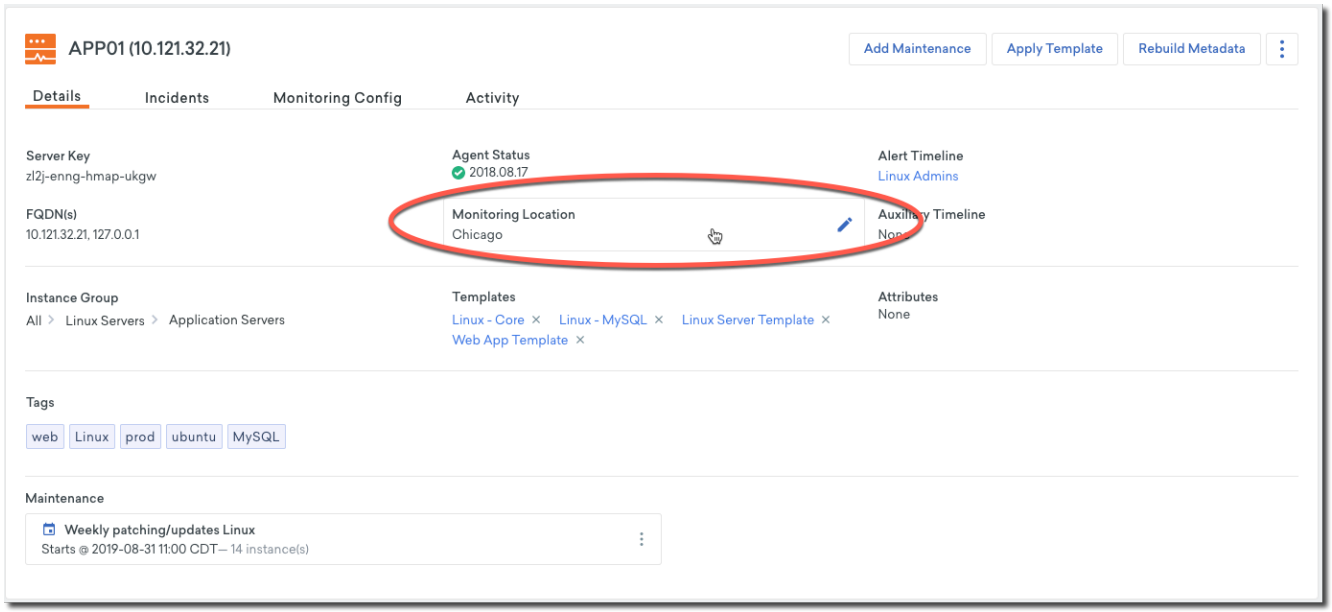

[lan1]
ip_range = 10.10.10.0/29
discover_servers = true
discover_network_devices = true
auto_add = true
disable_auto_provisioning = true

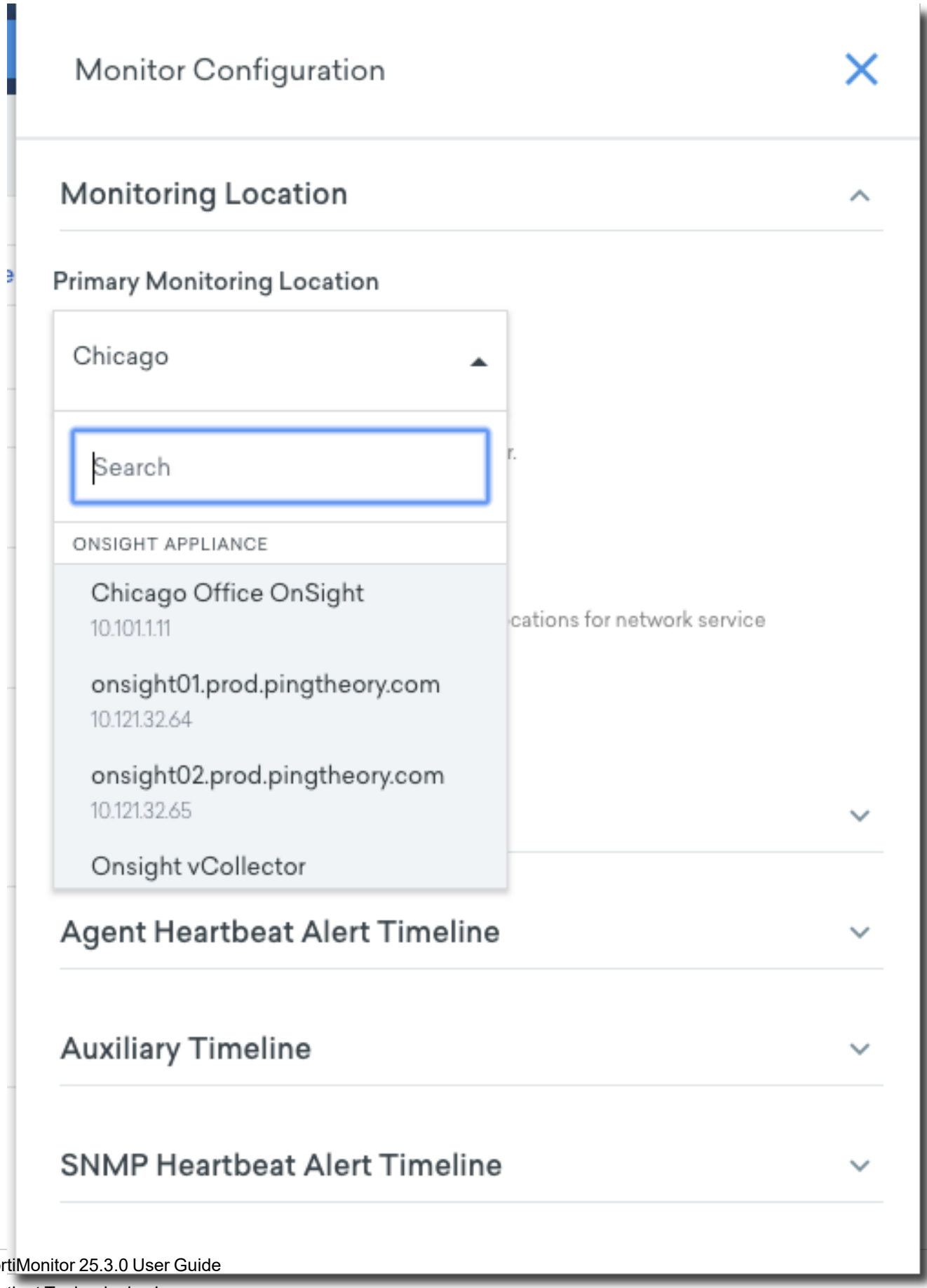
[lan2]
ip_range = 10.100.2.0/29
discover_servers = true
discover_network_devices = true
auto_add = true
disable_auto_provisioning = false
```

Monitor a device that has not been discovered with OnSight

You may find that you would like to add a device to be monitored by OnSight which has not been automatically discovered (for any number of reasons). In this case, you can add the device to your control panel as you would any other server. The only adjustment you will need to make is to change the monitoring location to your OnSight. All of your OnSights will be listed under **OnSight Appliance** in the **Primary Monitoring Location** dropdown of the configuration

drawer.


APP01 (10.121.32.21)
Add Maintenance
Apply Template
Rebuild Metadata
Details
Incidents
Monitoring Config
Activity
Server Key
zl2j-enng-hmap-ukgw
Agent Status
2018.08.17
Alert Timeline
Linux Admins
FQDN(s)
10.121.32.21, 127.0.0.1
Monitoring Location
Chicago
Auxiliary Timeline
Instance Group
All > Linux Servers > Application Servers
Templates
Linux - Core ×
Linux - MySQL ×
Linux Server Template ×
Web App Template ×
Attributes
None
Tags
web
Linux
prod
ubuntu
MySQL
Maintenance
Weekly patching/updates Linux
Starts @ 2019-08-31 11:00 CDT— 14 instance(s)


Monitor Configuration
Monitoring Location
Primary Monitoring Location
Chicago
Search
ONSIGHT APPLIANCE
Chicago Office OnSight
10.101.1.11
onsight01.prod.pingtheory.com
10.121.32.64
onsight02.prod.pingtheory.com
10.121.32.65
Onsight vCollector
cations for network service
Agent Heartbeat Alert Timeline
Auxiliary Timeline
SNMP Heartbeat Alert Timeline

Once you select the correct OnSight, and create the server, you can start setting up network services checks on this device.

OnSight discovery

The OnSight auto-discovery tool allows you to add new instances to your infrastructure based on configurable discovery rule sets. Using these rulesets, you can set which IP ranges to scan for instances, the scan frequency, and the ports to scan.

Add auto-discovery rules

Once your OnSight has been installed, you can add discovery rules by performing the following steps:

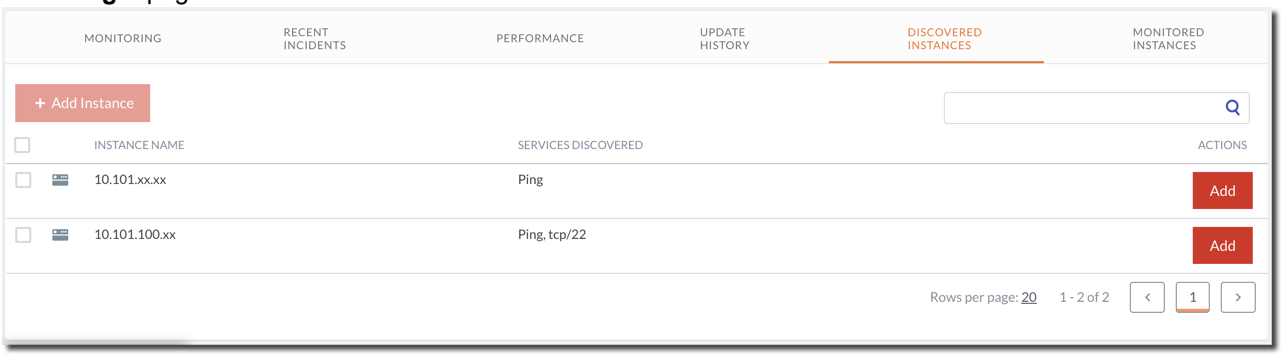
1. From an **OnSight vCollector's** page, select **Edit**.
   The OnSight's configuration drawer will slide out.
2. Expand the **Auto-discovery** module.
   This section allows you to add Discovery Rulesets and choose to whom new server discovery alerts will be sent.

There are a number of variables for server and network device discovery rules.

Field	Description
IP range to scan	The IP range to autodetect on, written in CIDR notation, e.g., 10.100.100.1/24
Re-scan frequency (minutes)	This is how many minutes the automatic discovery will wait before rescanning your network for new servers or changes. Enter 0 for a one-time scan. 10 mins is the lowest frequency allowed.
Ports to Scan	Select the ports to scan. **Note**: Make sure to enter a port either in the **Ports to Scan** field or the **Other ports to scan** field. Failure to do so will result to the port scan failing.
Other ports to scan	A comma-separated list of other ports you'd like to scan. **Note**: Make sure to enter a port either in the **Ports to Scan** field or the **Other ports to scan** field. Failure to do so will result to the port scan failing.
Discover servers	Enable the option to allow the OnSight to discover the servers in your network.
Discover network devices	Enable the option to allow the OnSight to discover network devices in your network. The OnSight will try each of the SNMP credentials you have configured in FortiMonitor when discovering devices and retrieving device information.
Automatically add newly discovered instances	Enable the option to automatically add the discovered instances to FortiMonitor. For network devices, an SNMP discovery will be started to find and provision hardware metrics.

Don't automatically provision metrics	Enable the option to stop automatic provisioning of metrics.
Send notification when new instances are discovered	Enable the option to send a notification when new instances are discovered. Once the option is enabled, you can select one or more users.

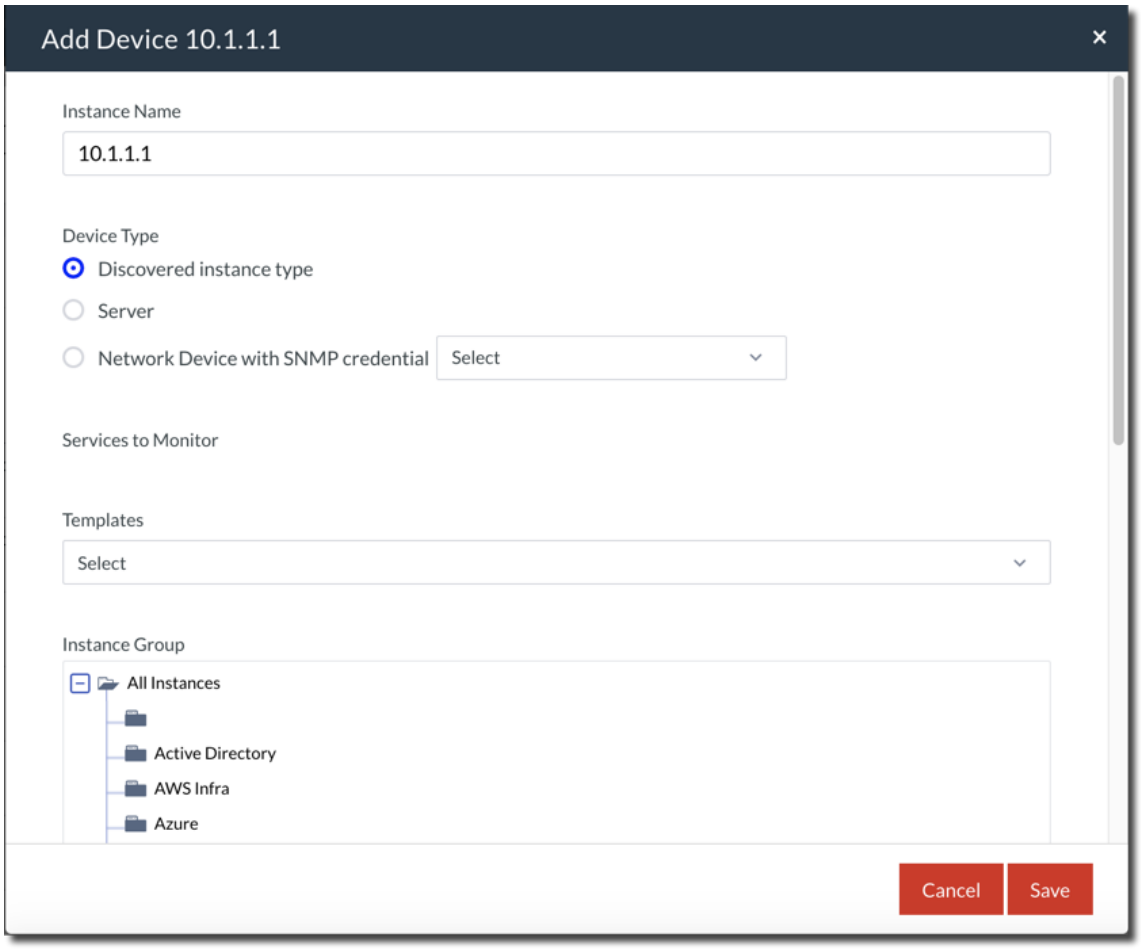
Once the discovery has run, you can find the discovered servers under the **Discovered Instances** tab of the **OnSight** page.



Add discovered instances

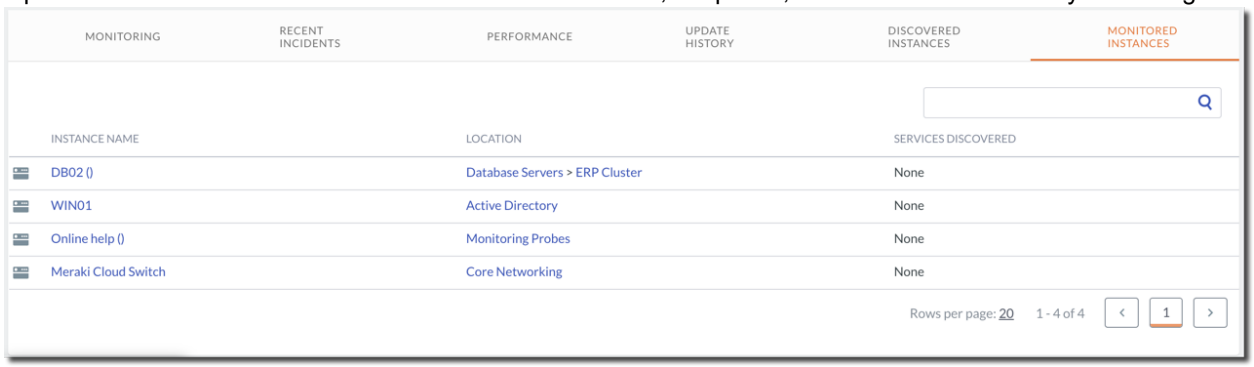
Checking the boxes next to the desired instance and clicking **Add Instance** will automatically add the selected instances to your account. To add multiple instances, select the instances you want to add then click **Add Instance**. In the **Add Device** modal that appears, you can configure the following fields:

Field	Description
Instance Name	Optionally change the name of the discovered instance.
Device Type	Set the device type of the discovered instance. • **Discovered instance type** – Use whichever device type was discovered by the OnSight network discovery. • **Server** – This option sets the device type to **Server** and overwrites the device type discovered by the OnSight network discovery. • **Network Device with SNMP credential** – Sets the device type to Network Device and overwrites the device type discovered by the OnSight network discovery.
Services to Monitor	Shows the services found by the OnSight network discovery.
Templates	Select the template to apply to the instance.
Instance Group	Select the group where you want to add the instance.



View monitored instances

Open the **Monitored Instances** tab to see all of the servers, endpoints, and devices monitored by the OnSight.



Upgrading the OnSight

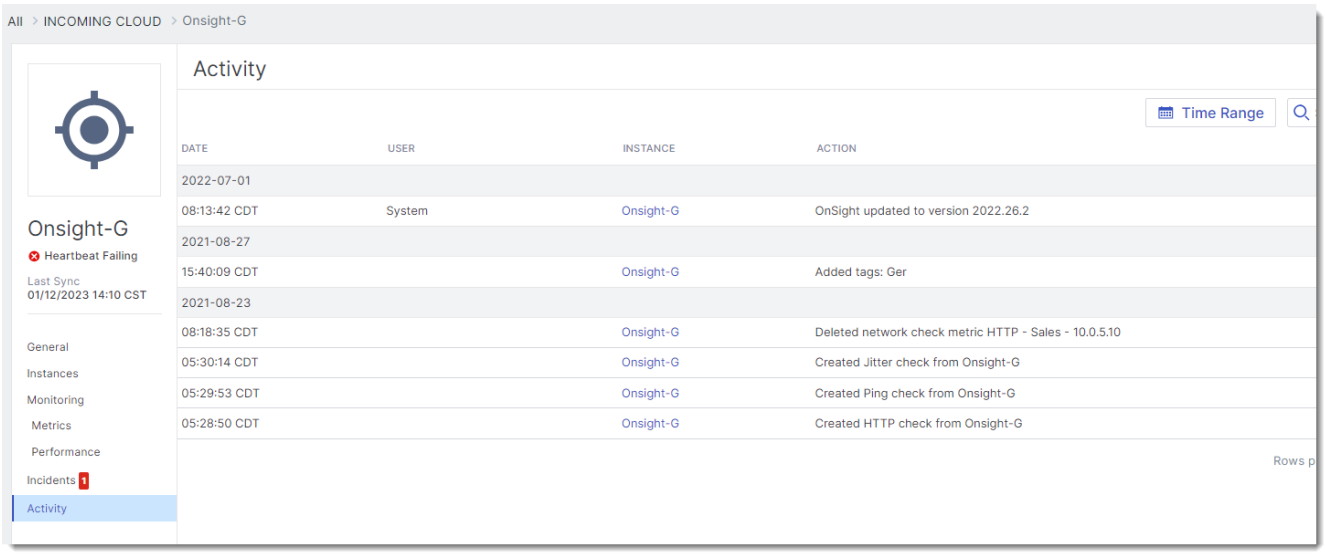
The command to upgrade the OnSight varies depending on the version you have installed. To check the version of your OnSight, go to the OnSight's details page. You can find the OnSight instance by either using **Search** or by going to **Monitoring** > **OnSights**.

Once you determine the version of the OnSight, follow the procedures detailed in the following pages:

- 2020.68 or newer OnSight versions
- 2020.51 to 2020.67
- 2020.50 and lower OnSight versions

Update history

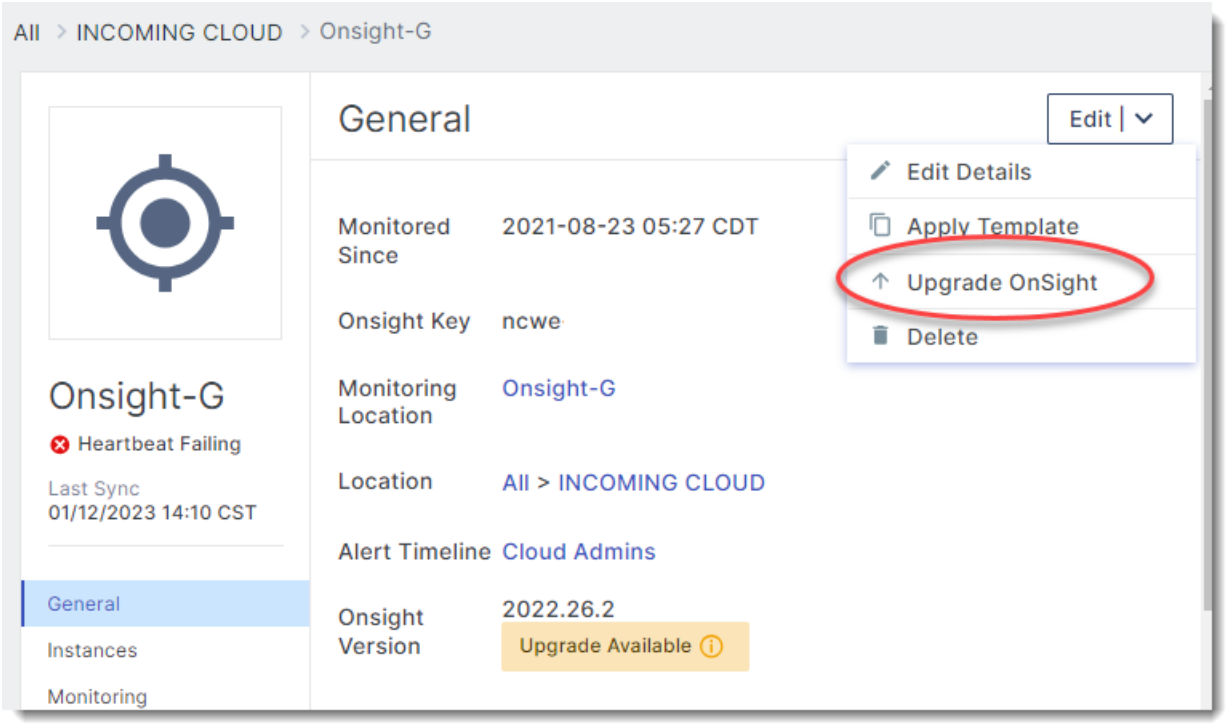
Click the **Activity** tab to see a record of OnSight updates which includes the time of the update, the user, and the OnSight version.



2020.68 or newer OnSight versions

If your OnSight version is 2020.68 or later, perform the following steps to upgrade it:

1. Go to OnSight's page.
2. Click the down arrow.



3. Select **Upgrade OnSight**.

The installer then checks to confirm that there is enough space available, there is no backlog of results waiting to sync, and then initiates the upgrade. The process typically takes between 5 and 10 minutes.

You can also upgrade the OnSight manually by performing the following steps:

1. SSH into your OnSight vCollector.
2. Switch to the root user.
3. Run the following command:

```
onsight upgrade
```

During the upgrade, metric data will also be synced. If you encounter any issues, use the `--force` parameter to bypass the syncing. For example:

```
onsight upgrade --force
```

Note: Using the `--force` option may result to data loss. Perform sanity checks before running the command..

2020.51 to 2020.67

If your OnSight version is from 2020.51 to 2020.67, perform the following steps to upgrade it:

1. SSH into your OnSight vCollector.
2. Switch to the root user.
3. Run the following command:

```
curl -fsSL https://packages.panopta.com/install/onsight/onsight.sh --output /usr/bin/onsight
```

4. If there are no issues found, run the following command:

```
onsight upgrade
```

2020.50 and lower OnSight versions

OnSight vCollector versions 2020.50 and lower can be upgraded using an upgrade script (**onsight-upgrade.sh**) similar to the script used when installing a new OnSight vCollector (see One-liner installer).

- **check-upgrade** - Runs checks to make sure that the OnSight is upgradeable and has sufficient resources.
- **upgrade** - Used to perform the actual upgrade.
- **teardown** - Used to uninstall a previously deployed OnSight after an upgrade.

To upgrade an existing OnSight deployment, perform the following steps:

1. Log in to the VM containing the OnSight vCollector.
2. Run the following command to verify your OnSight deployment:

```
curl -fsSL http://packages.panopta.com/install/onsight/onsight-upgrade.sh > onsight-upgrade.sh
```

```
bash onsight-upgrade.sh check-upgrade
```

3. If there are no issues found, run the following command to update the OnSight:

```
bash onsight-upgrade.sh upgrade
```

4. Once the upgrade is complete, remove the previous OnSight installation by running the following:

```
bash onsight-upgrade.sh teardown
```

Manage and configure your OnSight vCollector

This page is only applicable to OnSight versions **2020.51** and later.

The OnSight comes bundled with a set of CLI commands to manage and configure the OnSight. The commands utilize the following structure:

```
onsight [OPTIONS] COMMAND [ARGS]
```

To see the available commands, run:

```
onsight --help
```

Management and configuration commands

The available OnSight management and configuration commands are described in the following table:

Command	Use this to
check-upgrade	Check to see if a newer vCollector is available
configure-vcollector	`onsight configure-vcollector [OPTIONS]`

	where [OPTIONS] are: • `--enable-agent-proxy` – Enable the Agent proxy in the OnSight. • `--disable-agent-proxy` – Disable the Agent proxy in the OnSight • `--appliance-key [KEY]` – Set the Appliance key to use if already registered. • `--customer-key [KEY]` – Set the Customer key from FortiMonitor account. • `--aggregator-url [URL]` – Set the Aggregator URL to use for syncing. • `--name [NAME]` – Set the name to use when registering the new OnSight. • `--help` – Show this message and exit.
logs	Retrieves the OnSight logs and adds them to a zip file locally. You also have the option to send the logs to our support team. To use this command, run: `onsight logs [OPTIONS]` *where* [OPTIONS] are: • `--upload` – Sends the logs to FortiMonitor Customer Support team. • `--filter TEXT` – Filters the log files by name. You can use regular expressions as a filter and any log with a name that matches the given regex will be added to the zip file. • `--age TEXT` – Filters how many days of logs to be retrieved and added to the zip file. Running the `onsight logs` command without any option will create a zip file of all of the logs in your current directory.
register	Register a new OnSight and generate an appliance key.
shell	Open a shell connection to the collector.
start-vcollector	Start the OnSight vCollector container.
status	Get the current status of the OnSight and its components
uninstall	Uninstall the OnSight.
upgrade	Upgrade to the latest version of the OnSight vCollector.
list-proxy	List proxy server .
set-proxy	Set the proxy server.
configure-ntp	Configure the OnSight to use an internal NTP server. This command is only applicable to OnSight virtual image installations. To use this command, see NTP server configuration .
ncm-uninstall	Uninstalls the NCM. This command is only applicable if you are using NCM.
agent-install	Installs an Agent.
agent-uninstall	Uninstalls the Agent.

OnSight custom plugins

OnSights support two kinds of custom plugins that can be run on instances that are unable to support a FortiMonitor Agent such as network devices and PaaS cloud services.

This feature is only available on the OnSight vCollector.

OnSight custom plugins:

- Custom CounterMeasure action plugins
- Custom metric plugins

Custom CounterMeasure actions and custom metric plugins can also be developed locally using the Local development environment. These plugins can then be uploaded to OnSight to run in a production environment. You need Docker to use the development environment.

Note: Custom plugins are only available on OnSights where they are deployed. For OnSight groups, the custom plugins must be deployed to all OnSights in the group. See Deployment.

CounterMeasure action plugins

OnSight-based CounterMeasures allows CounterMeasures to be run on instances that cannot use the FortiMonitor Agent. OnSight-based CounterMeasures actions are executed from the OnSight and have the ability to target any number of remote targets corresponding to instances monitored from that OnSight. These CounterMeasure actions are primarily intended to target actions and data collection from remote devices and not from the OnSight host itself. The following are a few scenarios where OnSight-based CounterMeasures can be useful.

- Run CounterMeasures on network devices which do not run FortiMonitor Agents
- Trigger API calls to virtualization systems like VCenter, Nutanix, HyperV
- Trigger API calls to cloud providers (AWS, Azure, Google) for PaaS and SaaS services

To create a CounterMeasure action plugin, see Creating a custom CounterMeasure.

Custom metric plugins

You can add your own custom metric plugins to the OnSight to allow you to collect metrics from remote devices.

To create a custom metric plugin, see Creating a custom metric plugin.

Requirements

It is recommended that you upgrade to the latest version of the OnSight vCollector before proceeding. To verify that an OnSight vCollector is of appropriate version to run OnSight custom plugins, run the following:

From Root (using `sudo -i`, for example):

```
onsight check-upgrade
```

Directory Structure

The following outlines the directory structure relevant to using OnSight custom plugins. These directories are located on the host VM that runs the OnSight.

- **/usr/share/onsight/countermeasures**: Directory to put countermeasure plugins
- **/usr/share/onsight/custom-metrics**: Directory to put custom metric plugins
- **/usr/share/onsight/secrets**: Directory for secrets
- **/usr/share/onsight/ssh-keys**: Directory to put ssh private key files

- **/usr/share/onsight/dependencies**: Directory to install python packages required to run plugins. If your plugin requires other python packages, you can install them under **/usr/share/onsight/dependencies/** directory. For example, if your plugin requires pyyaml package, you can install it with this command: `pip install -t /usr/share/onsight/dependencies pyyaml`
- **/var/log/countermeasures**: Directory contains:
  - countermeasure.log
  - custom_metric.log

Creating a custom CounterMeasure

You can create custom CounterMeasure actions using Python. Some example actions that can be leveraged using Python:

- Rebooting
- Authentication
- Interacting with APIs
- Managing state over multiple steps

CounterMeasure workflow

1. Create a custom CounterMeasure.
2. Add CounterMeasure plugins you created to the **/usr/share/onsight/countermeasures** directory. See Deployment.
3. Add a CounterMeasure to a metric threshold, such as **Disk % used**. See Configuring a CounterMeasure.
4. If the threshold is crossed, the CounterMeasure will be run.

Any output provided by the CounterMeasure will be attached to the incident record and will be available in the FortiMonitor control panel.

Create custom CounterMeasure actions

To create a custom CounterMeasure action:

Create a new python file in the **/usr/share/onsight/countermeasures** directory and ensure your implementation subclasses the **CountermeasurePlugin** class.

A full reference is available in the Implementation Reference section. Largely, implementing the **run()** method and providing a few instance variables is all that is required. You can see a basic example in CounterMeasure Action metadata example.

Note: The name of your custom class needs to end with **Countermeasure**.

Example custom CounterMeasure action

The following CounterMeasure plugin allows you to reboot an AWS EC2 instance:

```
from CountermeasurePlugin import CountermeasurePlugin
import boto3
```

```
class AwsEc2RebootCountermeasure(CountermeasurePlugin):
    textkey = "aws.ec2.reboot"
    author = "demo"
    name = "AWS EC2 reboot countermeasure"
    description = "Reboot aws ec2 instance"

def run(self):
    # get server info from metadata
    server = self.metadata.get('server')
    fqdn = server.get('fqdn')

    # get access/secret key from secrent management framework
    aws_access_key_id = self.get_secret('vault-kv', 'aws_access_key')
    aws_secret_access_key = self.get_secret('vault-kv', 'aws_secret_key')

    client = boto3.client('ec2',
                          region_name='us-east-2',
                          aws_access_key_id=aws_access_key_id,
                          aws_secret_access_key=aws_secret_access_key)

    response = client.describe_instances()
    for reservation in response['Reservations']:
        instance = reservation['Instances'][0]

        instance_id = instance['InstanceId']
        public_dns_name = instance['PublicDnsName']
        private_dns_name = instance['PrivateDnsName']
        public_ip = instance['PublicIpAddress']
        private_ip = instance['PrivateIpAddress']

        if fqdn in (public_dns_name, private_dns_name, public_ip, private_ip):
            client.reboot_instances(InstanceIds=[instance_id])
            self.logger.info("AwsEc2RebootCountermeasure: rebooted %s (%s)" % (instance_id,
fqdn))
            self.save_output("rebooted %s (%s)" % (instance_id, fqdn))
            break
```

The following CounterMeasure plugin allows you to reboot a FortiGate:

```
from CountermeasurePlugin import CountermeasurePlugin
import json


class FortiGateRebootCountermeasure(CountermeasurePlugin):
    name = "FortiGate reboot"
    textkey = "fortigate.reboot"

    def run(self):
        server = self.metadata.get("server")
        instance_id = server.get("id")

        data = { "event_log_message": "Reboot by onsight countermeasure" }
        output = self.fortiapi_fortios(instance_id, "/api/v2/monitor/system/os/reboot",
method="POST", data=data)
        self.save_output(json.dumps(output))
```

Creating a custom metric plugin

You can add your own custom metric plugins to the OnSight to allow you to collect metrics from remote devices. If you are just getting started, see Local development environment to develop plugins on your local machine.

1. Create a new python file in **/usr/share/onsight/custom-metrics/** directory in host server.
2. You can start with the **TemplatePlugin.py** file, which defines the overall layout of the plugin class and the methods necessary to operate.
   Note that the class name needs to end in **Plugin** otherwise the plugin will not work, for example **SNMPPlugin**.
3. Define a class attribute **textkey** for your plugin. This is a unique dotted string defining the very general service or resource that your plugin is monitoring. The **textkey** should be prefixed with an identifier for your organization. For example, **com.abc.myapplication**.
4. Implement **get_metadata()**. This is the method that builds a list of information about the metrics that this plugin is able to collect. it should return a list of dictionaries, each with the following keys
   - metric_textkey: unique textkey that identifies this metric type within the plugin
   - name: human-friendly name for the metric, displayed in config UIs, graphs and alerts
   - description: human friendly description of the metric, displayed as supporting content when configuring the metric
   - unit: string which specifies the unit of the metric, or None if unit-less
   - resource_options: a list of options, for example port name, which uniquely identify a specific instance of the metric. Set to an empty list if there is only one instance of the metric.
5. implement **get_data()**. The purpose of this method is to actually measure the thing you want to monitor and return its value. It takes the following arguments
   - textkey: the specific metric type that should be gathered. one of metric_textkey returned from get_metadata().
   - option: the optional specifier for a specific instance of the metric, for example indicating what port on a switch to check. If not specified in resource_options returned from get_metadata(), it will have a value of None
   - instance_id: the global ID of the target instance that aligns with what is present in the control panel or FortiMonitor API
   - hostname: the IP or FQDN of the target instance
   - device_type, device_sub_type: strings which specify the type of the instance
   - tags: list of strings with each tag that's applied to the instance
   - attributes: dictionary of attribute keys and values that have been applied to the instance
6. If your plugin requires other python packages, you can install them under **/usr/share/onsight/dependencies/** directory. For example, if your plugin requires pyyaml package, you can install it with this command. **pip install -t /usr/share/onsight/dependencies pyyaml**
7. You can copy or generate ssh keys under **/usr/share/onsight/ssh-keys/** directory, which can be used with ssh helper functions to SSH to remote devices
8. You can create the config of secret repos under **/usr/share/onsight/secrets/** directory. Currently, FortiMonitor supports Hashicorp Vault for secret repos.

Deployment

CounterMeasures

Add CounterMeasure plugins you created to the **/usr/share/onsight/countermeasures** directory. To create a custom CounterMeasure, see Create a custom CounterMeasure.

Custom metrics

See Creating a custom metric plugin.

Install plugins automatically during OnSight installation

You can also host created plugins centrally and have them installed automatically during OnSight installation, so that you do not have to manually manage them post-install. To do this, you must define the **custom_plugin_url** or **countermeasure_plugin_url** in the OnSight manifest file. The value of this variable can be a compressed archive (.zip, .tar or .tgz) or a single python file (with a .py extension). The hosted plugins will automatically be re-fetched once an hour. The downloaded files are then saved in either:

- **/usr/share/onsight/custom-metrics**
- **/usr/share/onsight/countermeasures**

You can download plugins manually through the OnSight CLI:

onsight download-custom-plugin

```
Usage: onsight download-custom-plugin [OPTIONS]

   Download custom plugins from custom_plugin_url specified in the manifest file. custom_
        plugin_url can be set or changed with --url option. The url can be a compressed
        archive (.zip, .tar or .tgz) or a single python file (with a .py extension).
Options:
   --url TEXT
   --help Show this message and exit.
```

onsight download-countermeasure-plugin

```
Usage: onsight download-custom-plugin [OPTIONS]

   Download custom countermeasure plugins from custom_countermeasure_url specified in the
        manifest file. custom_countermeasure_url can be set or changed with --url option.
        The url can be a compressed archive (.zip, .tar or .tgz) or a single python file
        (with a .py extension).


Options:
   --url TEXT
   --help Show this message and exit.
```

Authentication

Password authentication

def execute_ssh_with_password(self, cmd, hostname, username, password)

SSH to a remote server and run a command. password must be specified as a parameter:

```
def get_data(self, textkey, option, instance_id, hostname, device_type, device_sub_type,
     tags, attributes):
   return_code, stdout_output, stderr_output = self.execute_ssh_with_password("some
        command", hostname, "username", "password")
```

def execute_ssh_with_secret_password(self, cmd, hostname, username, repo="local")

SSH to a remote server and run a command. password will be retrieved from the local repo based on username and hostname:

```
def get_data(self, textkey, option, instance_id, hostname, device_type, device_sub_type,
     tags, attributes):
   return_code, stdout_output, stderr_output = self.self.execute_ssh_with_secret_password(
        "some command", hostname, "username")
```

Secret management framework

You can also securely store your passwords encrypted in local storage. For more information, see Secret management framework.

Secret management framework

Secrets management is the practice of securely storing and handling sensitive information, including certificates, SSH keys, API keys, credentials, tokens, passwords, and other confidential data.

The Secret Management Framework provides multiple methods for ensuring secure connections when utilizing OnSight Countermeasures. These methods are described below - some are easier to configure but others are more scalable for larger deployments.

Method	Recommendation
Local SSH Key-pair access method	This is most recommended for small scale testing and deployment. Does not scale well for large deployment.
Vault method	This is recommended for scalable, widespread deployment of OnSight Countermeasures.

Note: For the vault method, only Hashicorp vault is supported. Instances of "vault" in this article refers to **Hashicorp vault**.

Configuration

Directories

The directories listed below reside on the host VM and not within the individual OnSight container.

- **/usr/share/onsight/countermeasures**: Directory to put countermeasure plugins
- **/usr/share/onsight/custom-metrics**: Directory to put custom metric plugins
- **/usr/share/onsight/secrets**: Directory for secrets
- **/usr/share/onsight/ssh-keys**: Directory to put ssh private key files
- **/usr/share/onsight/dependencies**: Directory to install python packages required to run plugins
- **/var/log/countermeasures**: Directory contains:
  - countermeasure.log
  - custom_metric.log

Secret management for password authentication

You can securely store your passwords encrypted in local storage. The stored passwords will be used in execute_ssh_with_secret_password(). If you omit --host, the password is used for a random host. If both a password with host and a password without host exist, the host-specified password is used

```
Usage: onsight set-secret [OPTIONS]
   Set secret for countermeasure and custom plugins in local repo. Password can be set for
        user or user@host. Password will be removed if --password is missing or empty
        string.
Options:
   --user TEXT [required]
   --host TEXT
   --password TEXT
   --repo TEXT [default: local]
   --help Show this message and exit.
```

You can view the stored passwords with the **onsight list-secret** command. The passwords will be displayed as encrypted.

SSH key authentication

def execute_ssh(self, cmd, hostname, username, private_key_file, disable_sha2_pubkeys=False)

SSH to a remote server and run a command, returning the command output. private_key_file should be placed under **/usr/share/onsight/ssh-keys/** directory in host. public key file must be registerd on the target server

def execute_ssh_with_secret_pkey(self, cmd, hostname, username, repo, key, disable_sha2_pubkeys=False)

private_key will be retrieved from the secret repo (currently, only Hashicorp vault is supported).

SSH Cert authentication

def execute_ssh_with_secret_cert(self, cmd, hostname, username, repo)

SSH cert will be retrieved from the secret repo (currently, only Hashicorp vault is supported.

Secret Management for Hashicorp Vault

You can add SSH keys directly to your OnSight VM in the directory outlined above, but a more scalable solution exists in the form of secret management vaults. Currently, FortiMonitor supports HashiCorp Vault.

You will need to utilize the Vault URL, token, path, mount point, and version as shown in the example below.

Secret repository configuration

- Create a repository directory under **/usr/share/onsight/countermeasures/secrets** and insert the **config.json** file in it (use the repository name for the directory name).

Hashicorp Vault Key-Value engine example:

- Sample **config.json** (**/usr/share/onsight/secrets/vault-kv/config.json**)

```
{
        "type": "hashicorp.vault.kv",
        "url": "Your-Vault-URL,
        "token": "Your-Vault-Token",
        "path": "Your-Vault-Path",
        "mount_point": "Your-Vault-MountPoint/Top-Level-Directory",
        "kv_version": "v1"
}
```

Hashicorp Vault SSH engine example:

- Sample **config.json** (**/usr/share/onsight/secrets/vault-ssh/config.json**)

```
{
        "type": "hashicorp.vault.ssh",
        "url": "Your-Vault-URL,
        "token": "Your-Vault-Token",
        "path": "Your-Vault-Path",
        "mount_point": "Your-Vault-MountPoint/Top-Level-Directory",
        "api_version": "v1"
}
```

- The CA public key must be added to target host's SSH configuration. Add the public key to all target host's SSH configuration.
  - Download the CA public key.
    `curl -o /etc/ssh/trusted-user-ca-keys.pem http://URL:PORT/v1/ssh-client-signer/public_key`
  - Edit SSH configuration.

    ```
    # /etc/ssh/sshd_config
    # ...
    TrustedUserCAKeys /etc/ssh/trusted-user-ca-keys.pem
    ```

  - Restart the SSH service to pick up the changes.

Secret management helper functions for CounterMeasure plugins

The following functions have been pre-defined to assist in developing custom CounterMeasures.

```
def get_secret(self, repo, key=None, **kwargs):
       """
       GET secret from Secret management framework
       """
def get_ssh_certificate(self, repo, username, **kwargs):
       """
       GET ssh certificate from Secret management framework
       """


def execute_ssh_with_secret_pkey(self, cmd, hostname, username, repo, key):
       """
       Use Spur/Paramiko to ssh to a remote server and run a command.
       Get private key value in secret repo with a given key
       Returns a tuple of (return code, stdout, stderr)
       """


def execute_ssh_with_secret_cert(self, cmd, hostname, username, repo):
       """
       Use Spur/Paramiko to ssh to a remote server and run a command.
       Get ssh cert file from secret repo
       Returns a tuple of (return code, stdout, stderr)
       """
```

SSH access scenarios

Signed SSH certificate

The instructions below include details for each aspect of this setup.

- Configure the Vault
  - Set up Hashicorp Vault server and create SSH engine.
  - Configure CA for signing and create a named Vault role for signing client keys.
  - Signed SSH Certificates
- On the remote host
  - Add the public key to all target host's SSH configuration.
  - Download CA public key.
    `curl -o /etc/ssh/trusted-user-ca-keys.pem http://URL:PORT/v1/ssh-client-`

```
signer/public_key
```

- Edit ssh configuration

```
# /etc/ssh/sshd_config
# ...
TrustedUserCAKeys /etc/ssh/trusted-user-ca-keys.pem
```

- Restart the SSH service to pick up the changes.

- On the OnSight
  - Create **config.json** under **/secrets/vault-ssh**

```
{
"type": "hashicorp.vault.ssh",
"url": "Your-Vault-URL,
"token": "Your-Vault-Token",
"role": "ssh-client-signer/sign/my-role",
"api_version": "v1"
}
```

- Within the CounterMeasure plugin
  - Use **execute_ssh_with_secret_cert()** method to execute command on a remote host

```
class VaultDemoCountermeasure(CountermeasurePlugin):
name = "Vault demo countermeasure"
   textkey = "testing.vault"


def run(self):
   server = self.metadata.get('server')
   hostname = server.get('fqdn')
   return_code, output, _ = self.execute_ssh_with_secret_cert("pwd", hostname,
        "vadmin", 'vault-ssh')
   self.save_output(output)
```

SSH key-pair access

- Store SSH private key in Vault KV engine and access remote Host with private key.
- Configure the Vault.
  - Set up Hashicorp Vault server and create KV engine.
  - Save private key in KV engine
    - For example: `key="mykey", value="-----BEGIN RSA PRIVATE KEY-----…"`
- On the remote host
  - Copy the public key to the remote host.
- On the OnSight
  - Create **config.json** under **/secrets/vault-kv**.

```
{
   "type": "hashicorp.vault.kv",
   "url": "YOUR-VAULT-URL:PORT",
   "token": "YOUR-VAULT-TOKEN",
   "kv_version": "v1",
   "mount_point": "Your-Vault-MountPoint/Top-Level-Directory",
   "path": "Your-Vault-Path",


}
```

- Within the CounterMeasure plugin
  - Use **execute_ssh_with_secret_pkey()** method to execute command on the remote host.

```
class VaultDemoCountermeasure(CountermeasurePlugin):
   name = "Vault demo countermeasure"
   textkey = "testing.vault"

   def run(self):
      server = self.metadata.get('server')
      hostname = server.get('fqdn')
      return_code, output, _ = self.execute_ssh_with_secret_pkey("pwd", hostname,
           "myuser", 'vault-kv', 'mykey')
      self.save_output(output)
```

  - Sse **get_secret()** method to get private key, if you want to use your own code to access to a remote host.

```
class VaultDemoCountermeasure(CountermeasurePlugin):
   name = "Vault demo countermeasure"
   textkey = "testing.vault"

   def run(self):
      server = self.metadata.get('server')
      hostname = server.get('fqdn')
      pkey = self.get_key('vault-kv', 'mykey')
      # access to a remote host with pkey
```

Fabric extension

Fabric extension must be set up if you want to create custom countermeasure actions and custom metric plugins for Fortinet devices.

If you want to build and test custom plugins which use Fabric Helper Functions to access FortiGate, you should use API token authentication in onsight-dev regardless of how you actually integrate them in the prod environment.

First, get API token from the FortiGate GUI (System > Administrators > Create REST API Admin)

Add **fortiapi-token** variable to metadata

```
"fortiapi-token": "csHkh94cr0QQf8tb9px5N80w34fh11",
```

Optionally, you may need to add **fortiapi-ip** and **fortiapi-port** to specify IP/Port of the target FortiGate. you may need these when you run onsight-dev from an external network. First, check the connectivity to the target FortiGate.

```
"fortiapi-ip": "35.232.85.200",
"fortiapi-port": 443,
```

In your plugins, fabric helper functions will work in the same way regardless of integration methods. For example,

```
self.fortiapi_fortios(
   instance_id, "/api/v2/monitor/system/resource/usage?resource=mem&interval=1-min")
```

FortiOS API

The following function calls FortiOS API on a FortiGate that is integrated with FortiMonitor. This function can be used for Fabric integrations using the Fabric tunnel, Fabric OnSight proxy, or for Fabric 6.x.

def fortiapi_fortios(self, instance_id, api_path, method="GET", data=None)

- instance_id: The instance ID of the target FortiGate. You can use the instance_id passed to get_data() function.
- api_path: FortiOS API path
- method: one of "GET", "POST", "PUT", "DELETE"
- data: The payload

Example of getting the memory usage of FortiGate:

```
def get_data(self, textkey, option, instance_id, hostname, device_type, device_sub_type,
      tags, attributes):
   response = self.fortiapi_fortios(instance_
        id,"/api/v2/monitor/system/resource/usage?resource=mem&interval=1-min")
   return response["results"]["mem"][0]["current"]
```

FortiManager API

This function calls FortiManager API on FortiManager that is integrated with FortiMonitor. This function can be used for Fabric integrations using the FortiManager proxy.

def fortiapi_fortimanager(self, instance_id, api_path, method="get", data=None)

- instance_id: The instance ID of the target FortiManager. you can use the instance_id passed to get_data() function
- api_path: FortiManager API path
- method: one of "GET", "ADD", "SET", "UPDATE", "DELETE", "MOVE", "CLONE", "EXEC"
- data: payload

Example of getting the memory usage of FortiManager

```
def get_data(self, textkey, option, instance_id, hostname, device_type, device_sub_type,
      tags, attributes):
   result = self.fortiapi_fortimanager( instance_id, "/cli/global/system/performance" )
   memory = result[0]["data"]["Memory"]
   return float(memory["Used"].split()[-1].split("%")[0])
```

SNMP extension

If you want to build and test custom plugins which send SNMP queries, you should add variables to the metadata for SNMP access to the target device.

def snmp_get(self, hostname: str, oid: str) -> int | str | None

Get a single value for oid on the SNMP device that is integrated into FortiMonitor.

- hostname: hostname of target device. you can use the hostname passed to get_data() function
- oid: object identifier to query

```
def get_data(self, textkey, option, instance_id, hostname,
             device_type, device_sub_type, tags, attributes):
    value = self.snmp_get(hostname, "1.3.6.1.4.1.12356.101.4.1.4.0")
    return value
```

def snmp_get_multi(self, hostname: str, oids: list[str]) -> dict[str, int | str]

Get multiple value for a list of oids on the SNMP device that is integrated into FortiMonitor.

- hostname: hostname of target device. you can use the hostname passed to get_data() function
- oids: list of object identifiers to query

```
def get_data(self, textkey, option, instance_id, hostname,
             device_type, device_sub_type, tags, attributes):
    result = self.snmp_get_multi(
        hostname,
        ["1.3.6.1.4.1.12356.101.4.1.4.0", "1.3.6.1.4.1.12356.101.4.1.8.0"]
    )
    return sum(v for v in result.values() if type(v) == int)
```

def snmp_walk(self, hostname: str, oid: str) -> dict[str, int | str]

Get multiple value for oid and the subtree of oid on the SNMP device that is integrated into FortiMonitor.

- hostname: hostname of target device. you can use the hostname passed to get_data() function
- oid: object identifier to walk

```
def get_data(self, textkey, option, instance_id, hostname,
             device_type, device_sub_type, tags, attributes):
    result = self.snmp_walk(hostname, "1.3.6.1.4.1.12356.101.4.1")
    return sum(v for v in result.values() if type(v) == int)
```

Local development environment

A development environment is available to create and test custom plugins before deploying directly to an OnSight vCollector. This may be run within any environment supporting Python and does not need to be installed to an OnSight.

onsight-dev.sh provides the local development environment for both custom plugins and countermeasure plugins. you can build and test your custom plugins and countermeasure plugins locally.

Install

1. Setup a recent Linux (Ubuntu or RHEL) machine and install Docker and **jq**
2. Download onsight-dev.sh.

Environment setup

1. Set environment variables based on the build you are running:
   - CM_DEV_BASE (optional, default = current directory)
2. plugins
   - copy custom plugins to **{CM_DEV_BASE}/onsight/custom-metrics**
   - copy countermeasures plugins to **{CM_DEV_BASE}/onsight/countermeasures**
3. packages

- install the python packages required to run a plugin in **{CM_DEV_BASE}/onsight/dependencies** directory

```
pip install -t onsight/dependencies pyyaml
```

4. ssh-keys
   - copy (or generate) ssh private key in **{CM_DEV_BASE}/onsight/ssh-keys** directory
5. secrets
   - create secret repo configuration in **{CM_DEV_BASE}/onsight/secrets** directory

Usage

```
$ ./onsight-dev.sh --help
Usage: onsight-dev [OPTIONS] COMMAND [ARGS]...

Options:
       --help  Show this message and exit.

Commands:
       countermeasure  run countermeasure
       custom-metric   run custom metric plugin
       dump            print metadata via API
       list-secret     List secrets for countermeasure and custom plugins
       set-secret      Set secret for countermeasure and custom plugins
```

Run Test

Download the metadata of target instance from FortiMonitor API

You can get the api-key from https://fortimonitor.forticloud.com/report/ApiKeyManagement.

(Specify --api-host if you run the API locally)

```
$ ./onsight-dev.sh dump \
       --api-key=6336a1b3-b89d-442e-ae13-3fdf3e8ce2bc \
       --instance-id=2486 > 2486.metadata
```

Sample metadata:

```
$ cat 2486.metadata
{
       "instance_id": 2486,
       "hostname": "144.202.55.142",
       "device_type": "server",
       "device_sub_type": null,
       "tags": [],
```

```
        "attributes": {
                "server.origin": "manual",
                "server.os": "Linux"
        }
}
```

Execute custom plugin with metadata:

1. specify --textkey and --metric-textkey to run
2. specify metadata that will be passed to get_data() function

```
$ ./onsight-dev.sh custom-metric \
        --textkey com.fortinet.cisco \
        --metric-textkey power_supply_status \
        --metadata 2486.metadata
```

3. You can omit **--textkey** and **--metric-textkey** by adding **textkey** and **metric_textkey** into the metadata file.

```
$ cat 2486.metadata
{
        "textkey": "com.fortinet.cisco",
        "metric_textkey": "power_supply_status",
        "instance_id": 2486,
        "hostname": "144.202.55.142",
        "device_type": "server",
        "device_sub_type": null,
        "tags": [],
        "attributes": {
        "server.origin": "manual",
        "server.os": "Linux"
        }
}
$ ./onsight-dev.sh custom-metric --metadata 2486.metadata
```

4. You can execute custom plugin directly with metadata retrieved from the API.
   (Specify --api-host if you run API locally)

```
./onsight-dev.sh custom-metric \
        --textkey com.fortinet.cisco \
        --metric-textkey power_supply_status \
        --api-key=6336a1b3-b89d-442e-ae13-3fdf3e8ce2bc \
        --instance-id=2486
```

5. logs are stored under **{CM_DEV_BASE}/onsight/logs/ directory**.

Custom metric caching

Custom metric caching solves the challenge of monitoring device data used in custom plugins that may be redundant with data already polled. In these cases, it is more efficient to cache the results where a plugin can reference them.

Example custom metric plugin

CacheTestPlugin.py

```
from CustomMetricPlugin import CustomMetricPlugin
import time


class CacheTestPlugin(CustomMetricPlugin):

    # This textkey identifies the plugin, not individual metric types.  It should be
    # of the form "com.example.sample" which uniquely identifies your organization and
    # the purpose of the plugin
    textkey = "com.cache.test"

    # This is the name of the plugin, which will be used when displaying metric
    # options supported by the plugin.
    name = "Cache Test Plugin"

    def get_metadata(self):
        return [
            {
                "metric_textkey": "test",
                "name": "test",
                "resource_options": None,
                "resource_option_type": "none",
                "description": "test function",
            },
        ]

    def get_data(
        self,
        textkey,
        option,
        instance_id,
        hostname,
        device_type,
        device_sub_type,
        tags,
        attributes,
    ):

        if textkey == "test":
            cached = self.get_cache("test")
            if cached is not None:
                self.logger.info(f"cache found: {cached}")
                return cached
            else:
                self.logger.info(f"cache not found")
                value = int(time.time()) % 86400
```

```
                self.set_cache("test", value, ttl=60)
                return value
```

Cache functions

```
    def set_cache(self, key, value, ttl=3600):
        """
        Store the value of the given key in the cache
        ttl is time to live in seconds (default 1 hour)
        """
        PluginCache.set_cache(key, value, ttl)

    def get_cache(self, key, default=None):
        """
        Get a value for the given key from the cache
        default is the value to return if the key is not found. If not provided, None is
returned.
        """
        return PluginCache.get_cache(key, default)
```

TemplatePlugin.py

```
from CustomMetricPlugin import CustomMetricPlugin



class TemplatePlugin(CustomMetricPlugin):

       # This textkey identifies the plugin, not individual metric types.  It should be
       # of the form "com.example.sample" which uniquely identifies your organization and
       # the purpose of the plugin
       textkey = "TBD"

       # This is the name of the plugin, which will be used when displaying metric
       # options suppoorted by the plugin.
       name = "TBD"



       def get_metadata(self):
              """
              Builds a list of information about the metrics that this plugin is able to
collect.
              Returns a list of dictionaries, each with the following keys:
```

```
            - metric_textkey - unique textkey that identifies this metric type within the
plugin
            - name - human-friendly name for the metric, displayed in config UIs, graphs
and alerts
            - description - human friendly description of the metric, displayed as
supporting content when configuring the metric
            - unit - string which specifies the unit of the metric, or None if unit-less
            - resource_options: a list of options or None
            """
            return [
                  {
                  "metric_textkey": "power_supply_status",
                  "name": "Power Supply Status",
                  "description": "Status of power supply",
                  "unit": "boolean",
                  "resource_options": None,
                  },
            ]



def get_data(self, textkey, option, instance_id, hostname, device_type, device_sub_type,
tags, attributes):
            """
            Gather data for a specific metric specified by the textkey argument.


            textkey identifies the specific metric type that should be gathered.


            option is the (optional) specifier for a specific instance of the metric, for
example indicating what port on a switch to check.  If not specified in the central
configuration it will have a value of None


            Passes the following items that can be used by the plugin:
            - instance_id - the global ID of the target instance that aligns with what is
present in the control panel or FortiMonitor API
            - hostname - the IP or FQDN of the target instance
            - device_type, device_sub_type - strings which specify the type of the
instance
            - tags - list of strings with each tag that's applied to the instance
```

```
            - attributes - dictionary of attribute keys and values that have been applied
to the instance


            Returns either a floating point metric value or None if no value is available
            """
            return None
```

Additional setup and helpful tips

FortiMonitor OnSight allow you to monitor servers from behind a firewall. In effect, OnSight acts as a monitoring node that you install behind your firewall, which can securely communicate back to our servers. OnSight can be set up to automatically discover all of the servers behind your firewall, and then add these to your FortiMonitor control panel so that you can perform network service checks just as you would with any of your public facing server. Once you install your OnSight you can use the web console to update and track your OnSight's perfomance.

Access the OnSight web console

Note: This function has been deprecated from OnSight version r2020.70 onwards. Please refer to Manage and configure your OnSight vCollector for newer instructions.

Your OnSight web console allows you to reboot/shutdown, update, and view logs associated with your OnSight.

To access the web console:

1. Navigate to the OnSight's **Instance Details** page and select the **IP Address**.
2. Enter the following credentials:
   - Username: **admin**
   - Password: `<OnSight key>`
     Successfully logging in will open the **OnSight Console**.

On this page, you can check for upgrades, download logs, reboot or shutdown your OnSight. You can also enable your OnSight to become an Agent proxy. This is useful for servers which are behind your firewall and have no outbound public internet access. Instead, you can install the Panopta agent on those servers and set the aggregator URL in their config files to your local OnSight.

Set up FortiMonitor OnSight in HA or for failover

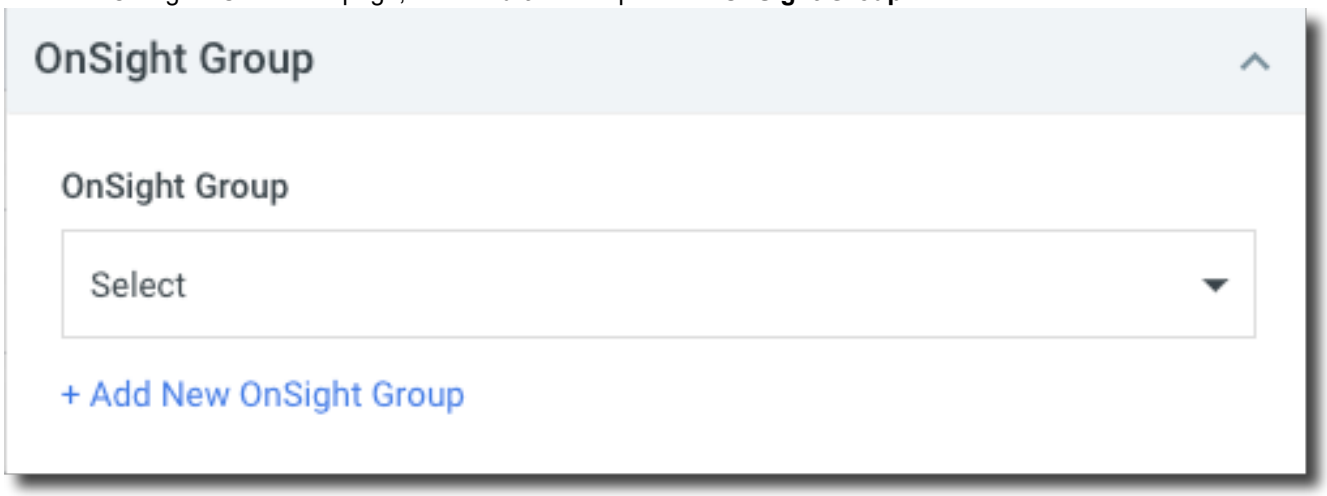
High Availability and Failover Support

If you run a business critical IT infrastructure, you need a monitoring system that you can always rely on to be available and online. Our SaaS cloud services and monitoring nodes are designed to withstand individual node failure with automatic failover to provide an always-on experience. If you are using the FortiMonitor OnSight vCollector, you can

achieve this same level of high availability and fault tolerance. OnSight vCollectors can be entered into a cluster so that checks are automatically failed over to another OnSight node in the event of a failure. If you have server agents proxying through your OnSight appliance to reach the external network, you can make the server agents aware of the multiple nodes.

In order to utilize the high availability and failover features for the OnSight, you would have to add them to an OnSight group. You can do this by first selecting the OnSight you want to add into that group from the control panel.

From the OnSight vCollector's page, select **Edit** then expand the **OnSight Group** module.



Enter a name for the OnSight group. When you do this to two or more appliances, they will automatically use failover for any network checks they are running. This would mean that if an OnSight were set as a monitoring location for a server and it went down, the second OnSight would then take over any network checks that were running on that server if they are within the same OnSight group. Note that load balancing turns on by default when you put an OnSight group in place.

You can also use our high availability features for agent checks if you have the OnSight being run as an agent proxy. Instructions to do that can be found here.

Use OnSight as a proxy for the FortiMonitor Agent

Monitoring your internal infrastructure when it does not have outbound access can be problematic. You can use an OnSight vCollector instance as a proxy, centralizing the flow of data and reducing outband access to a single instance.

Set OnSight as a proxy during Agent installation

During Agent installation, you have the option to set up the OnSight as a proxy for the FortiMonitor Agent using the Agent manifest file.

1. Create an Agent manifest file. To create the file, see the following sections:
   - Windows PowerShell or MSI
   - Linux
2. Edit the manifest file and set the **aggregator_url** parameter to point to the URL or IP address of your OnSight or OnSights.
   If you replace the aggregator URL value within the Agent configuration file with the OnSight Agent Proxy URL, all Agent communication will flow through the proxy. You can also place multiple URLs should you have more than one OnSight. This introduces high availability to your internal monitoring to ensure that you are always receiving the Agent metric data, even if one of your OnSight instances is not responding.

Using Multiple Aggregator URLs
In most mission critical environments, it is highly recommended that you deploy multiple OnSights for a high availability pair. You can also specify each OnSight as an aggregator endpoint in your agent's config file. It is a best practice to use DNS with multiple A records in order to make changes centrally without having to visit each agent.

Linux

For Linux, this file is located in **/etc/fm-agent/** with the following content:

```
[AgentConfig]

version = 2017.03.14

server_key = ****-****-****-****

aggregator_url = https://10.121.32.25:8443, https://10.121.32.26:8443
```

Windows

For Windows, the file is located in **C:\Program Files (x86)\FortimonitorAgent\Agent.cfg** and the relevant section of the configuration is shown below:

```
<?xml version="1.0" encoding="utf-8"?>

<agent>

<service>

<add key="AggregatorUrl" value="https://10.121.32.25:8443, https://10.121.32.26:8443"
/>

<add key="ServerKey" value="****-****-****-****" />

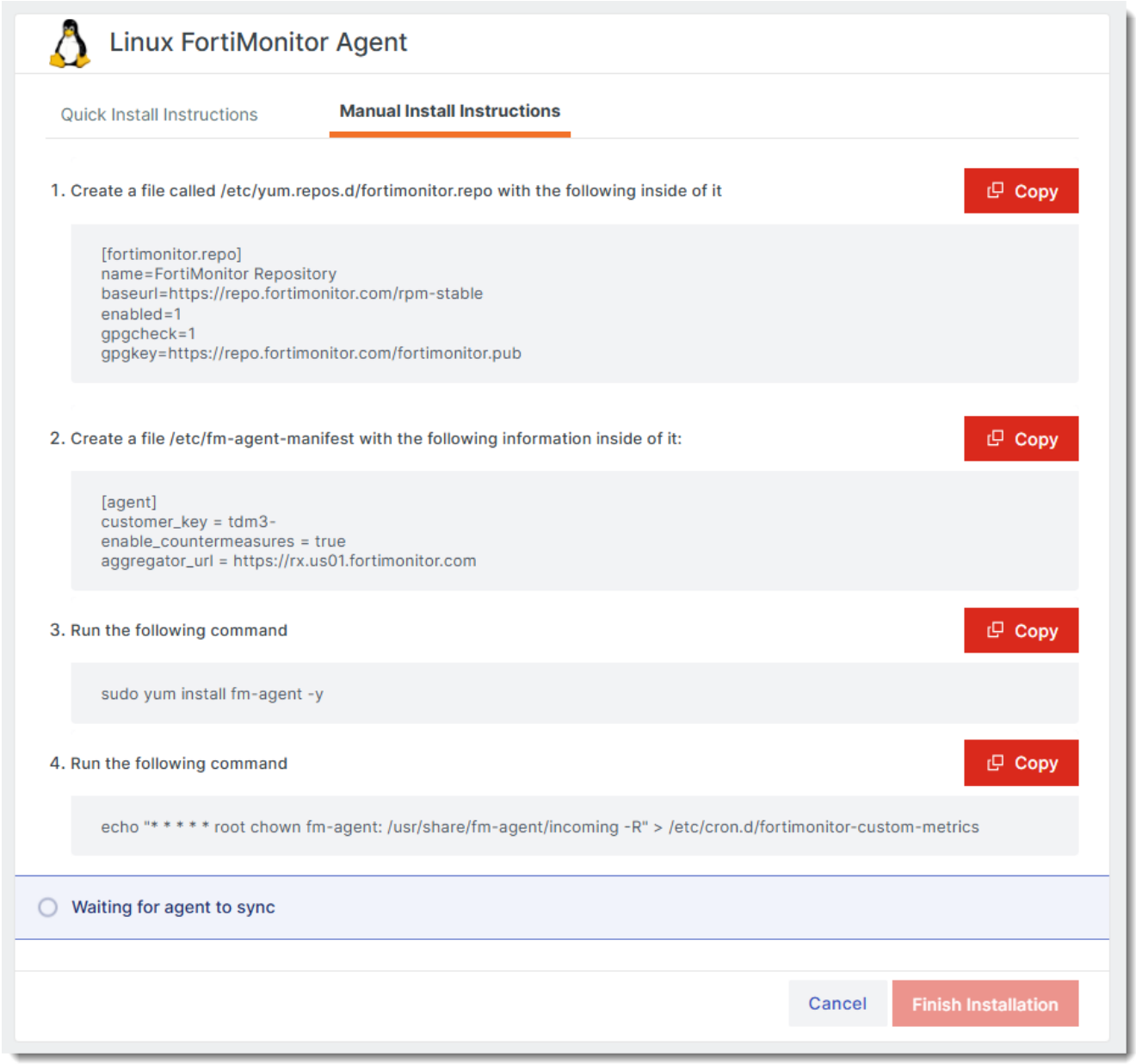
</service>

</agent>
```

Note: If you are using the Windows version of the Agent, you will have to restart the service from within the services menu before seeing any configuration changes take place.

3. Save and close the file.

4. Use the **Manual Install** method to install and add the OnSight proxy to FortiMonitor. For **step 2** of the **Manual Install Instructions** (shown below), use the manifest file you created in the previous steps.



See Install the FortiMonitor Agent to learn more about Agent installation methods.

Agent manifest file

The manifest file is only used for initial installation. For more information, see Agent manifest file.

Use OnSight as a proxy for an existing FortiMonitor Agent

OnSight version 2020.68 or later

If your OnSight version is 2020.68 or later, you can use the following OnSight CLI commands:

To enable the proxy functionality, run:

```
sudo onsight configure-vcollector --enable-agent-proxy
```

To disable the Agent proxy, run:

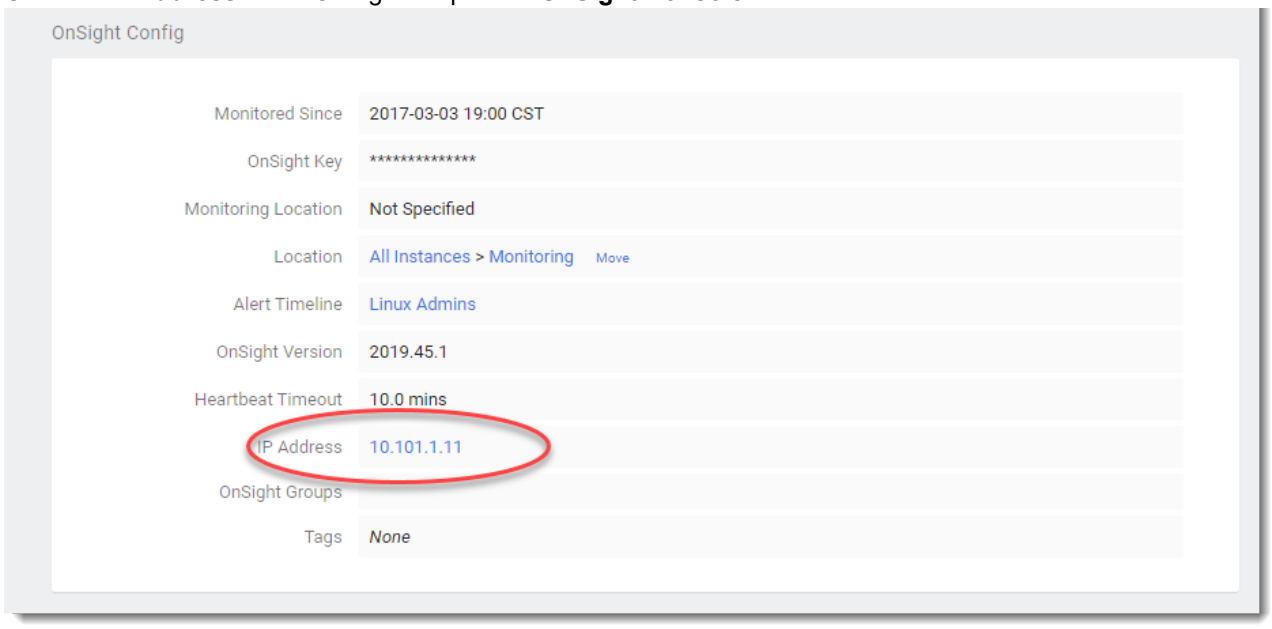
```
sudo onsight configure-vcollector --disable-agent-proxy
```

For more information, see Manage and configure your OnSight vCollector.

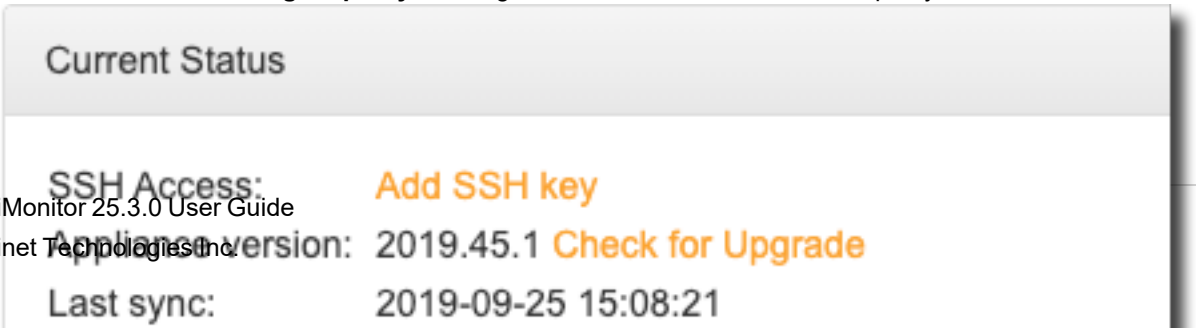
OnSight version 2020.67 or lower

To use OnSight as a proxy for an existing FortiMonitor Agent, perform the following:

1. Define the **Aggregator URL** in the FortiMonitor Agent configuration file. For Linux this can be found in **/etc/panopta-agent/panopta_agent.cfg**. For Windows, it is usually the **agent.conf** file in the directory you created for the FortiMonitor Agent. Keep this file open for later.
2. From the instance's tree in the control pane, select the OnSight instance to open its details page.
   Click the **IP Address** of the OnSight to open the **OnSight Console**.



3.
4. On the **OnSight Console** login page, enter the following credentials:
   - Username: **admin**
   - Password: `<OnSight key>`
     Successfully logging in will open the **OnSight Console**.
5. Select **Enable** in the **Agent proxy** field to get a URL that can be used as the proxy.



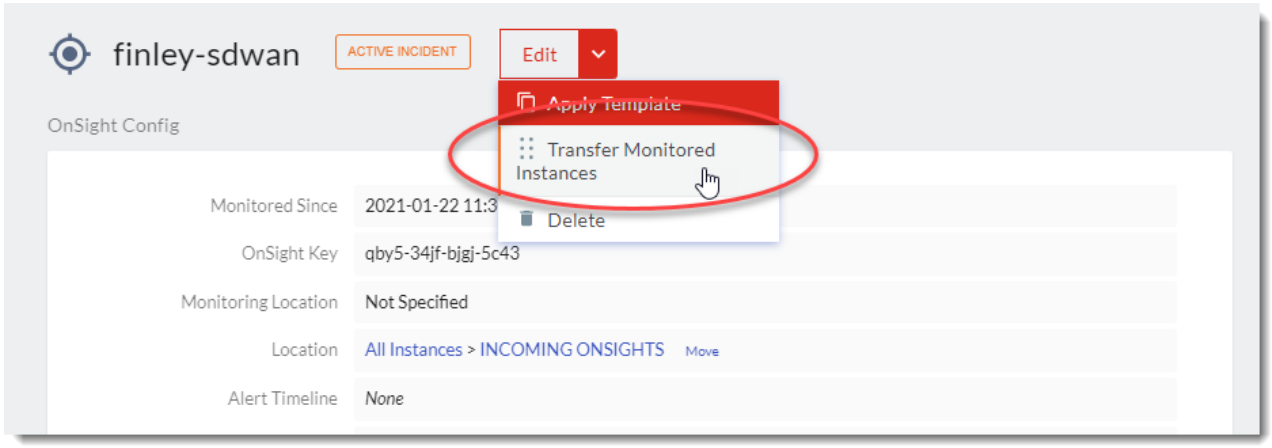
If you replace the aggregator URL value within the Agent configuration file with the OnSight Agent Proxy URL, all Agent communication will flow through the proxy. You can also place multiple URLs should you have more than one OnSight. This introduces high availability to your internal monitoring to ensure that you are always receiving the Agent metric data, even if one of your OnSight instances is not responding.

Move instances from an OnSight to another

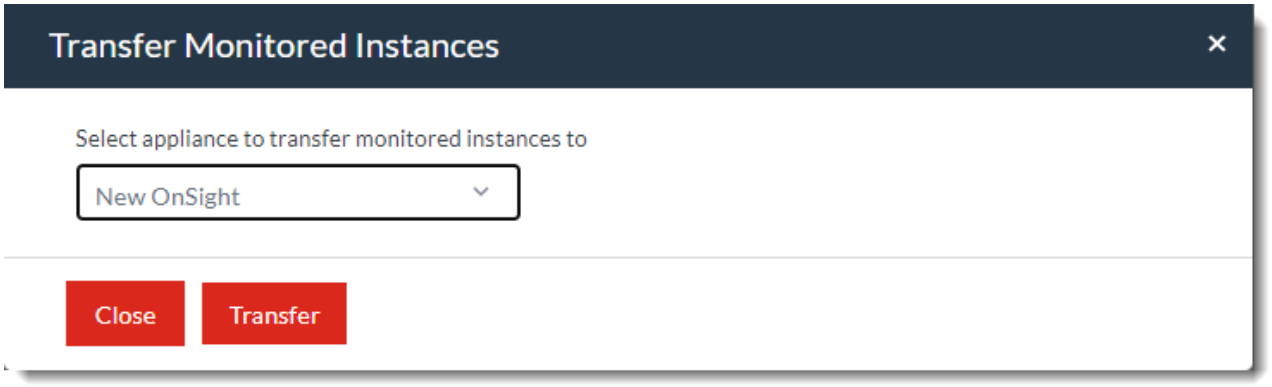
Note: The following steps should not be performed on any OnSight belonging to an HA (high-availability) group. This may cause issues with data collection.

You can move all of the instances monitored by an OnSight to another by performing the following steps:

1. Navigate to the OnSight page where you want to move instances from.
2. Select **Edit** > **Transfer Monitored Instances**.



3. Choose which Onsight to transfer monitored instances to.



4. Click **Transfer**. The instances will now be monitored by the selected OnSight.

OnSight security

Deploying OnSight onto the private segment of your cloud allows you to gain uptime and performance insight on each of the individual nodes and resources in your application stack. This additional insight helps diagnose problems you may encounter and helps you detect issues before they result in downtime or major service degradation. Combining the view of your internal and external infrastructure delivers the complete visibility you require to provide your end-users with the best experience possible.

Network Communication + Encryption

The OnSight™ appliance exclusively communicates with the FortiMonitor cloud via an outbound encrypted (TLS 1.2) connection. It establishes the outbound connection to securely send monitoring data and events back to our SaaS cloud to power all of your reporting, notifications, and dashboards, as well as to download monitoring configuration. No inbound connectivity is required which keeps your private infrastructure unexposed. In addition, if you have servers that do not have outbound internet access, you can install our monitoring agent and configure it to send its data to the OnSight™ appliance instead. The OnSight™ appliance operates as a proxy, enabling server monitoring on your private servers without requiring outbound access.

Automatic Updates

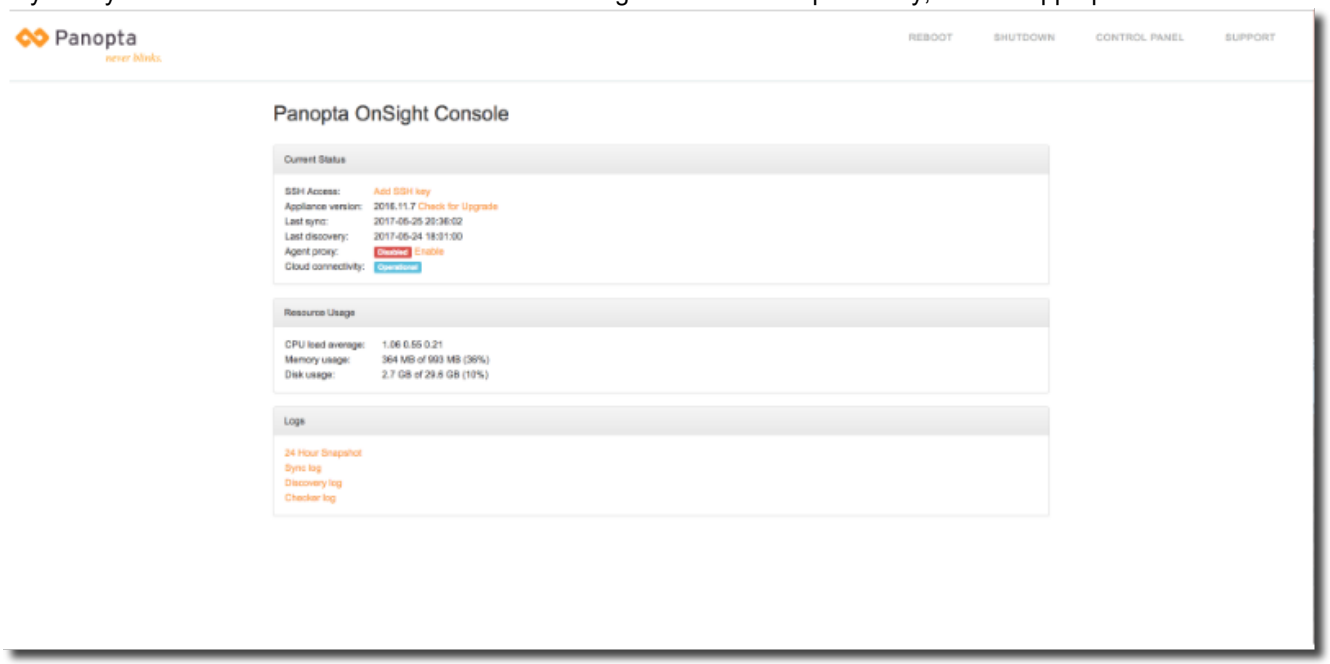
The OnSight virtual appliance automatically applies all critical patches from the Ubuntu core repositories.

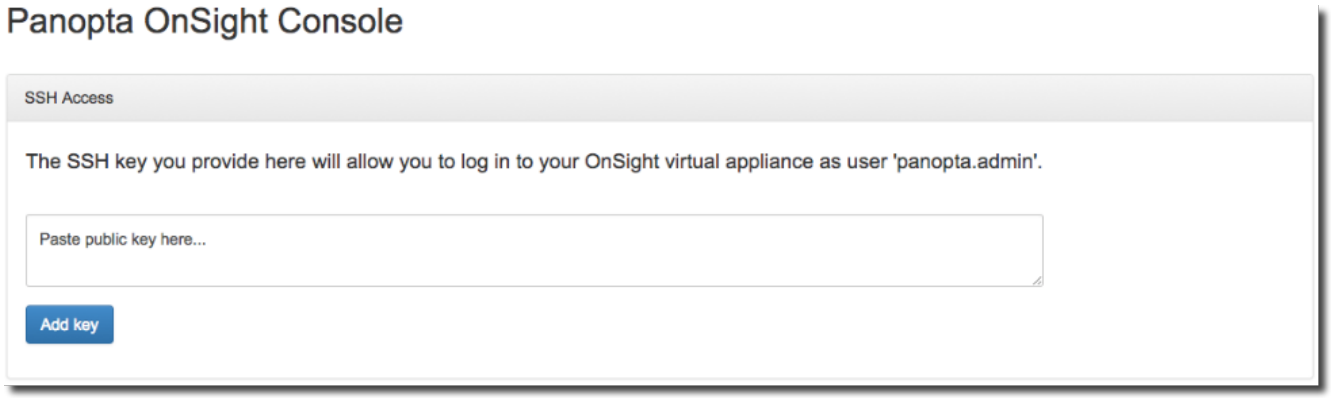
Set up SSH access to the OnSight virtual appliance

Newer OnSights

Newer OnSight images have the password set for the **panopta.admin** user as the OnSight key.

Under certain scenarios, you may require shell access into the OnSight virtual collector. Login to the vCollector VM is only allowed via SSH keys. Start by visiting the OnSight web console. From that page, you"ll be able to enter your public key from your local machine. For instructions on how to generate an SSH public key, visit the appropriate section below.





Once the SSH public key has been added to the OnSight, allow a few minutes for that key to be imported and then proceed to login via your SSH client of choice using the username `panopta.admin` and the IP of the vCollector.

Creating an SSH Key for Windows

Windows requires two programs, PuTTY SSH client and PuTTYgen. These tools can be downloaded here. Under the conversion tab on the PuTTY Key Generator, there is an option to generate a key, and then options to save the public and private key. Ensure that the private key is saved on your computer, only the public key goes into the OnSight SSH console. For more detailed instructions, read along here.

Creating an SSH Key for Linux or MacOS

Unlike Windows, Linux and MacOS do not require a key generator like PuTTY. The SSH public and private key generation can be generated via the command line. Similar to the instructions for Windows, the private key will be stored on your machine, and the public key will be pasted and saved on your OnSight web console. If you'd like some more detailed instructions, please read here.

AMI OnSight vCollector rotation procedure

This page outlines the steps required to replace an OnSight vCollector from an AMI while preserving all existing checks and historical data associated with the original OnSight vCollector. This document assumes the user has a basic knowledge of Amazon Machine Images and/ or other disk images as well as basic knowledge of virtual machines provisioning.

Installation Procedure

- The standard AMI is available within the AWS Marketplace by searching for FortiMonitor.
- Keep a text editor open as some of the following steps require some temporary notation.
- There will be a short window of downtime while completing these steps, typically 1-2 minutes.

The instructions below outline the procedure for replacing an existing OnSight.

1. Provision a new VM using the OnSight vCollector machine image.
2. Perform appropriate network configuration.
3. Obtain the customer key for your tenant - this may be found via the Control Panel by navigating to the portrait icon in the upper right corner of the navigation bar and looking under **My Account**. Note this for a later step.

4. Obtain the OnSight key for the current OnSight vCollector that will be replaced. This may be found by navigating to the details page for the OnSight vCollector in the Control Panel. Note this for a later step.
5. Shutdown the existing OnSight vCollector that is being replaced. The container or VM should be disabled to prevent any future registration issues with this appliance.
6. On the newly created OnSight vCollector, run the following commands from **Root-level** access (items in CAPS should be replaced with the respective key values saved in steps 3 and 4):

```
onsight configure-vcollector --customer-key <customer-key>
onsight configure-vcollector --appliance-key <appliance-key>
onsight configure-vcollector --aggregator-url <aggregator-ur>
onsight start-vcollector
```

7. Return to the Control Panel, navigate to the details page of the OnSight vCollector, and verify data flow has been restored.

Rotating clustered OnSights

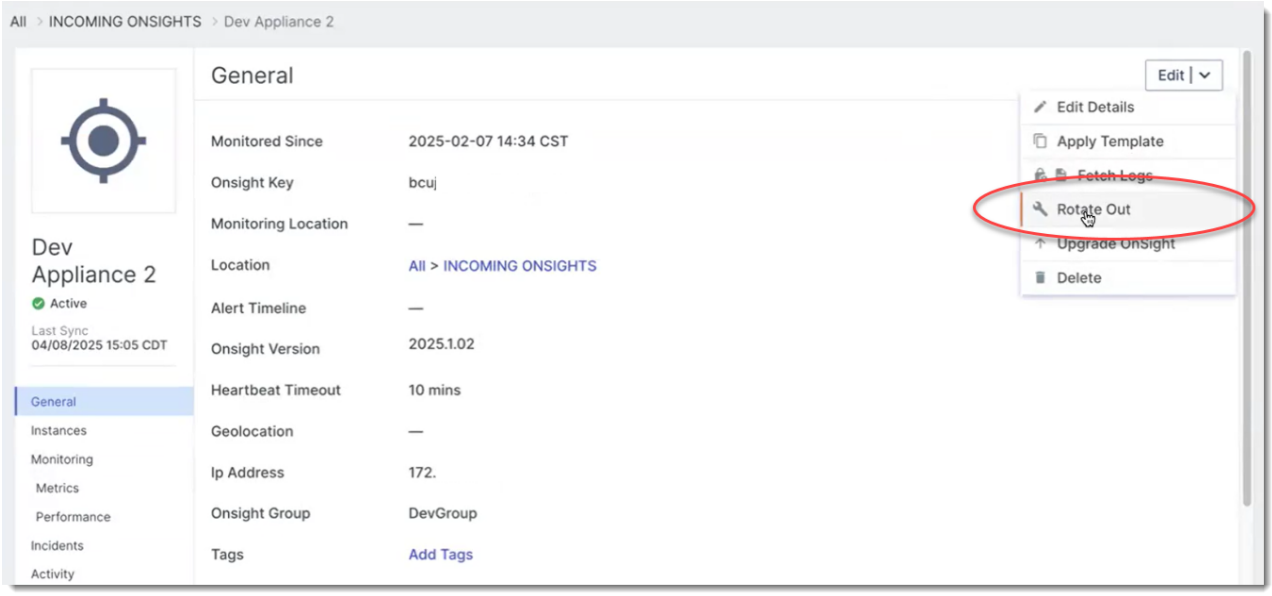
This process helps you upgrade an OnSight in a cluster without causing gaps in collected monitoring data. It works by moving all metrics from the OnSight you plan to take offline to another OnSight in the same cluster. While this rotation is happening, no heartbeat incidents will be triggered. Note that an incident will be opened while the OnSight is out of rotation and will be closed once it returns. This incident will not appear in the history or affect availability calculations.

You will get an email when the rotation is complete and another when the original OnSight is fully restored.

This feature only works for OnSights that are part of a cluster.

To rotate an OnSight out, perform the following steps:

1. Navigate to the desired OnSight in the control panel and click the **Rotate Out**. Confirm the action by clicking **I'm Sure**.



2. The system will automatically reassign all metrics monitored by the selected OnSight to another available OnSight in the same cluster.
3. Wait until you receive confirmation that the rotation process is complete. This confirmation will appear in the control panel and will also be sent via email.

Note: Do not shut down the OnSight until you receive the rotation complete notification. Premature shutdown may result in monitoring data gaps.

4. If needed, you can cancel the rotation process by going to **Edit** > **Cancel Rotation**. This will roll back all changes and return the metrics to their original OnSight.
5. Once rotation is complete, you can safely take the OnSight offline to perform an upgrade or other maintenance without triggering a heartbeat incident.
6. After maintenance, restart the OnSight using the following command:

```
onsight start-vcollector
```

7. Once the OnSight is running again, the system will automatically move the metrics back to the original OnSight.
8. You will see an **OnSight Upgrade Complete** banner in the control panel and receive an email notification once restoration is complete. There should be no gaps in collected metrics if the rotation is successful.

Third-party integration

As networking equipment vendors transition to a post-SNMP era, offering custom APIs for access to device and metric data, a general framework has been developed to support third-party integrations with the OnSight vCollector.

Third-party integrations are deployed using a sidecar Docker container installed on the OnSight. This container houses all the necessary vendor-specific integration logic and communicates with the core OnSight container, which manages the communication between the integration and FortiMonitor.

Note: Currently, only Alkira and Arista integrations are supported. In upcoming feature updates, you'll be able to create custom integrations using a provided template or even develop them from scratch.

The following section defines the key terms used throughout this article.

- **Integration**: The sidecar container that houses all the third-party specific logic.
- **Integration configuration**: One or more configuration sets that are stored centrally and provided to the integration.
- **Instance**: A specific device or endpoint being monitored.
- **Metric**: An individual measurement associated with an instance that can be collected by the integration.

Integration workflow

1. **Installation**: This process involves adding the container and enabling it to run.
2. **Integration Configuration**: You then configure details centrally to define the actions the integration should perform. This step is optional, but it is similar to configuring a Fabric or VMware connection. It specifies what to connect to and the credentials to use. These configuration details are sent back to the OnSight and provided to the integration container.
3. **Instance Discovery**: After the configuration is complete, the integration performs discovery based on the configuration and returns a list of available instances that can be monitored. Discovery results are sent to FortiMonitor, where you can either manually select instances to provision. If auto-provisioning is enabled, instances will be automatically created.
4. **Metric Discovery**: Using the integration and instance configuration details, the integration returns a list of available metrics that can be gathered. These metrics can be global or specific to each instance and may vary depending on the instance type. Different integrations will handle metric discovery differently based on the complexity of the monitored environment. Metadata about available metrics is sent to FortiMonitor for auto-provisioning or to be made available to users via the control panel.
5. **Metric Collection**: The core functionality of the integration involves collecting metric values for the metrics that have been provisioned. These values are then transmitted to FortiMonitor for storage, incident management, and inclusion in dashboards and reports.

Prerequisites

- Latest version of the OnSight
- Configured third-party integration

Note: Currently, only Alkira and Arista integrations are supported. In upcoming feature updates, you'll be able to create custom integrations using a provided template or even develop them from scratch. See the following sections for details on how to configure third-party integration support for Alkira and Arista:

- Arista CloudVision integration
- Alkira integration

Arista CloudVision integration

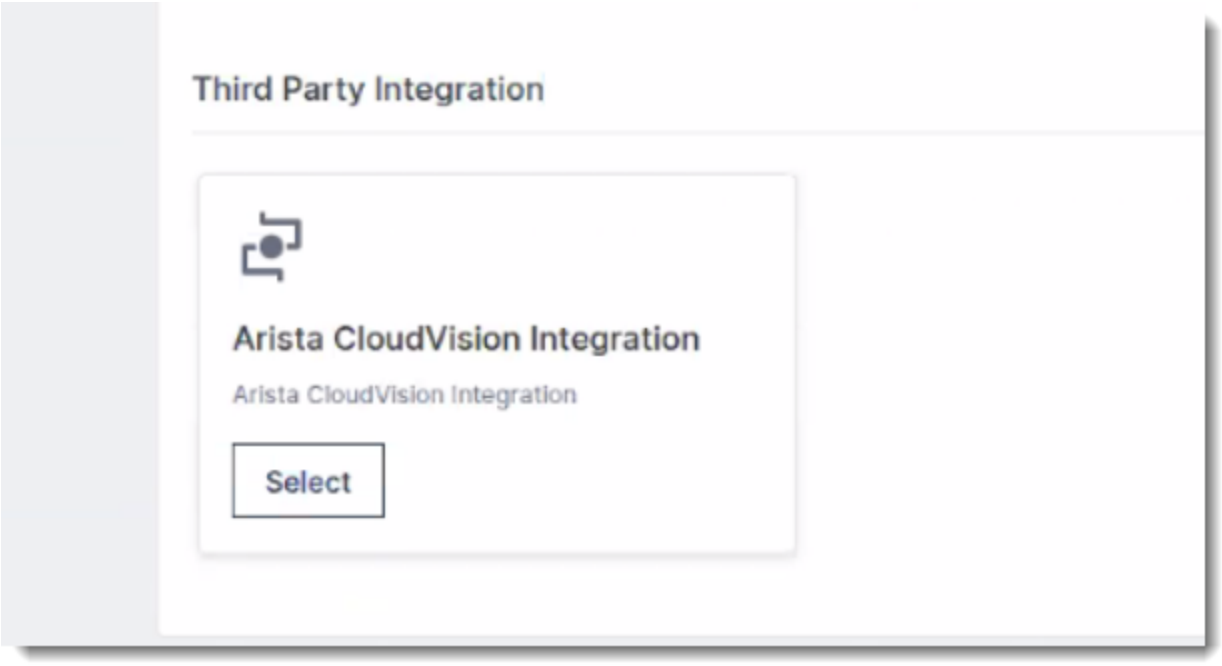
Integration installation

1. Enter the following command to install the integration:
   `onsight-integ install panopta/onsight-integ-cloudvision`
   Use the command `onsight-integ install --help` to see more installation options.
2. Once the image is running and if there are no issues, the new integration will show up in the Control Panel.

Control panel configuration

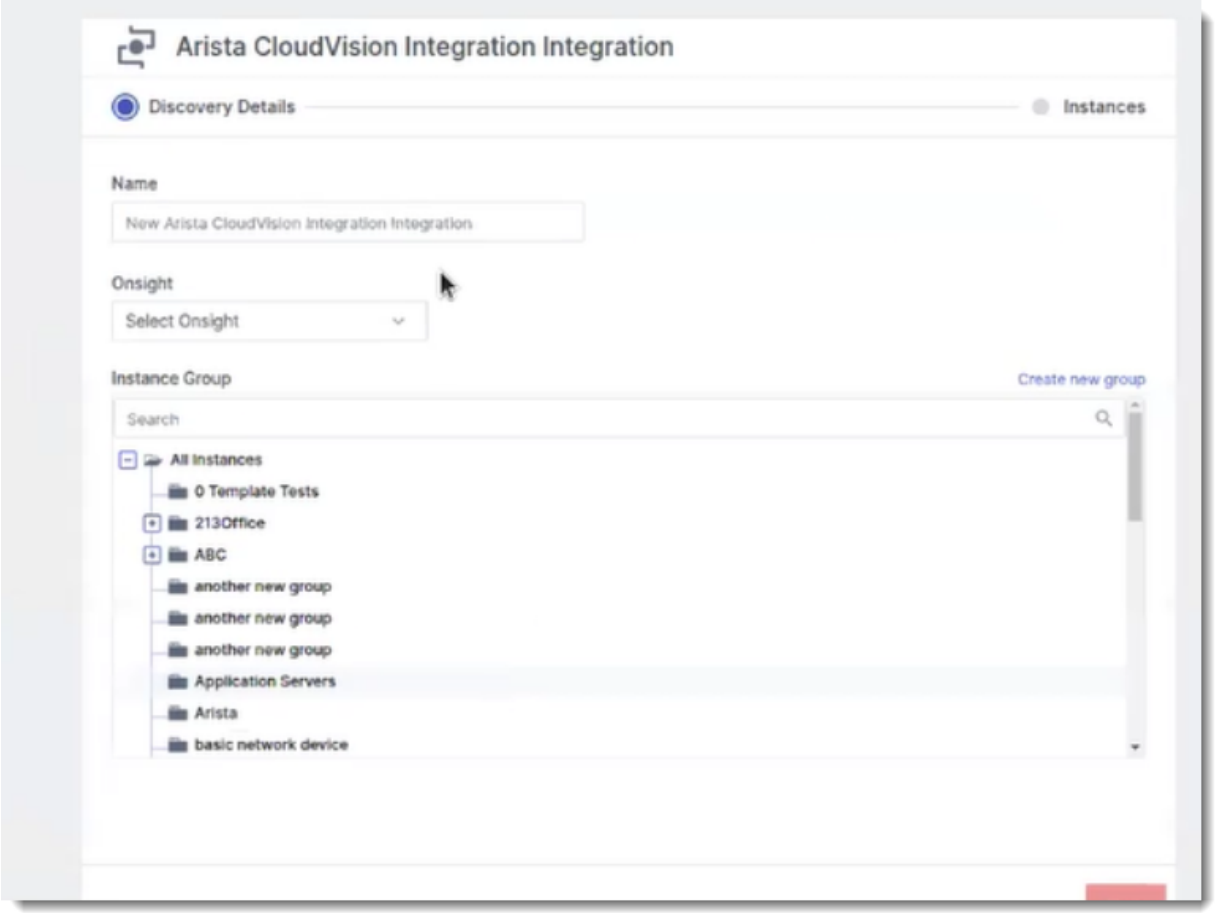
This allows FortiMonitor to subscribe to metrics from Arista CloudVision.

1. From the navigation menu, click **Add**. The **Infrastructure and Resource Catalog** is displayed.
2. Select the third-party integration that you created.

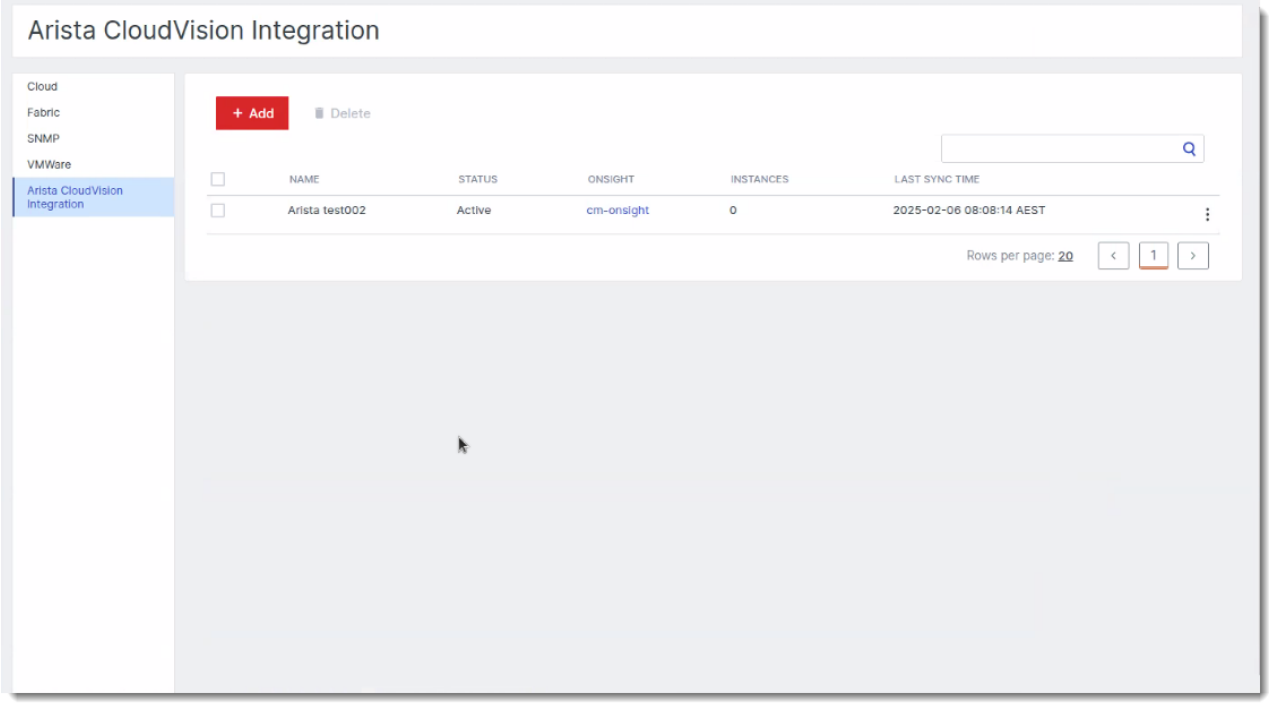


3. In the modal:

a. Enter a name.
b. Select the OnSight where the third-party integration is configured. In this case, choose the OnSight with the sidecar configured for Arista CloudVision.
c. Select an instance group.
d. Add your Arista CloudVision token.
e. Enter a domain (optional). If no information is entered, the default value is used (www.arista.io).



4. Click **Next** to begin the discovery process. This may take a few minutes to complete.
5. Once discovery is complete, a list of discovered instances will be displayed, along with their attributes, type, and tags.
6. Select the checkboxes for the instances you wish to monitor.
7. Click **Finish**.
8. After selections are applied, the configured integration and selected instances will appear on the **Infra Settings** page.



Additional integration commands

List integrations:

```
onsight-integ list
```

Command to remove integration:

```
onsight-integ remove <integration-name>
```

Command to upgrade the integration:

```
onsight-integ upgrade <integration-name>
```

Alkira integration

Integration installation

1. Enter the following command to install the integration:
   `onsight-integ install panopta/onsight-integ-alkira`
   Use the command `onsight-integ install --help` to see more installation options.
2. Once the image is running and if there are no issues, the new integration will show up in the Control Panel.

Control panel configuration

This allows FortiMonitor to subscribe to metrics from Alkira.

1. From the navigation menu, click **Add**. The **Infrastructure and Resource Catalog** is displayed.
2. Select the third-party integration that you created.
3. In the modal:
   a. Enter a name.
   b. Select the OnSight where the third-party integration is configured. In this case, choose the OnSight with the sidecar configured for FortiMonitor Alkira.
   c. Select an instance group.
   d. Add your Alkira AIPI key
   e. Enter your Alkira tenant FQDN. For example **tenant_name.portal.alkira.com**.
4. Click **Next** to begin the discovery process. This may take a few minutes to complete.
5. Once discovery is complete, a list of discovered instances will be displayed, along with their attributes, type, and tags.
6. Select the checkboxes for the instances you wish to monitor.
7. Click **Finish**.
8. After selections are applied, the configured integration and selected instances will appear on the **Infra Settings** page.

Additional integration commands

List integrations:

```
onsight-integ list
```

Command to remove integration:

```
onsight-integ remove <integration-name>
```

Command to upgrade the integration:

```
onsight-integ upgrade <integration-name>
```

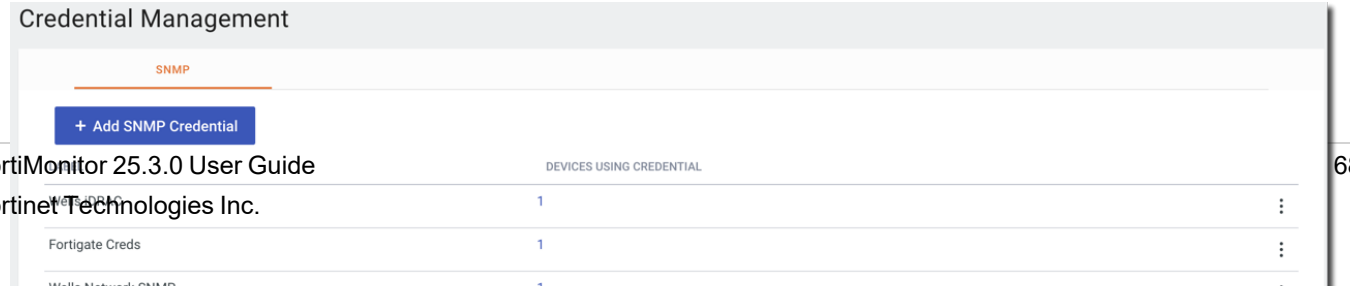
SNMP

SNMP is an effective way to monitor the health of network devices that you cannot monitor with an Agent. Our OnSight vCollector enables you to collect SNMP metrics in your account alongside your other monitoring like server health, application metrics, and network uptime, and availability.

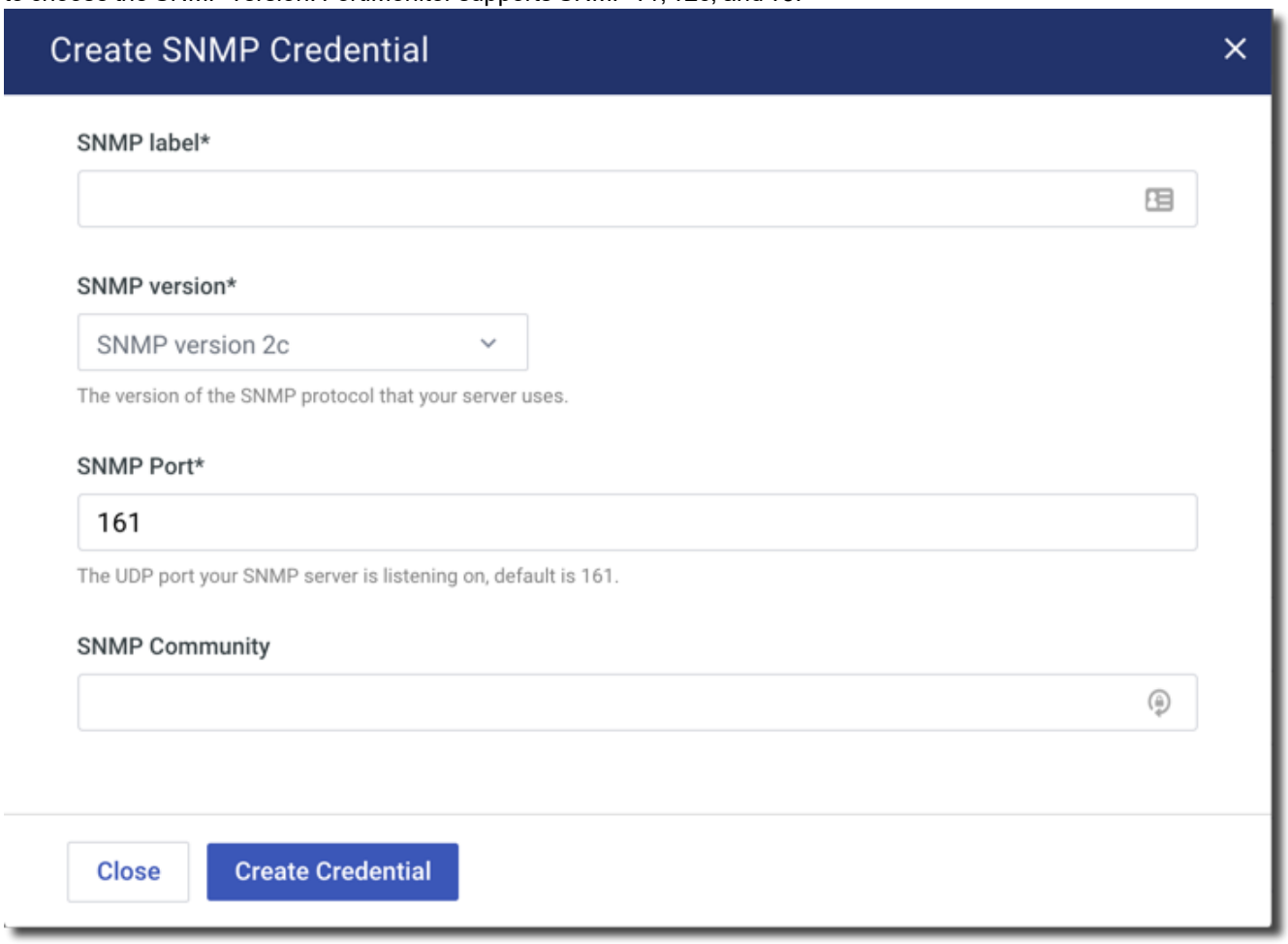
Create an SNMP credential

To create an SNMP credential, perform the following steps:

1. Select **Monitoring** > **Infra Settings** > **SNMP Credentials** from the main navigation header.
2. On the **Credential Management** page, select **+ Add SNMP Credential**.



There are a few different variables to fill in, such as the SNMP Label, which is the user-friendly reference to the credential. You'll use this when you are adding the credential to devices in the FortiMonitor control panel. You also have to choose the SNMP version. FortiMonitor supports SNMP v1, v2c, and v3.



The fields are described in the following table.

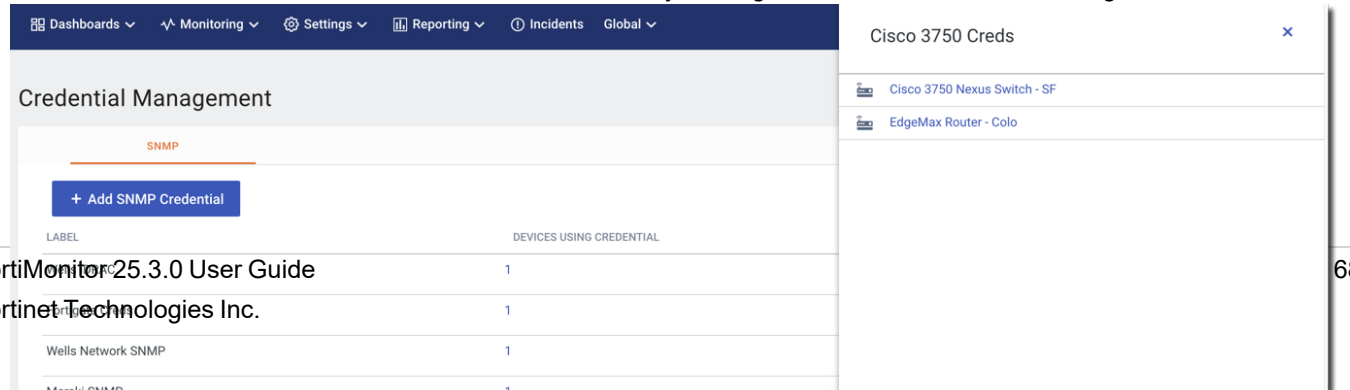
Field	Description	Version 1 and 2c	Version 3
SNMP label	Enter a name for this SNMP credential.	✓	✓
SNMP version	Select the SNMP version for this credential. **Values**: • SNMP version 1 • SNMP version 2c • SNMP version 3	✓	✓
SNMP port	Enter the UDP port your SNMP server is listening to. **Default**: 161	✓	✓

SNMP community	Enter the SNMP community string for this SNMP credential.	✓	
SNMPv3 User	Enter the associated SNMP user name.		✓
SNMPv3 Authentication Algorithm	Select the authentication type to use for this SNMP credential. **Values**: • MD5 • SHA • SHA-224 • SHA-256 • SHA-384 • SHA-512 • No Authentication		✓
SNMPv3 Authentication Key	Enter the key to be used for user authentication. The *key* is the encrypted version of a cleartext password		✓
SNMPv3 Encryption Algorithm	Select the encryption algorithm to use for this SNMP credential. **Values**: • DES • AES • AES 128 • AES 192 • AES 256 • AES 56C • No Encryption		✓
SNMPv3 Encryption Key	Enter the key associated with your selected encryption algorithm.		✓

3. Once the SNMP credential is set up, you have to import your network devices. There are two main ways to accomplish this; manually and with OnSight Discovery. If you want to use OnSight discovery or the devices are already in FortiMonitor, you can skip this section.

View the list of devices associated with an SNMP credential

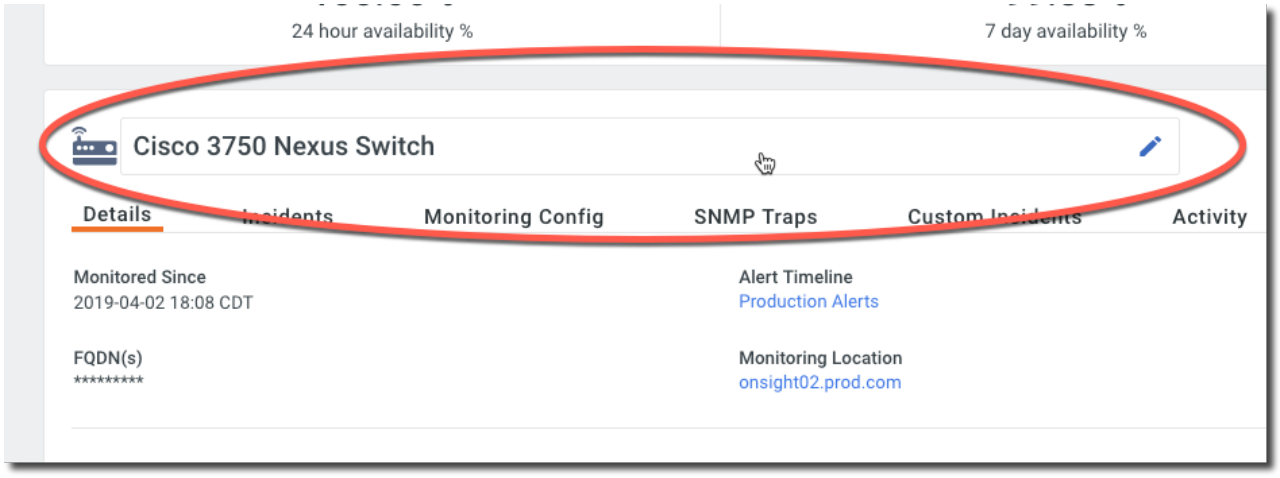
You can view a list of devices associated with a credential by clicking on the number of devices using the credential.



Configure the network device

To configure a network device, perform the following steps:

1. Go to the **Instance Details** page of the network device that you want to configure.
   Select the name of the network device or the **Edit** icon.



2. The **OnSight Configuration** drawer will slide out.
3. Edit the following settings as needed:

Option	Description
General	
Name	Change the name of the network device.
Server Key	The unique key used to reference this device.
Instance Group	Select an instance group for the network device.
Tags	Enter tags that can help define the network device. For example network, cisco. Press Enter after each tag. Known tags will be auto-suggested to you. Tags should not contain spaces.
Attributes	Add key-value pairs that can provide more structure to your instance metadata.
FQDNs and IPs	Add additional interface names. See Additional IP addresses or FQDNs.
Description	Enter a description for the template.
Parent Instance	(Optional) The device upstream from this server that it depends on. If the parent device experiences an outage, this server's outage notifications will be suppressed.
Prometheus Endpoints	Add a Prometheus endpoint to the network device.
SNMP Credentials	Add an SNMP credential to the template.
Select Credentials	The SNMP credential used for this device.
Recurring SNMP Discovery	Enable this option if you want the OnSight to regularly scan the network device. Detected metrics will automatically be provisioned.

SNMP Scan Frequency	This option only appears when the **Recurring SNMP Discovery** option is enabled. Specify how often (minutes) the network device is scanned for new metrics.
Port Metric Provisioning	Enable this option to auto-provision SNMP metrics.

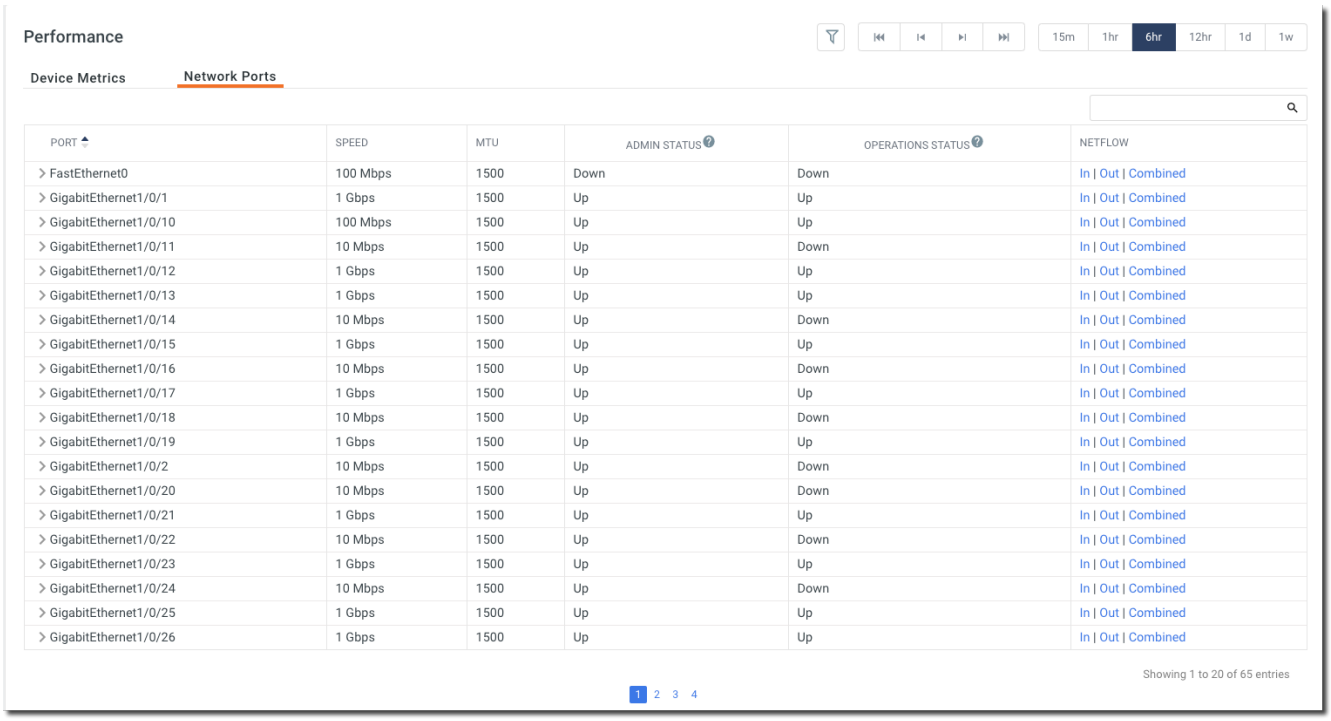
4. Select **Save Changes**.

Monitor metric performance

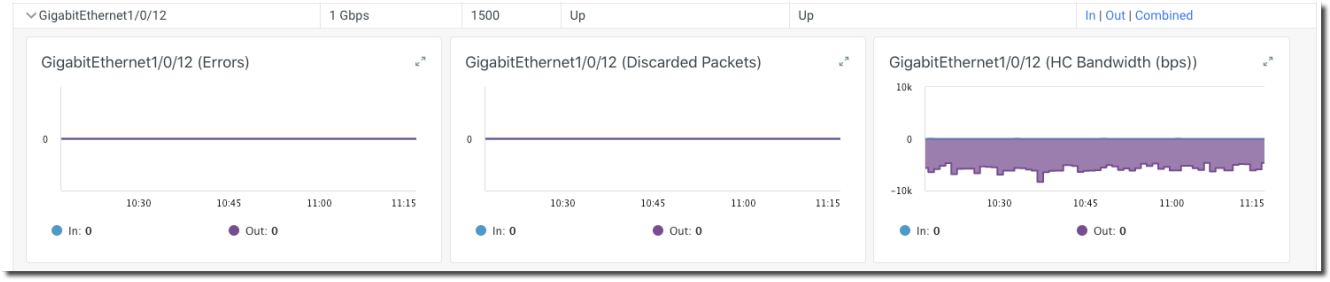
In the **Performance** section of the network device's **Instance Details** page, each metric is presented differently. Device-level metrics use performance line charts along the top.



Port/interface-level metrics use a customized roll up display which shows interface stats in expandable rows.



Click on a specific row to expose all interface-specific performance and health metrics. The graphs will show both inbound/outbound stats stacked on the same graph with zooming ability. Click on the expand icon in the upper right-hand corner of a graph to see a fullscreen display.



Using templates allows you to standardize your thresholds set and alerts and implement them across similar devices.

SNMP vendor documentation

The following is a list of some of the supported SNMP vendors and links to their respective SNMP configuration help pages.

- Fortigate
- Cisco
- Meraki
- Palo Alto
- Juniper
- F5

SNMP traps

The FortiMonitor OnSight vCollector can act as a trap receiver for SNMP monitored devices. You can configure trap alerts by pointing your network devices to send traps to OnSight by adding the IP address of your OnSight vCollector and sending the traps on port 162. SNMP traps are accessible to all accounts, provided all of the following are true:

Requirements

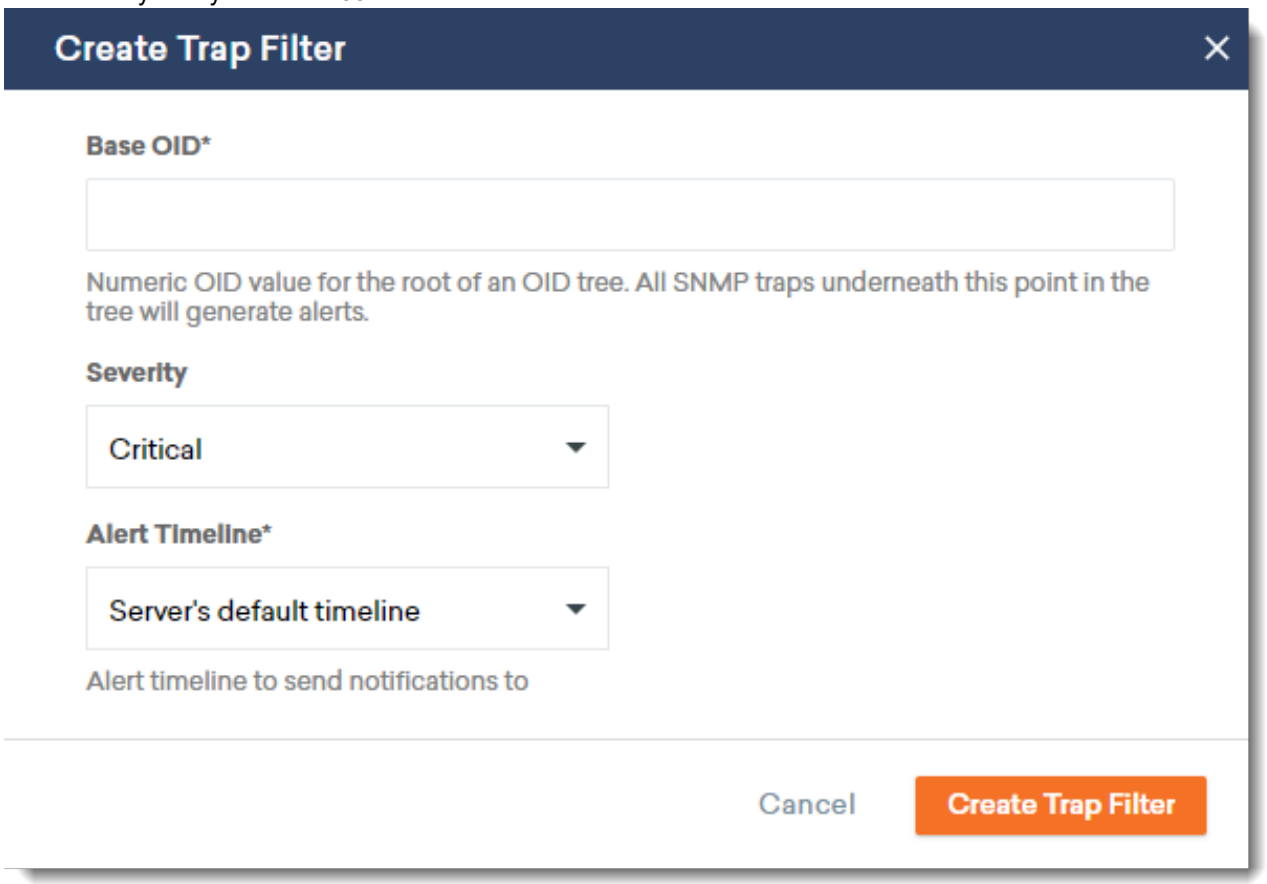
- The instance is of type network device - if you have existing instances you would like to convert, contact our support group
- That network device has a primary monitoring location set to an OnSight vCollector
- The latest version of the OnSight vCollector

When a trap is fired, the OnSight receives the trap containing a message with the IP address of the device, the OID, as well as the value. When the OnSight syncs back with our cloud, the alert timeline that you have configured for that particular device will begin and alerts will be sent.

SNMP trap configuration can be baked into a template and stamped onto other devices for common trap configuration.

To set up an SNMP trap, perform the following:

1. On the network device's **Instance Details page**, Select the **SNMP Traps** tab.
   To add a new trap filter, click **+ Add Trap Filter** and configure the trap.
   The **Base OID** field is for determining the type of event you'd like to be notified about using a match pattern. The incident that will be created will use the severity and alert timeline that you specify in this configuration. Also, incidents that are created by traps have an option to manually close/resolve. Otherwise, they will be automatically closed out by the system after 60 minutes.



2.

3. Select **Create Trap Filter**.

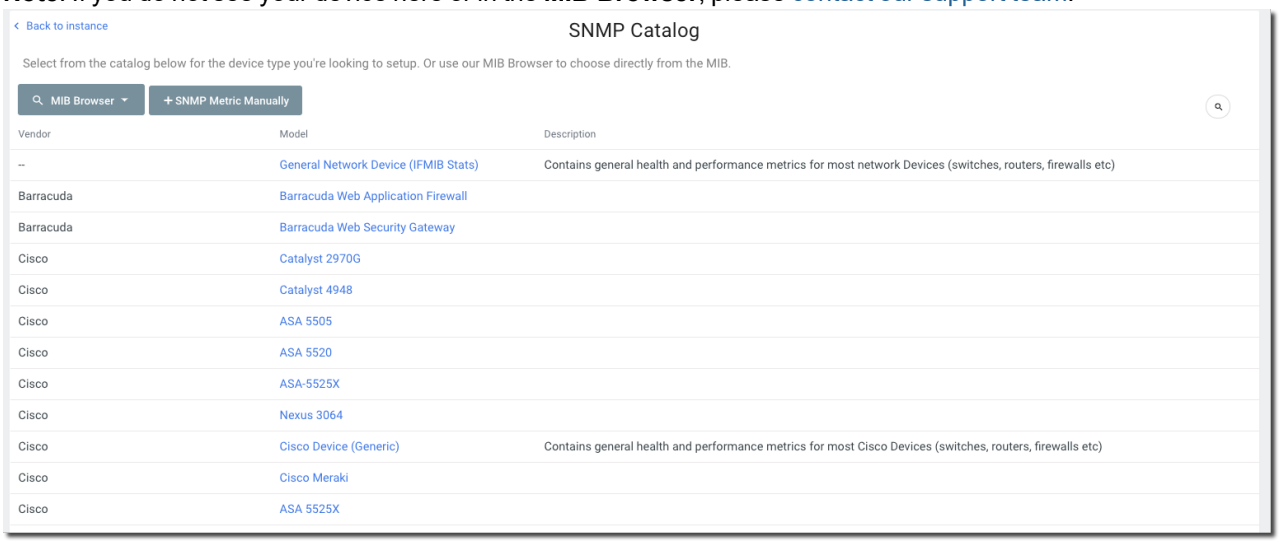
SNMP polling

After adding a network device to FortiMonitor, the OnSight automatically detects its type then provisions the appropriate standard metrics. If you require additional metrics beyond the provisioned standard metrics, use the **SNMP catalog** to add the metrics. SNMP polling is available to users who have installed an OnSight vCollector on their private network and want to monitor SNMP-capable network devices. For instructions on how to set up an OnSight vCollector, see OnSight vCollector overview.

To get started with SNMP polling, perform the following steps:

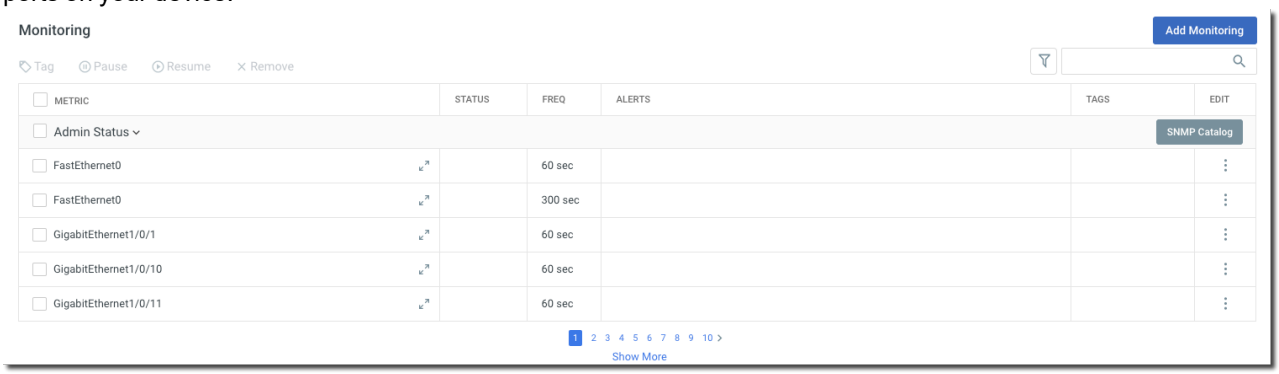
1. From the **Monitoring Config** tab of the network device's **Instance Details** page, select **Add Monitoring** > **SNMP**.
2. Select **Configure Metrics**.

   Select a model from the **SNMP Catalog**. You can also use the MIB Browser or, if you know the OID, add the SNMP metric manually.

   Note: If you do not see your device here or in the **MIB Browser**, please contact our support team.
3. 

   Selecting the model type will bring you into the catalog with the OID's for that device. The list includes the name, OID, and a description of the metric. You can also search for a particular metric with keywords that may be contained in the description or name using the magnifying glass icon at the top right.

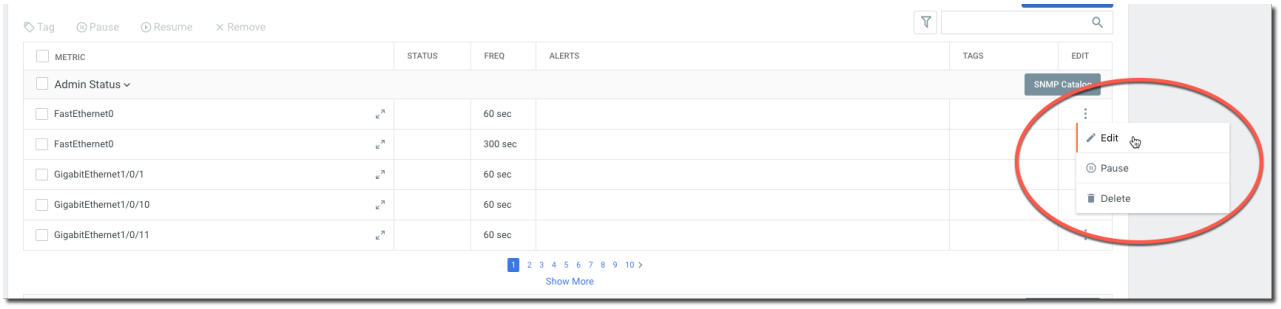
   Select the metrics to monitor for this device.

   Metrics like **Admin Status** or **ifInOctets** will expand into a number of individual metrics, based on the number of ports on your device.
4. 

Configure SNMP alert thresholds

To add a threshold to an SNMP metric, perform the following steps:

1. Select the 3-dot menu of the metric then select **Edit**.



2. From the metric configuration drawer, select the **Thresholds** tab.
3. Select **+ Add threshold**.
4. Configure the following fields:
   - **Generate alert when *metric* is** - Set the alert threshold such that if that number is greater than, less than, equal, or not equal to a number you specify for an amount of time that you also specify, an alert will trigger.
   - **For more than** - Specify a time.
   - **Using severity** - Select a severity (Critical, Warning).
   - **Notify with alert timeline** - Select the Create an Alert Timeline
   - **Include in availability calculation** - Select if you want to include this alert to your server's availability calculation.
5. Select **Save**.

SNMP troubleshooting

If you've configured an SNMP metric to be collected and it's not returning any data, it could be related to one of the following issues:

1. The SNMP credentials are incorrect.
2. The network device is not reachable from the OnSight vCollector.
3. The OID you are entering may not exist on the device you are querying.

In order to identify which problem it is, we recommend running an SNMP walk with all the parameters while logged into the OnSight vCollector itself. If any of the required OID's which are returned from the walk don't exist in your FortiMonitor dashboard, our support team can import additional MIBs as well.

The command needs to be run from the OnSight set up in your account, which you'll need to SSH into. If you haven't set up SSH access, read this document to access the web console. Then follow these instructions for setting up SSH.

The command for SNMP walk is:

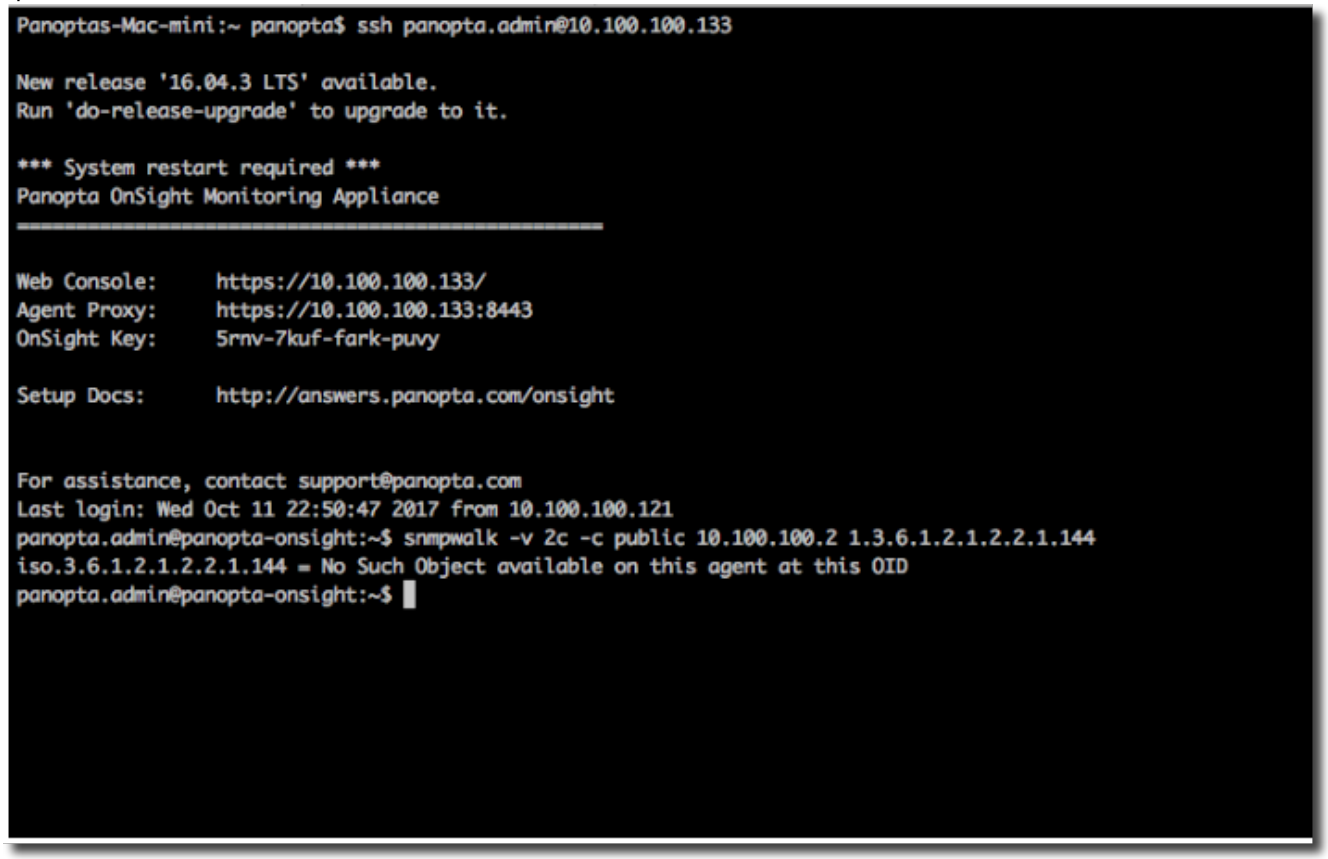
```
V2c

snmpwalk -v 2c -c [SNMP credential] [IP of device] [OID]

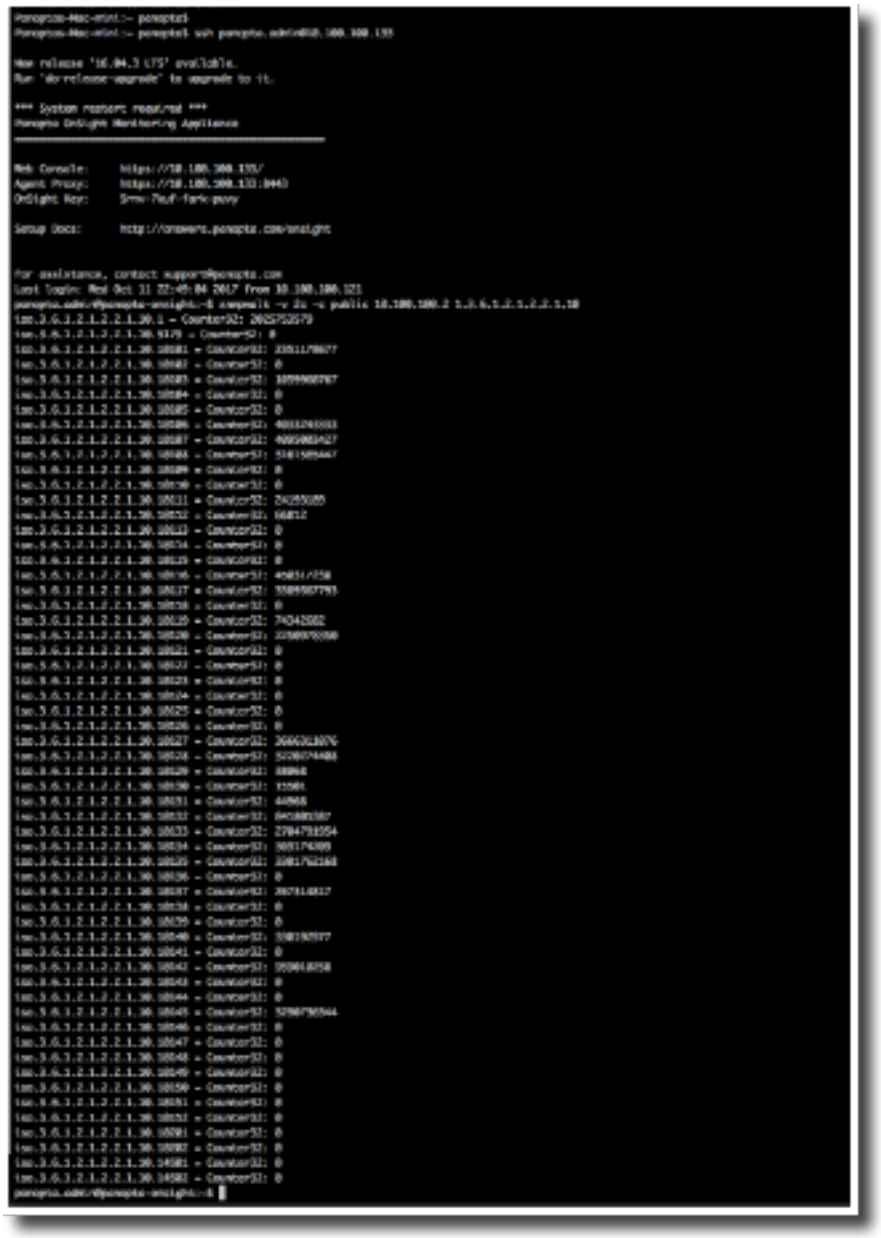
v3
```

```
snmpwalk -v3  -l authPriv -u snmp-poller -a SHA -A "PASSWORD1"  -x AES -X "PASSWORD1" [IP of device]
```

If you do not get any output, or if the output looks like the image below. Try running the SNMP walk without the OID specified.



A successful SNMPwalk output should look like the image below.



Once you have established that you have the correct SNMP credentials, ensure that the metrics present are the ones you're looking for. If not, you may need to reach out to your network device provider and request access to the proprietary MIBS.

If you cannot get a successful SNMPwalk, please contact our support team.

You can add custom plugins that activate after the SNMP filtering process. However, you still need to configure SNMP trap filters by OID as you currently do. The plugin will activate only after a trap has passed through the filtering stage.

Multiple custom plugins can be created. The system executes file logic by sorting "textkey" in alphabetical order. At startup, all provided plugins load, and the system checks for new ones.

Trap processing

When processing a trap:

1. If the trap passes the initial SNMP filter, it will be passed through each loaded plugin in order.
2. Processing stops if any plugin returns **True**.
   - A log entry is added indicating that the plugin handled the trap.
   - The trap is marked with a **Filtered** status.
   - The corresponding SNMPTrapQueue record is deleted from the database.

Plugin installation

Custom plugins should be added to the **/usr/share/onsight/custom-traps** directory on the host VM. These plugins must subclass the **SNMPTrapPlugin** parent class, which defines a single method:

```
class SNMPTrapPlugin:
    def filter_trap(self, source_ip, source_hostname, trap_oid, varbinds):
```

- `varbinds`: A dictionary of OID - value mappings for all OIDs in the trap payload.

The `filter_trap()` method must return a Boolean value:

- `True`: Indicates that the plugin acted on the trap and further processing should stop.
- `False`: Means the plugin did not take action on the trap.

For example:

Plugin path: **/usr/share/onsight/custom-traps/**

Each queue will process every script in the directory above. The resolved varbinds from the SNMP trap will serve as input for this plugin.



The custom plugin file name under **/usr/share/onsight/custom-traps/** can be any name except **SNMPTrapPlugin.py**.

Creating a custom SNMP trap plugin

1. Create an SNMP trap filter. In this example, we will use **1.3.6.1.4.1.50345** as the base OID.
2. Create the plugin in the **/usr/share/onsight/custom-traps** directory on the host VM.

```
import logging
```

```
from SNMPTrapPlugin import SNMPTrapPlugin

class Plugin3(SNMPTrapPlugin):
    textkey = "plugin0529"
    name = "plugin0529"

    def filter_trap(self, source_ip, source_hostname, trap_oid, varbinds: dict) -> bool:
        logging.info(f"{source_ip}, {source_hostname}, {trap_oid}, {varbinds}")
        if source_ip == "50.171.247.250":
            return True
        return
```

3. Verify in the logs that the trap was not processed.

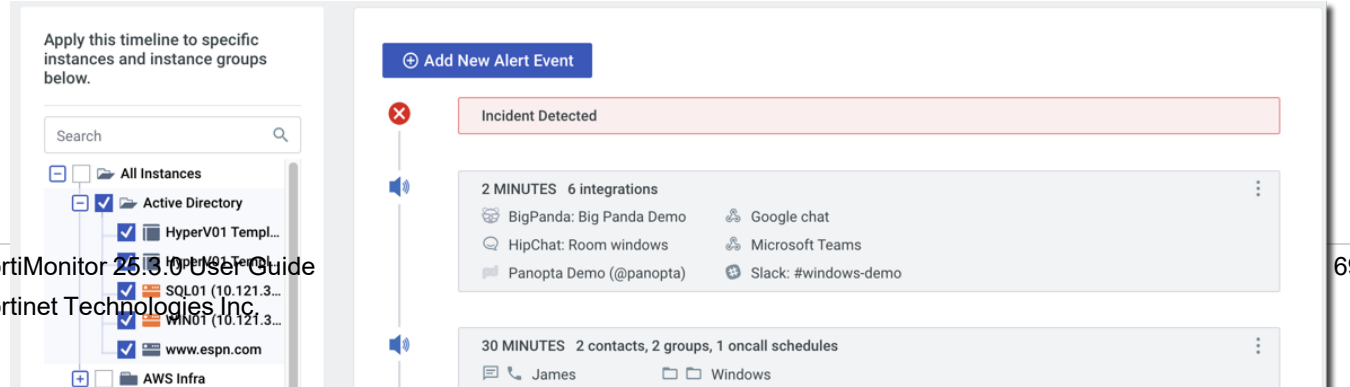
Alerting

FortiMonitor allows you to choose how you find out about outages and service degradation. The following sections detail how to properly configure alerts:

- Alert Timelines
- Email
- SMS
- Integrations
- Limit the number of confirmation checks for an outage
- Set up an On-Call schedule
- Multiple server outages
- Notification for an individual service on a server
- Long-running incidents
- Setting aside alerts on page 755

Alert Timelines

Alerts are sent out to a timeline that is highly configurable. You as the user can set the time you'd like to first know about an outage, and then build in escalations to other people or teams.



Once an outage is detected, you can choose how quickly after an outage you'd like to be notified, who gets the notifications, and on which channels. From there, if the outage isn't acknowledged or resolved by the next trigger event, the alerts will escalate as you configure them.

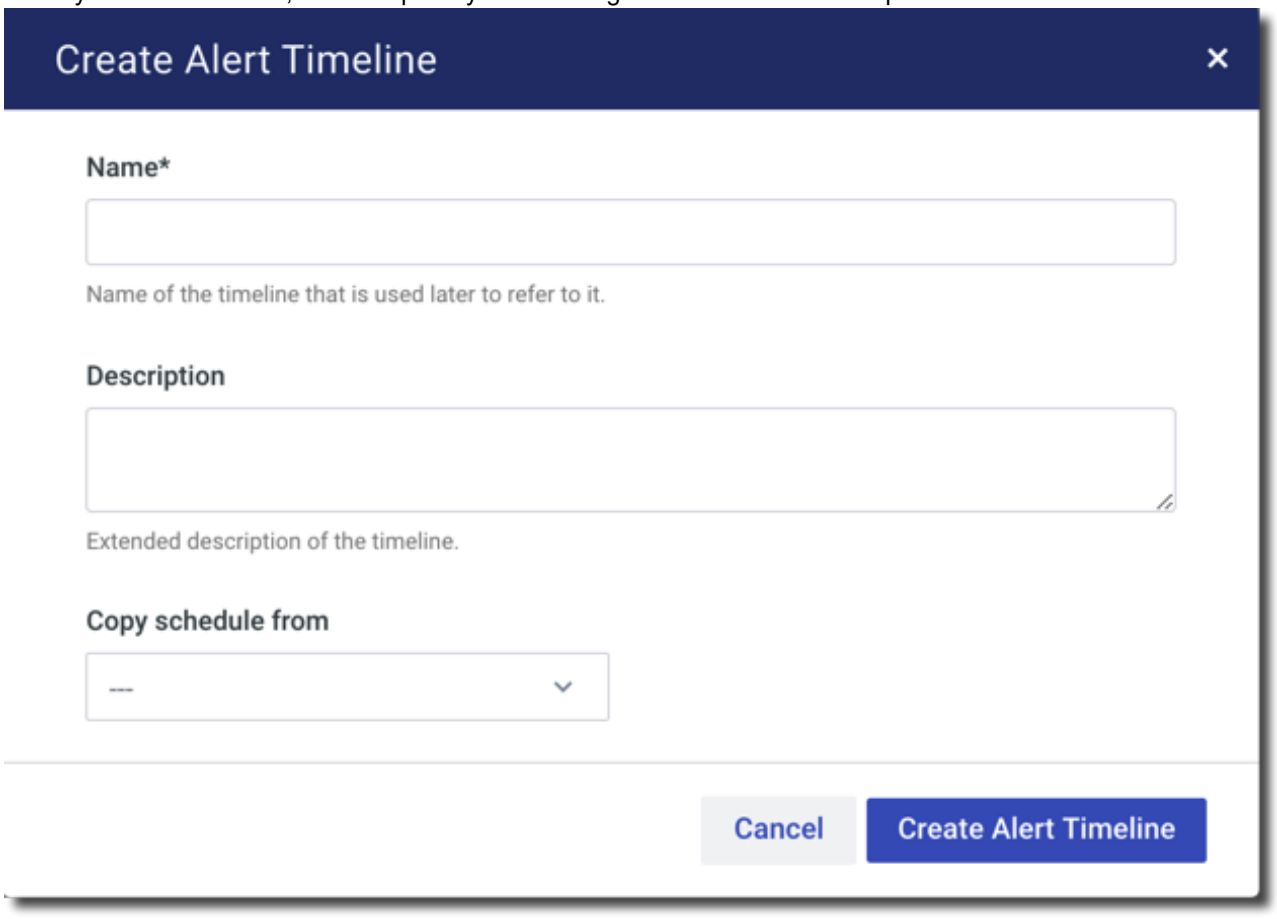
You can also see which servers and server groups this alert timeline is attached to, and edit from this view.

To create an Alert Timeline, see Create an Alert Timeline.

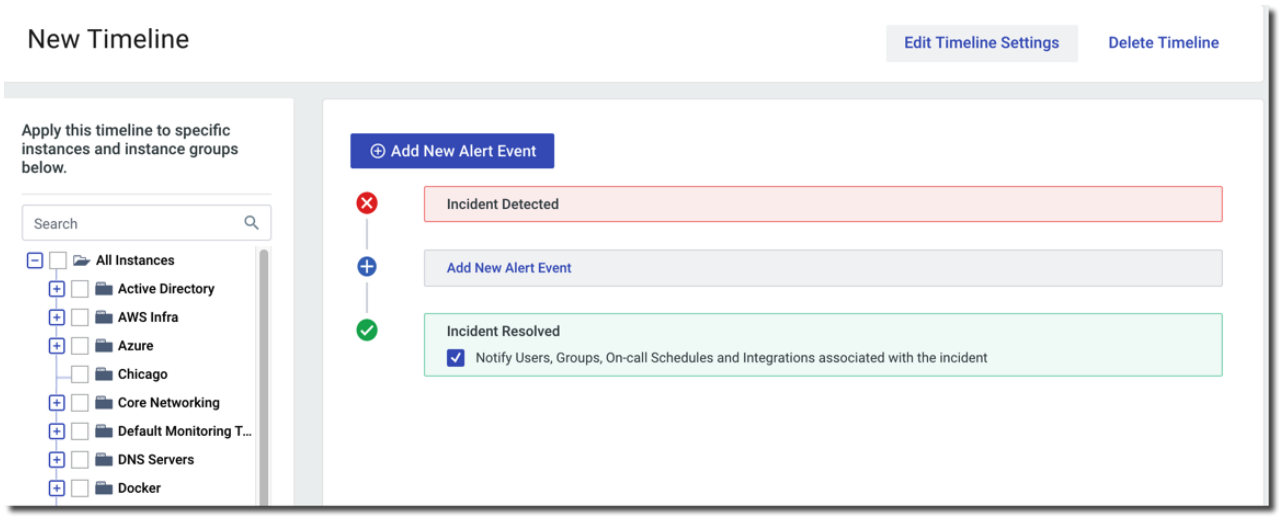
Create an Alert Timeline

To create an Alert Timeline, perform the following steps:

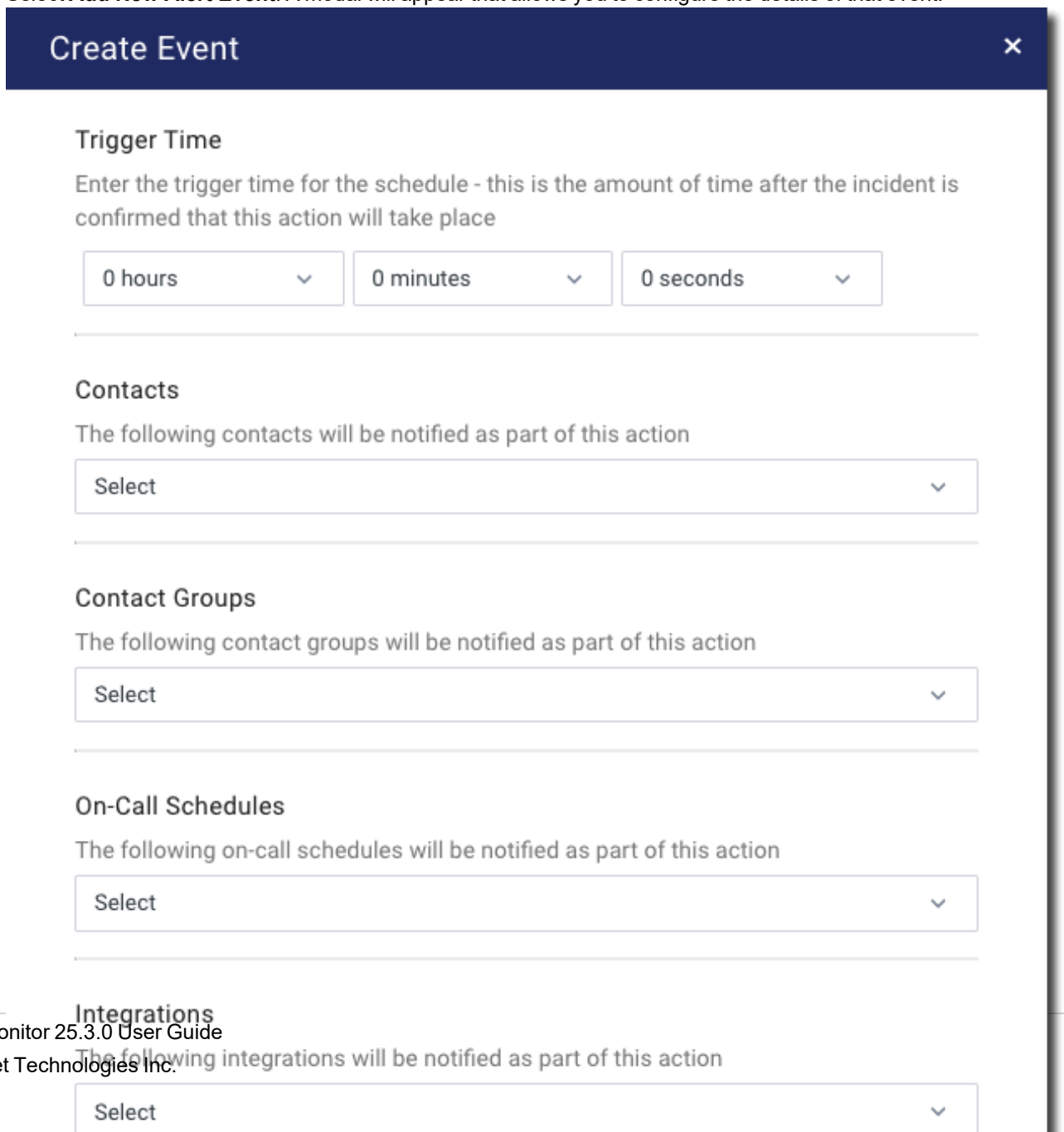
1. From the navigation menu, select **Add** > **Alert Timeline**.
   Start by naming the timeline and giving it a description. You have the option to copy the timeline from another if you already have one created, this is helpful if you're looking to build off another for special cases.



2. You will then see a page with a blank timeline.

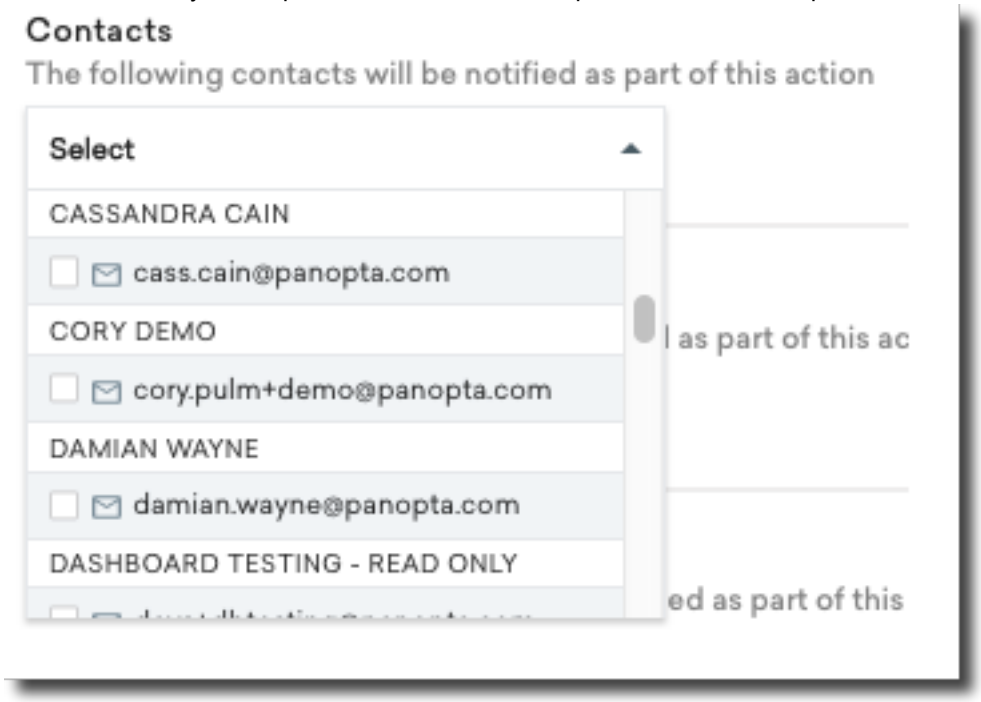


Select **Add New Alert Event**. A modal will appear that allows you to configure the details of that event.



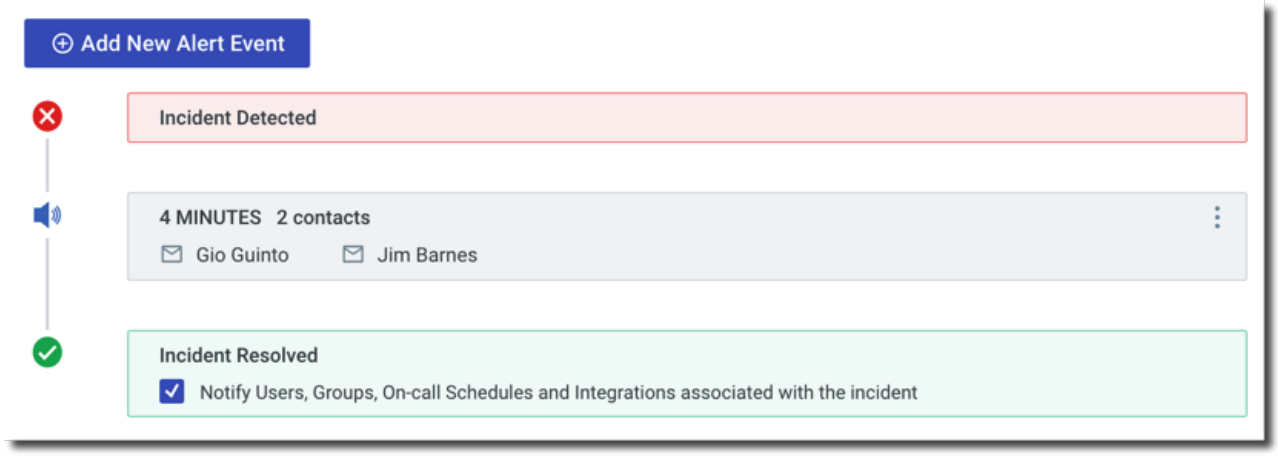
The first thing to decide is how long after an outage starts do you want to be notified. We generally suggest two minutes as the minimum to avoid getting notified about short-lived outages. We also suggest setting multiple events on timelines to act as an escalation in case the first alert goes unnoticed or is forgotten.

Next, pick some combination of users, groups, on-call schedules, and integrations to add to that event. Because you can define multiple events for escalations, you can pick the appropriate level of noise for each event. For individual users, you can pick between email, SMS, phone call, or mobile push notification.



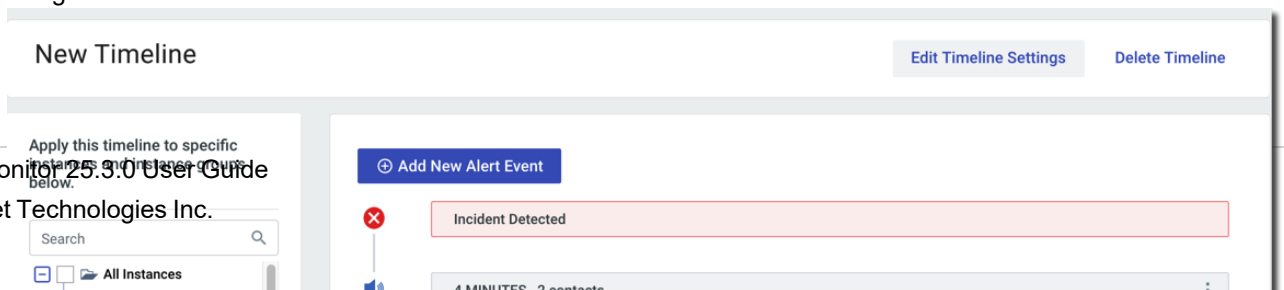
4.

The first alert on your timeline may look something like this.

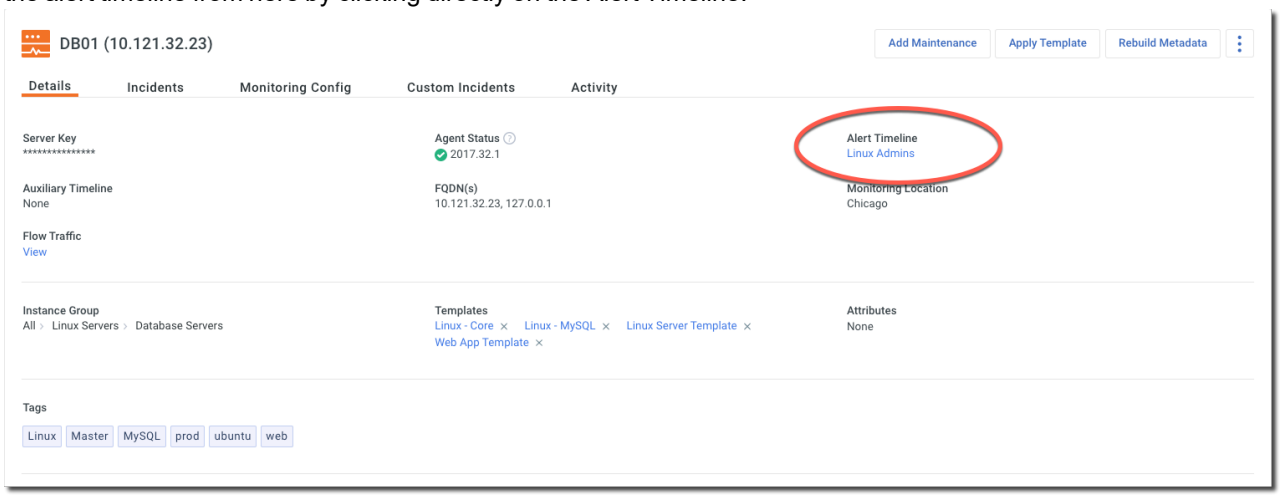


5. At this point, you can build escalation into your alert timeline by adding another event with different alert types, perhaps something more aggressive than an email, like an SMS or voice call.

   On the left side of the page, you can see the infrastructure tree. From here, you can select the servers you'd like to assign this Alert Timeline to.



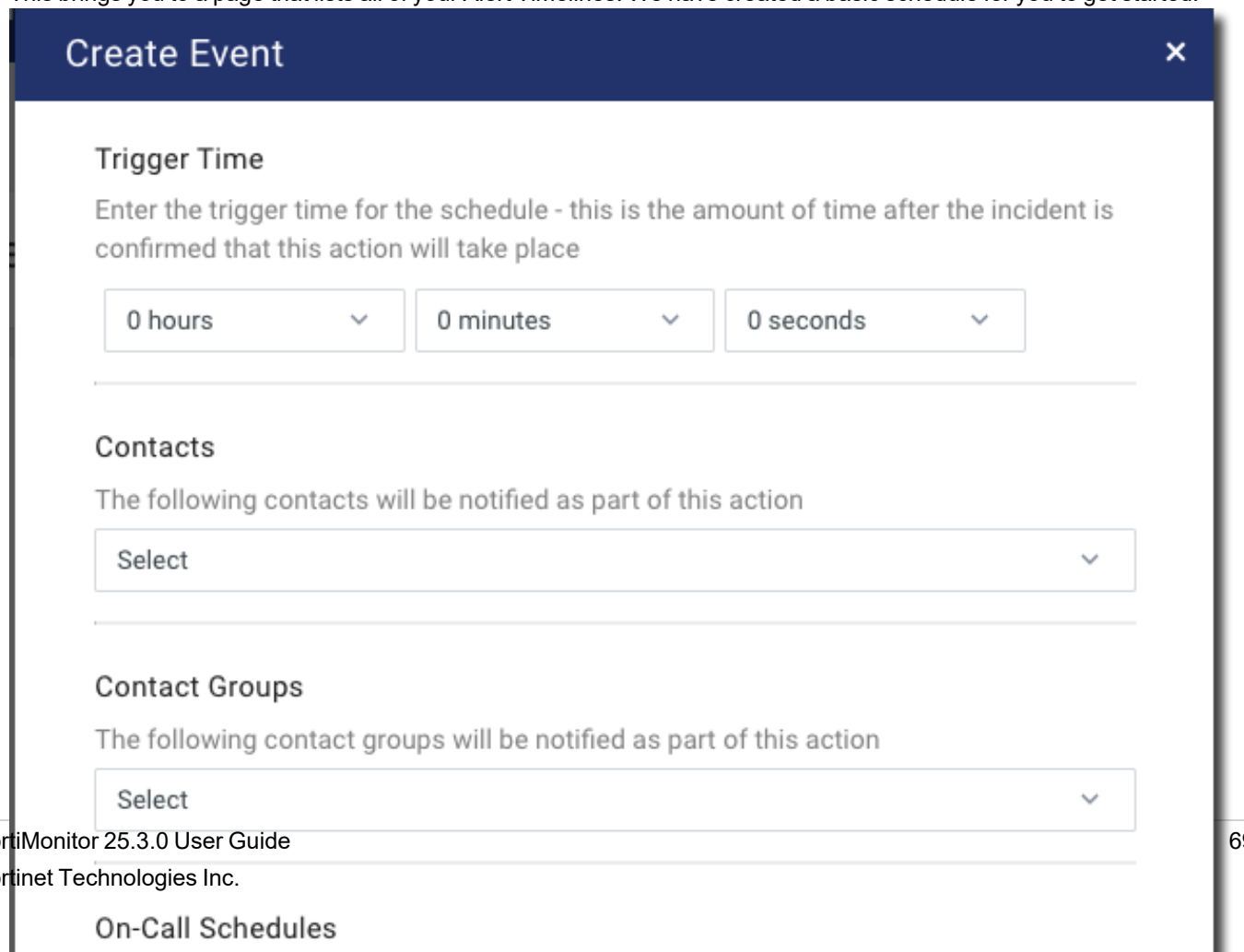
Select **Save Changes** after you select the servers.
You can now see the Alert Timeline you created on the **Details** tab of the **Instance Details** page. You can also edit the alert timeline from here by clicking directly on the Alert Timeline.

7. 

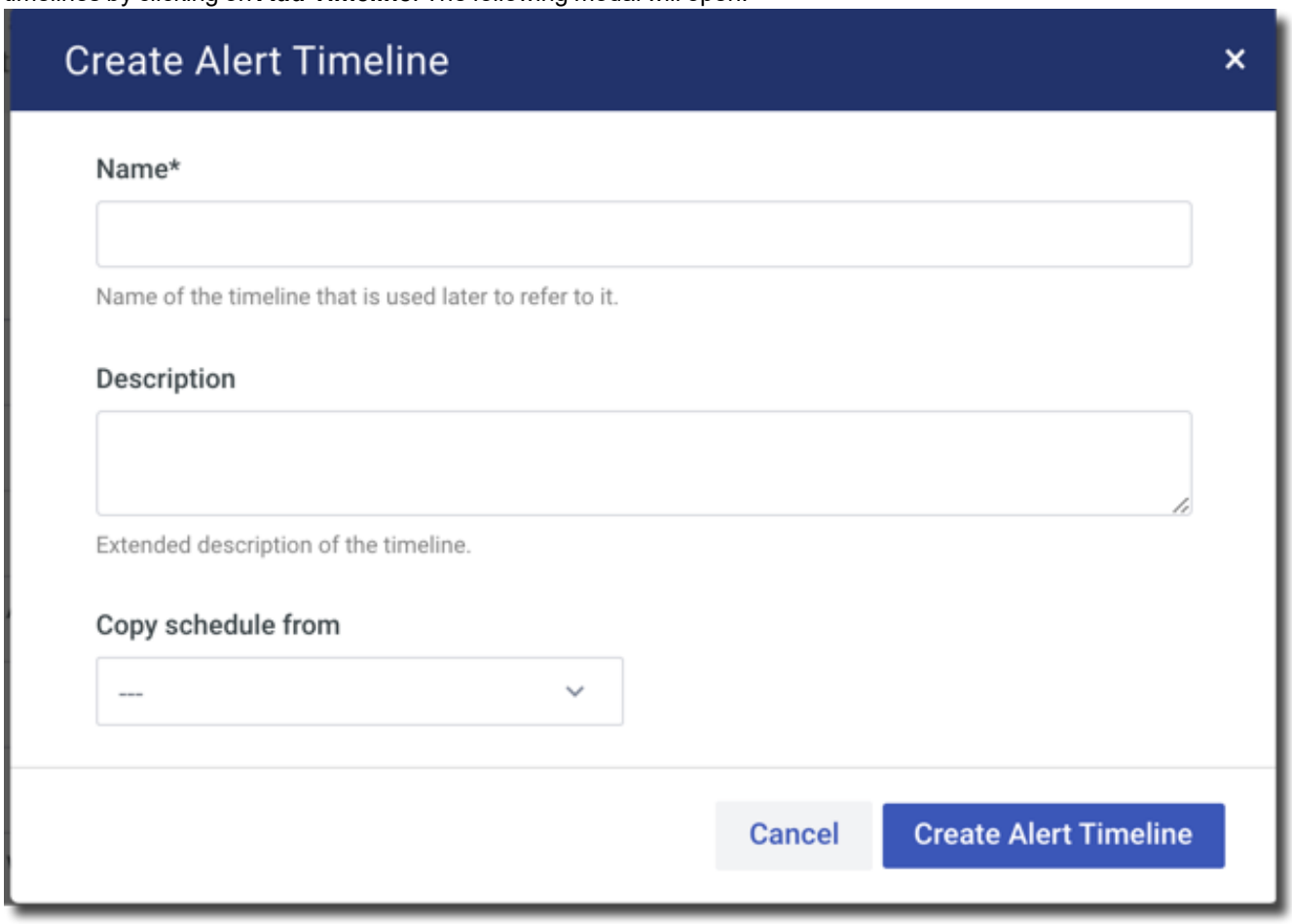
Configure who gets alerted during an outage

Alert timelines control who is alerted and when after an outage is detected. You should only create an Alert Timeline after configuring your users and contacts. If you have not done so, see this article for help. To get started, select **Incidents** > **Alert Timelines** from the navigation menu.

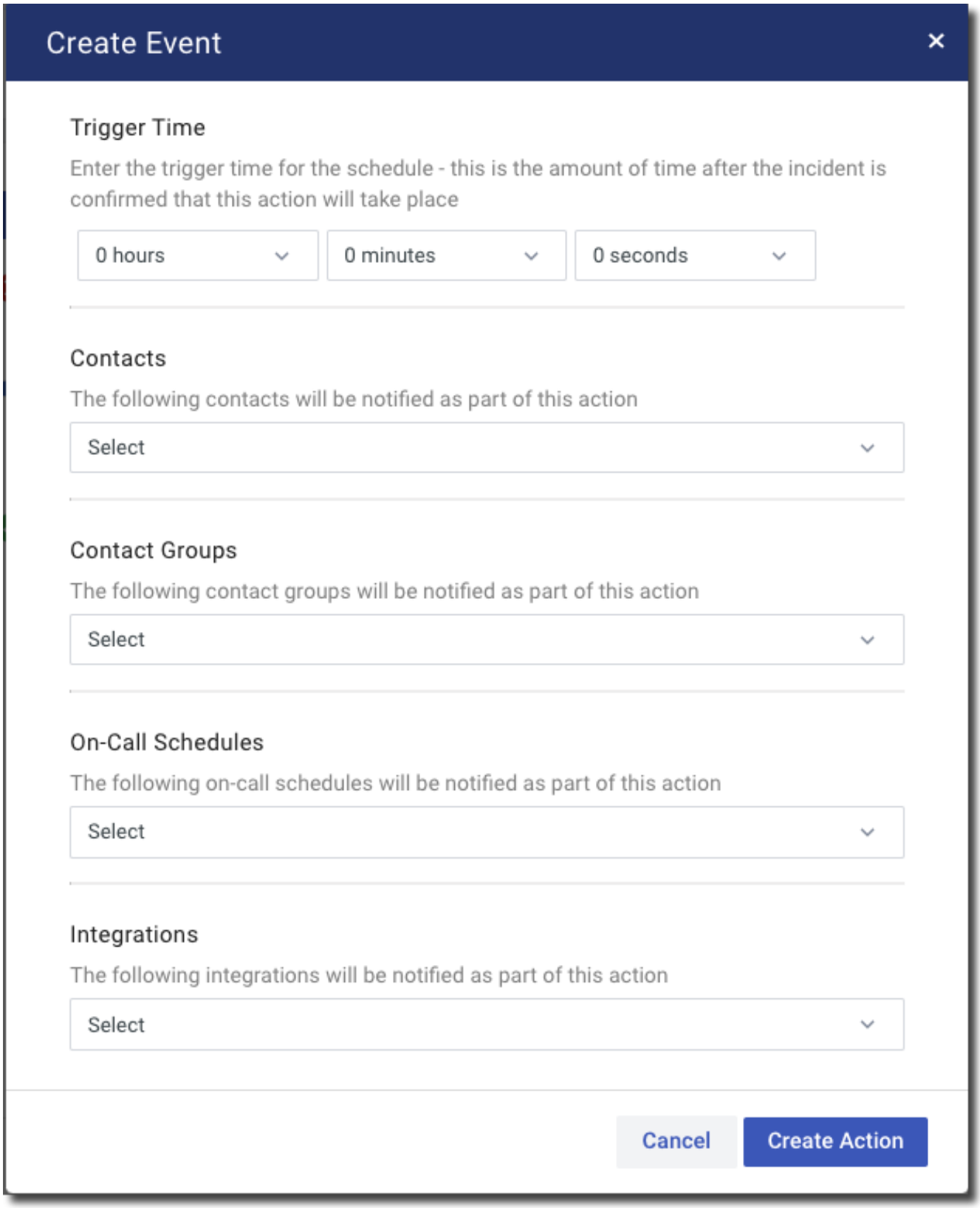
This brings you to a page that lists all of your Alert Timelines. We have created a basic schedule for you to get started.



You can either edit the timeline by clicking on the **3-dot menu** at the end of the timeline name or create additional timelines by clicking on **Add Timeline**. The following modal will open:



There are two major components of an Alert Timeline: the timeline of actions that will be taken and the instance or instance groups that the schedule applies to. To begin configuring the timeline click **Add New Alert Event**. A modal where you can choose who gets contacted and after how long will pop up.


Create Event
Trigger Time
Enter the trigger time for the schedule - this is the amount of time after the incident is confirmed that this action will take place
0 hours
0 minutes
0 seconds
Contacts
The following contacts will be notified as part of this action
Select
Contact Groups
The following contact groups will be notified as part of this action
Select
On-Call Schedules
The following on-call schedules will be notified as part of this action
Select
Integrations
The following integrations will be notified as part of this action
Select
Cancel
Create Action

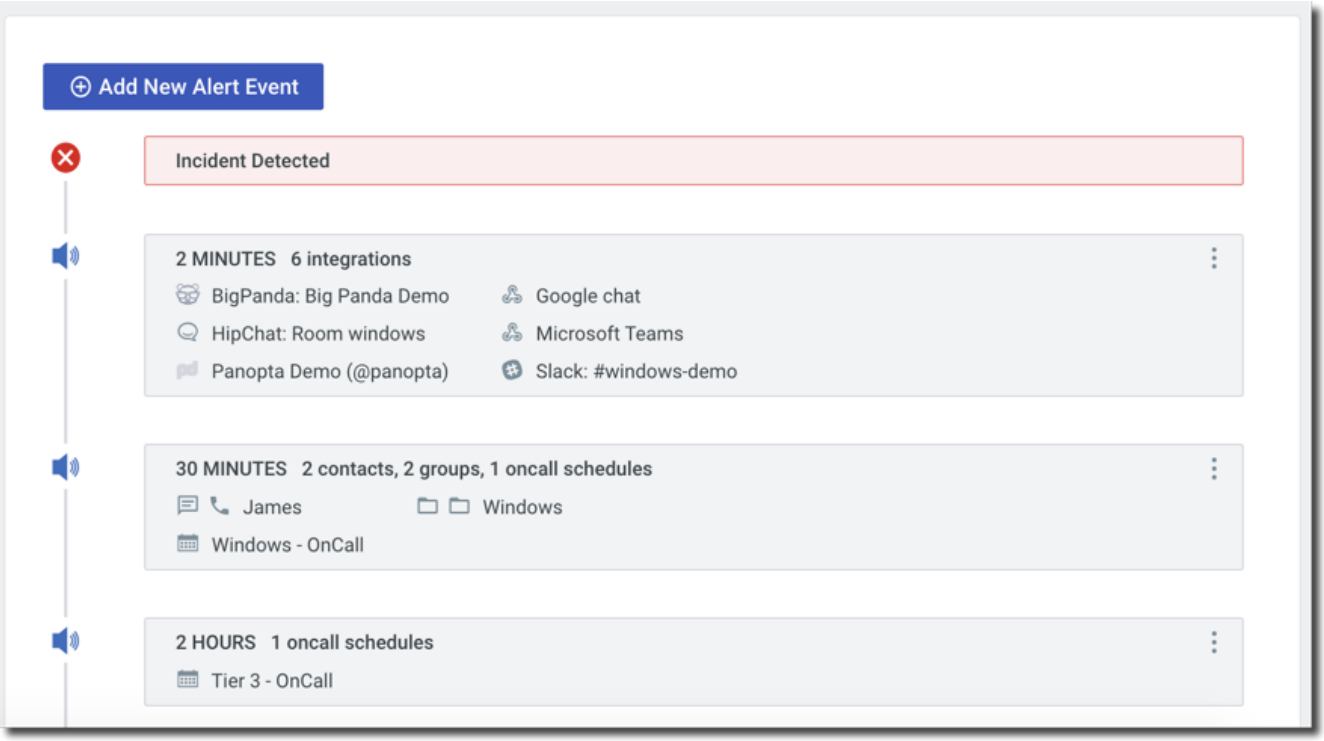
You can also click on any of events in the timeline to edit the contacts associated with that action. To add an instance or instance group, simply check any servers you would like from the list on the left side of the page then click **Save**. Once this is done, any outage that is confirmed for one of these servers will begin notification using this timeline.

Change the amount of time between an outage happening and the first notification

This time lapse is called the **Trigger Time** and you can adjust it however you like. By default, it is set to two minutes, so if you have an outage that lasts less than two minutes no alert message will be triggered. It should be noted that we do not recommend shortening your trigger time below two minutes in most cases, this will only result in a higher volume of alerts that are already resolved by the time you can respond to them. We would only recommend shortening the trigger time significantly on essential and reliable servers.

To change the trigger time:

1. Select **Incidents** > **Alert Timelines** from the navigation menu.
2. Select the Alert Timeline that you want to edit. You should see something like this on your screen:



3. Click an event (gray box). A modal will open up where you can configure the trigger time. You can also add new notification events that will notify more people based on how long the outage has lasted. To do this click **Add New Alert Event.**

Respond to an incident and delay further alerts

When you receive an alert for an incident, you have the following options:

- **Acknowledge**: mark that incident with your username so your team knows that you are taking responsibility for it.
- **Escalate**: immediately send an alert based on the next event, overriding the configured amount of time in the Alert Timeline.
- **Delay**: postpone the next event on the Alert Timeline for a specified length of time.

Acknowledging an incident

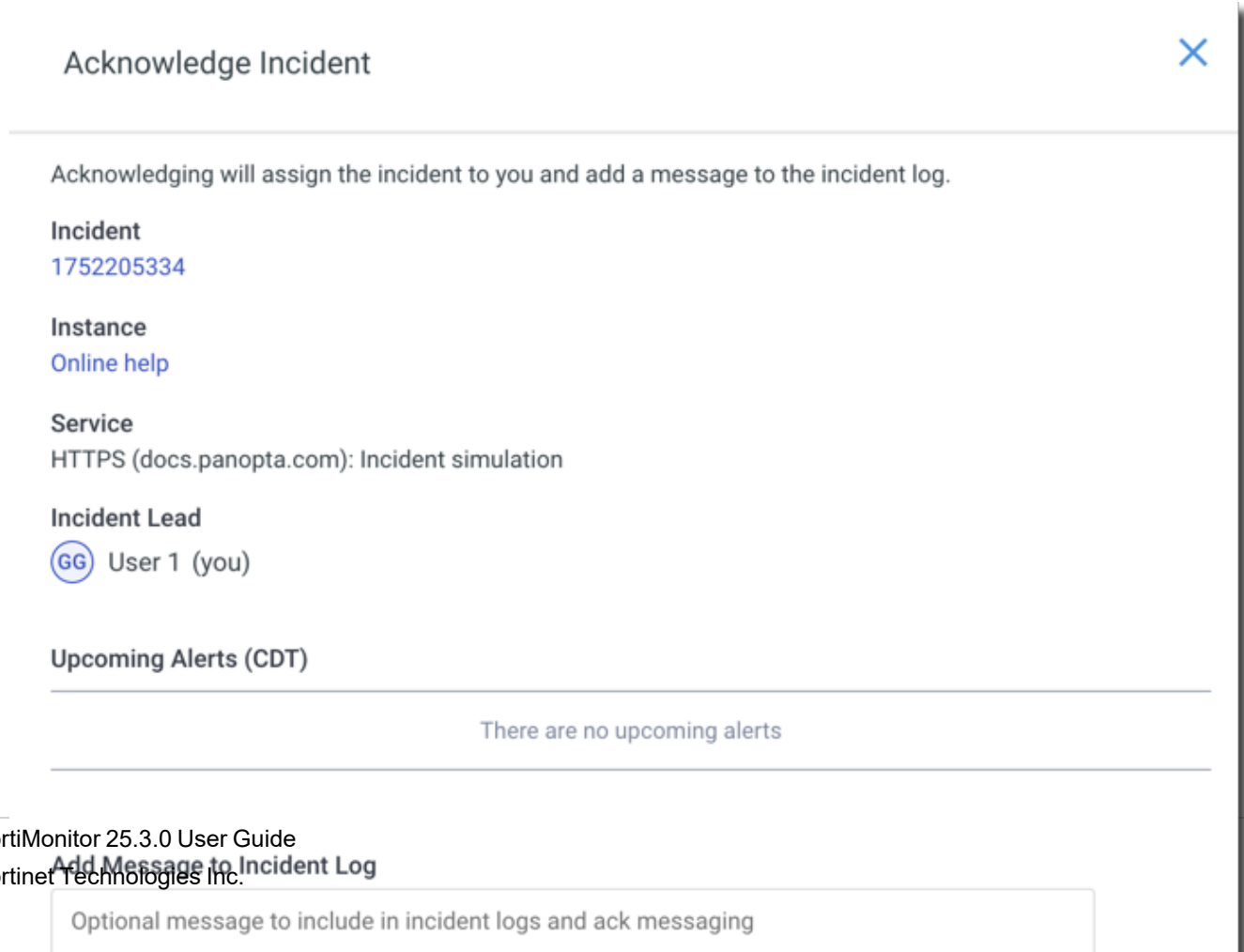
Acknowledging an incident marks the incident with your username and assigns you as the Incident Lead. You also have the option to send a message to all configured users and integrations in the Alert Timeline or delay further alerts for a set amount of time. There are several ways to acknowledge an incident:

- See **Incident Hub** and View incident details**.**
- From the **Infrastructure Map**, select the instance that has an incident then click **Take Action** > **Acknowledge**.
- From the **Instance Details** page, click the incident from the incident header. This will take you to the instance's incident page. You can acknowledge the incident from there. See View incident details**.**
- From the navigation menu, select **Incidents** > **All Incidents**. Select the incident then click **Acknowledge**.
- From the email you receive during an incident, select the **Acknowledge** link. This will take you to the control panel.

In addition to acknowledging an incident, the following actions are also available depending on which page you are on:

- **Broadcast** - Send a message to all configured users and channels in the Alert Timeline
- **Exclude from Availability** - Exclude the incident from the availability calculation
- **Delay alerts by** - When you acknowledge an incident, you have the option to postpone alerts for a set amount of time
- **Enable all clear** - this functionality alerts all previously alerted users/integrations that the incident has resolved
- **Cancel alerts** - cancels succeeding alerts. No further alerts will be sent
- **Escalate incident** - Immediately send an alert based on the next event, overriding the configured amount of time in the Alert Timeline. This means that even if you have configured the Alert Timeline to send an alert after 10 minutes, you can override this and send the alert earlier.

Once you click the **Acknowledge** button, the **Acknowledge Incident** drawer will slide out. From this drawer, you can choose to delay alerts, acknowledge the outage, and add a message to the incident's log.



Escalating an incident

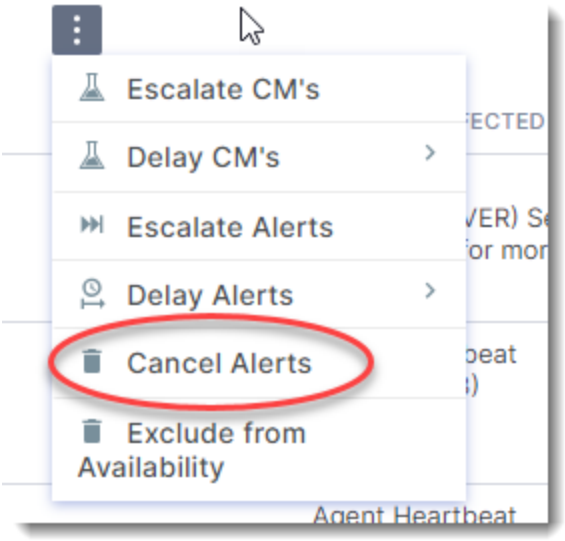
Escalating means you send an alert to the next-in-line on the Alert Timeline immediately. If you had an Alert Timeline set up to send an alert to someone 5 minutes after an incident is detected, and it was only three minutes into the incident, you can escalate it and instead of waiting 2 more minutes for the 5-minute alert to be sent, it would be triggered right away.

Delaying an incident alert

Say you have an Alert Timeline to have one person alerted as soon as the incident occurs, and another person 5 minutes into the incident. You can choose **Delay alert by 5 minutes** and the person who would normally get an alert at 5 minutes would not be alerted until 10 minutes into the outage. This is for all alerts on the Alert Timeline, so all alerts would be pushed back by 5 minutes. It only adds 5 minutes to the original time, not the time you acknowledged it.

Prevent active outages from creating alerts

You can silence your current outages by going to **Incident** > **All Incidents**. Select the incidents that you want to silence then click **3-dot menu** > **Cancel Alerts**.



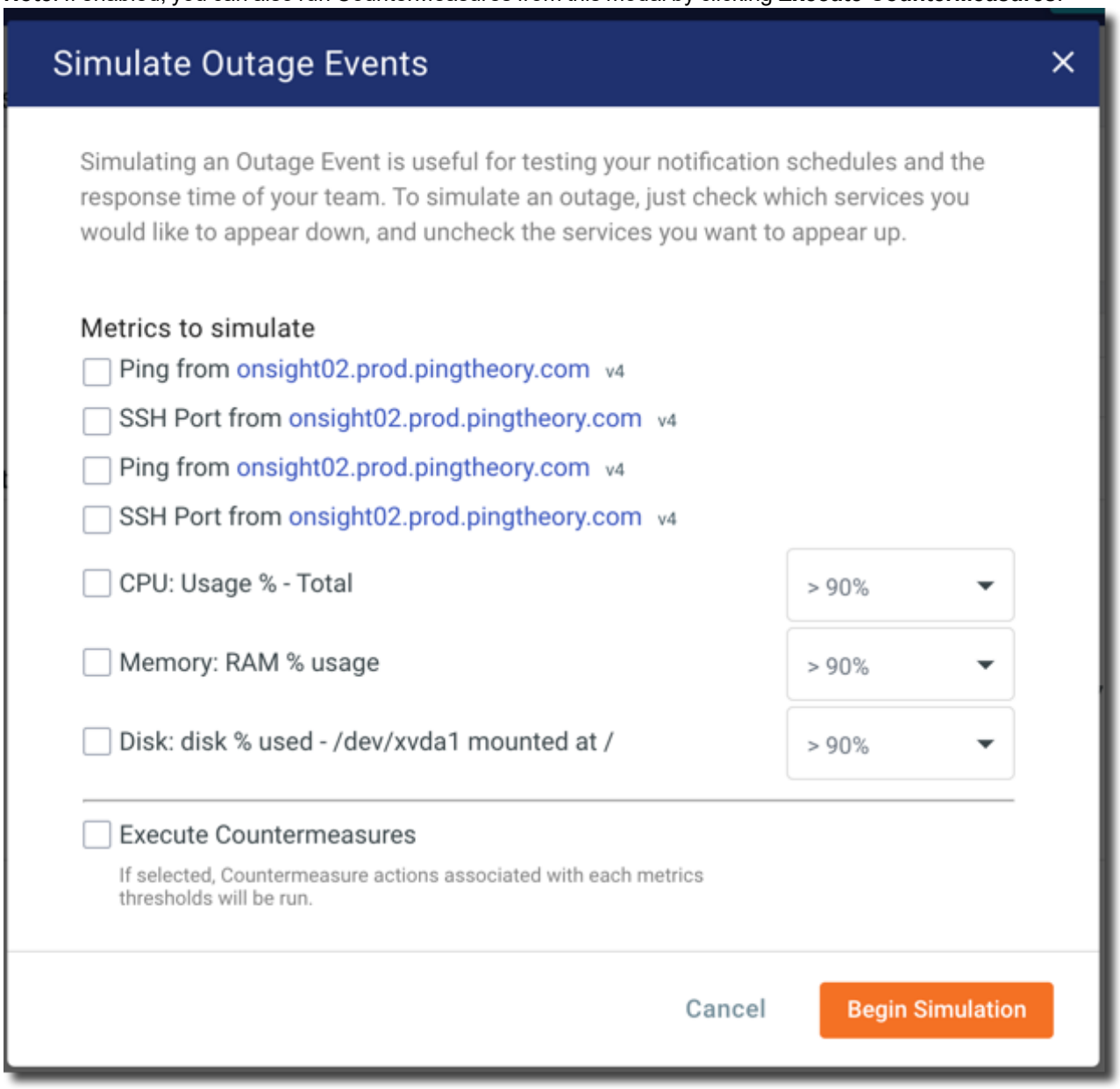
This action will prevent these outages from sending out any additional alerts.

Simulate an outage

Outage simulations are a great way to check that your alerts are set up just as you would like. If you choose to start an outage simulation on a server or a compound service, then FortiMonitor will act as if that server is down. Whoever you have assigned to be notified, will be. The outage will also show up in the control panel but it will not affect your uptime calculations.

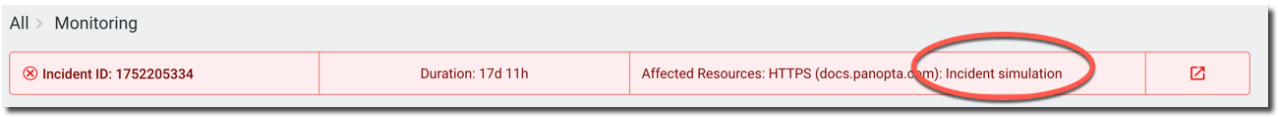
Start an outage simulation

To start an outage simulation, perform the following steps:

1. From the **Instance Details page** of the instance where you want to start the outage simulation, click the **Incidents tab** then select **Incident Simulator**.
   From the modal that appeared, select the service or services that you would like to go down then click **Begin Simulation**.
   Note: If enabled, you can also run CounterMeasures from this modal by clicking **Execute CounterMeasures**.
2. 

   When the outage simulator begins it will not actually affect the server itself, however, this will affect the data collection for that server. While the outage itself will be correctly labeled as just a simulation, any actual outages that occur during the simulation will not be recorded. Simply it is best not to forget to turn the simulation off when you are done with it, lest you miss an actual outage.

   Instance Details page incident simulation label



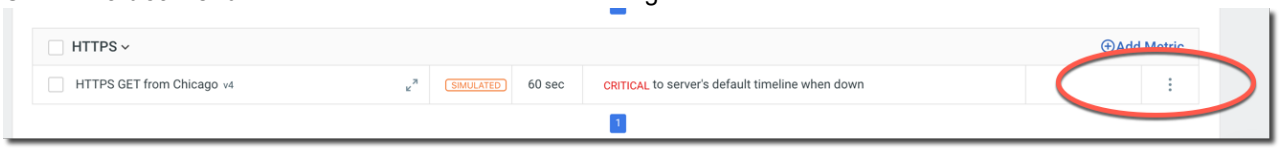
Incident Hub incident simulation label



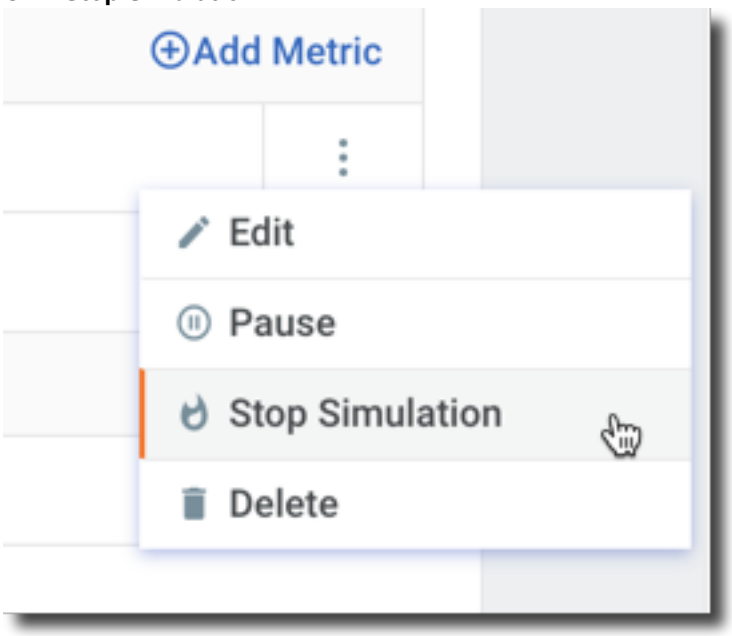
Stop an outage simulation

To stop the outage simulation, perform the following steps:

1. Go the **Monitoring Config** tab of the **Instance Details** page where you started the outage simulation.
   Click the **3-dot menu** of the metric with the simulated outage.



2. Click **Stop Simulation**.

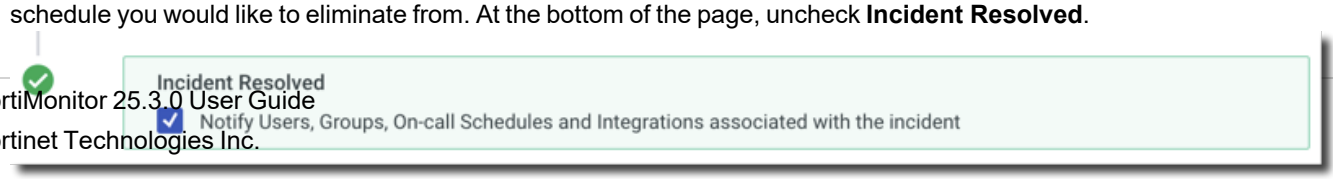


3.

Disable Incident Resolved alerts

Once an incident is resolved, **Incident Resolved** alerts are sent out to everyone that was previously alerted for the incident. If you trust your workers to handle an incident, or you are handling the incident yourself, then you may not feel you need these alerts. You can disable them, and there are two basic ways of doing this based on what you are trying to accomplish.

If you would like to disable the alert, select Alert Timelines from the Monitoring drop-down. From here select the schedule you would like to eliminate from. At the bottom of the page, uncheck **Incident Resolved**.

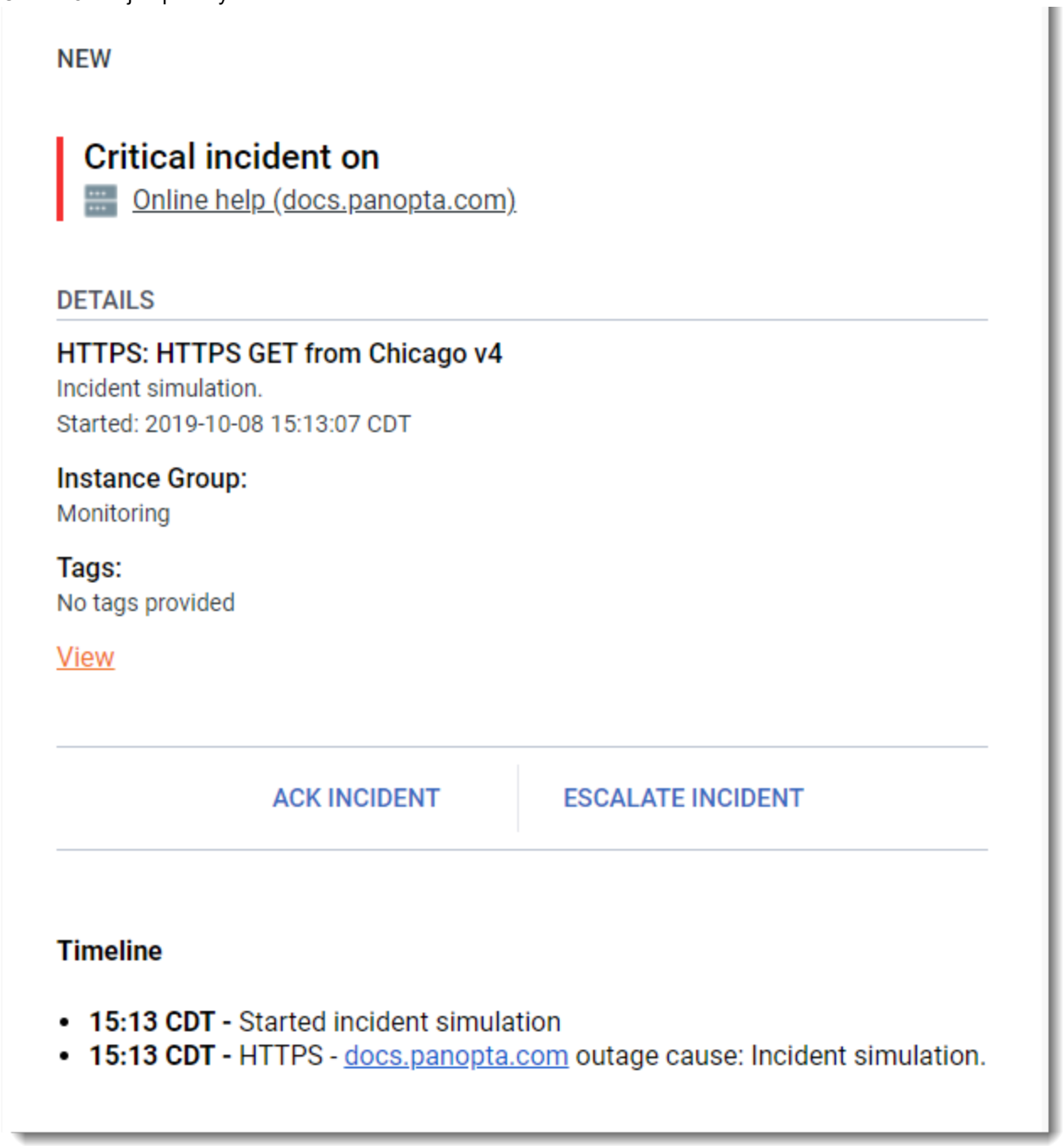


Email

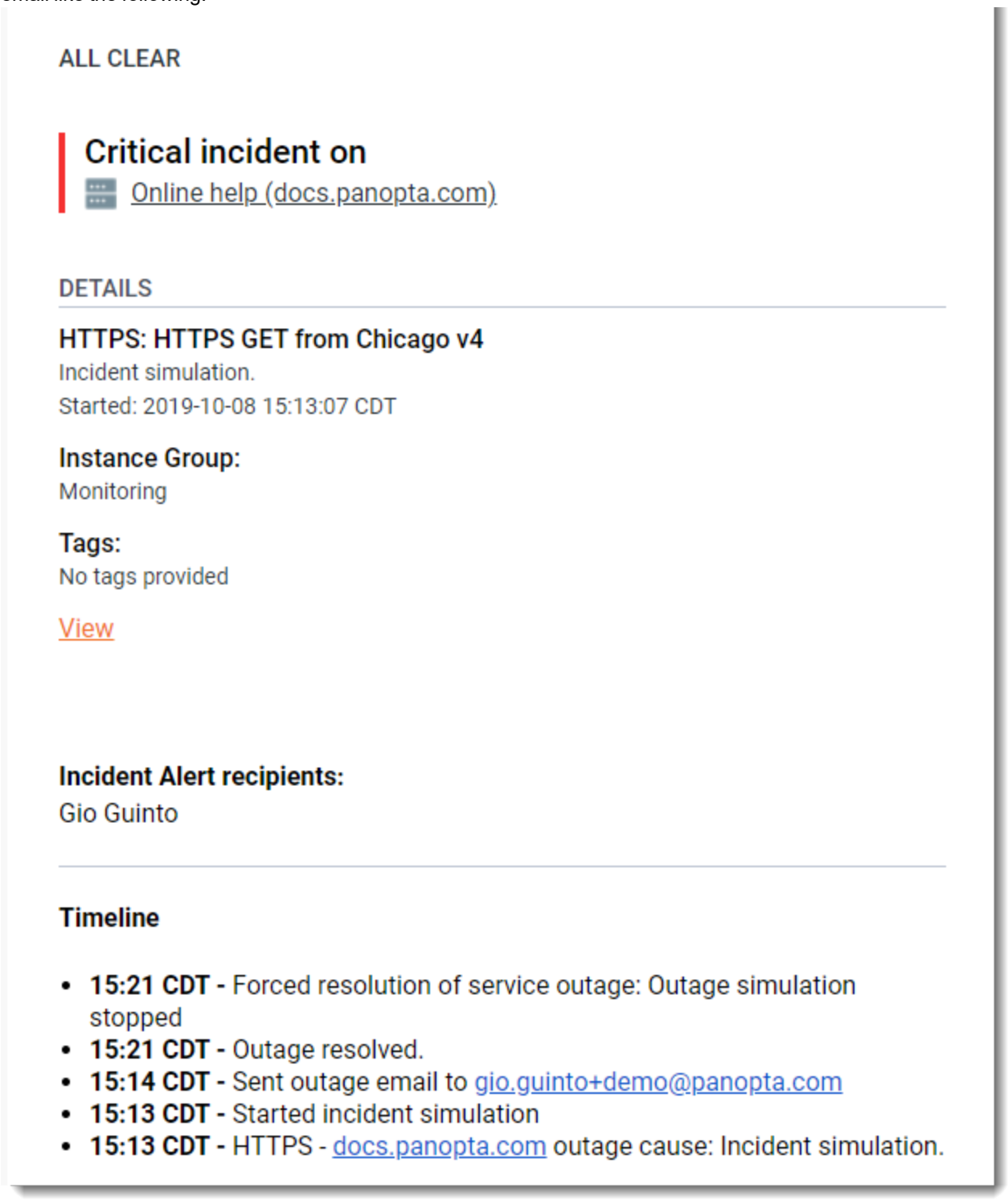
One of the options for receiving alerts after an incident on your infrastructure is an email alert. The FortiMonitor incident alert emails are divided into two different sections. The first identifies the affected instance. The second allows you to take action by either viewing the incident, acknowledging the incident, or escalating the incident to the next contacts on the alert timeline.

You can see the affected service, the group the instance belongs to, the severity, and when the incident started in the first part of the email. The **Timeline** section shows you when and where the incident originated.

Click **View** to jump into your account and see more incident information.



Once the incident has cleared, if you have enabled **All Clear** in the Alert Timeline for the instance, you will receive an email like the following.



ALL CLEAR

Critical incident on
Online help (docs.panopta.com)

DETAILS

HTTPS: HTTPS GET from Chicago v4
Incident simulation.
Started: 2019-10-08 15:13:07 CDT

Instance Group:
Monitoring

Tags:
No tags provided

View

Incident Alert recipients:
Gio Guinto

Timeline

- **15:21 CDT -** Forced resolution of service outage: Outage simulation stopped
- **15:21 CDT -** Outage resolved.
- **15:14 CDT -** Sent outage email to gio.guinto+demo@panopta.com
- **15:13 CDT -** Started incident simulation
- **15:13 CDT -** HTTPS - docs.panopta.com outage cause: Incident simulation.

You can see the actions taken, the time the incident resolved, and who was contacted according to the assigned Alert Timeline.

In situations with users who are contact only, none of the links will be active, they will only receive the alert. They will not see the options to **View**, **Ack**, **Delay**, or **Escalate**.

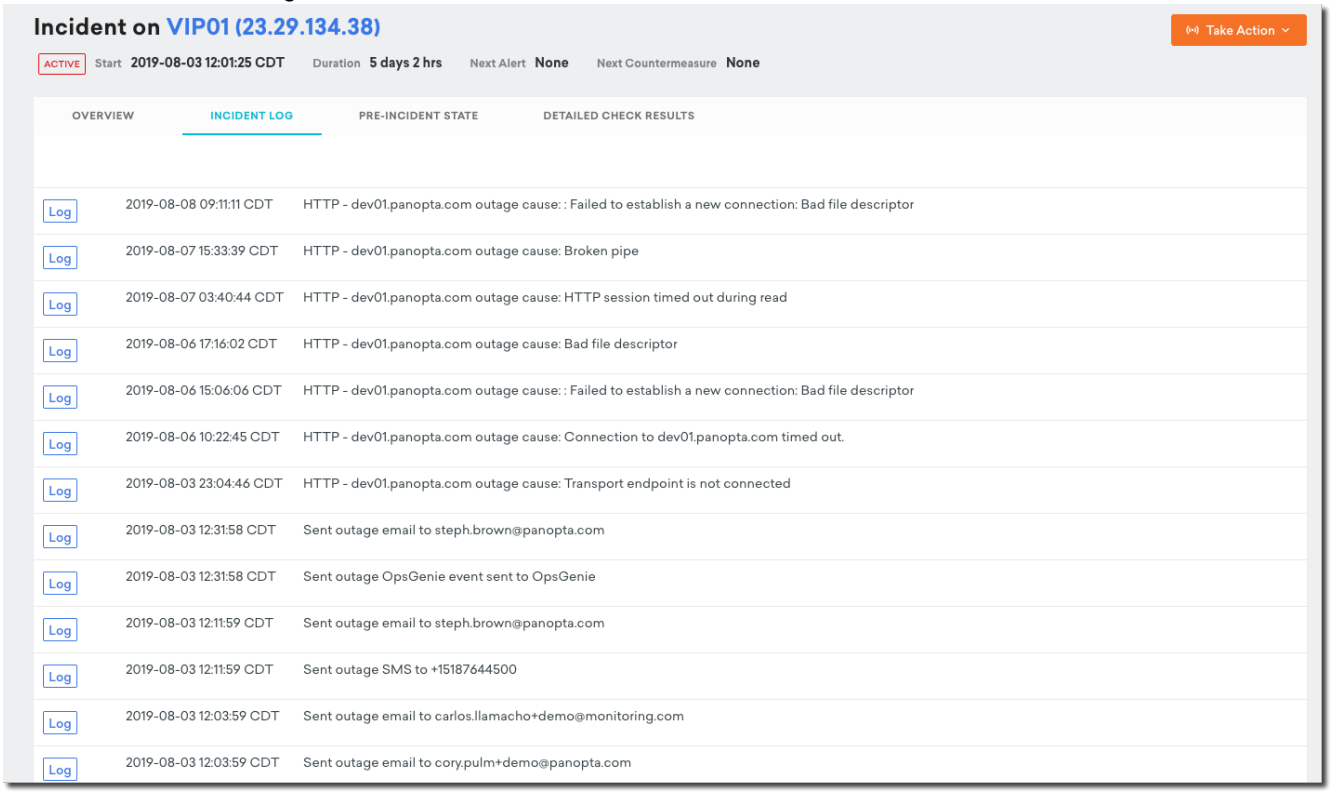
SMS

Troubleshooting SMS alerts

This article describes the reasons why you may not be receiving SMS alerts.

The first step for any SMS troubleshooting is to double check the country code and phone number you provided to the FortiMonitor website. It may sound silly, but a simple typo could be the cause of all your problems, so do not forget to double-check this first.

Your second step should be to check your incident logs. To do this, select **Incidents**> **All Incidents** from the navigation menu. Find the incident you were expecting SMS alerts from and click on its Event ID, this will take you to a new page that summarizes the outage.



Select the **Incident Log** tab to see a detailed description of the incident including which alerts were sent out. If there was no SMS alert sent, read the next paragraph. However if the logs state that the SMS was sent and was not received, please contact our support team. If you do, please remember to include your phone number and mobile carrier.

The most common problem is a simple one, it is a configuration problem. If you have not received an SMS for an outage, despite expecting one, first check the duration of said outage and the trigger time for the Alert Timeline that covers this server. If the outage lasted less time than the trigger time on your Alert Timeline no messages were sent out because the Outage was resolved before an alert was sent. You can shorten your trigger time if you would like to be notified for even

very short outages. It should be noted that we do not recommend shortening your trigger time in most cases, this will only result in a higher volume of alerts that are already resolved by the time you can respond to them.

If your configuration is set up correctly and there are still missing SMS messages then you should contact our support team. We will need your phone number and the name of your mobile provider and we will contact your SMS gateway to set up a test message.

Unfortunately no SMS gateways guarantee delivery 100% of the time, but we can change you to a gateway better suited for your mobile carrier if that will ensure better service.

SMS and voice numbers for alerts

FortiMonitor uses several gateways to send voice and SMS alerts. If you want to set up custom ringtones or Do Not Disturb behavior for alerts, use the following numbers.

US/Canada

Voice

- 312-234-0186 (Main)
- 312-757-5683 (Backup)

SMS

- 312-234-0186
- 312-257-2526
- 67082
- 41782

Rest of the world

Voice and SMS

- 41782
- 67082
- 442033223696 (UK SMS only)

Which wireless carriers are compatible with FortiMonitor alerts?

FortiMonitor works with multiple gateways supporting over 700 mobile networks around the world. Unsure if your carrier is supported? Please contact our support team.

Integrations

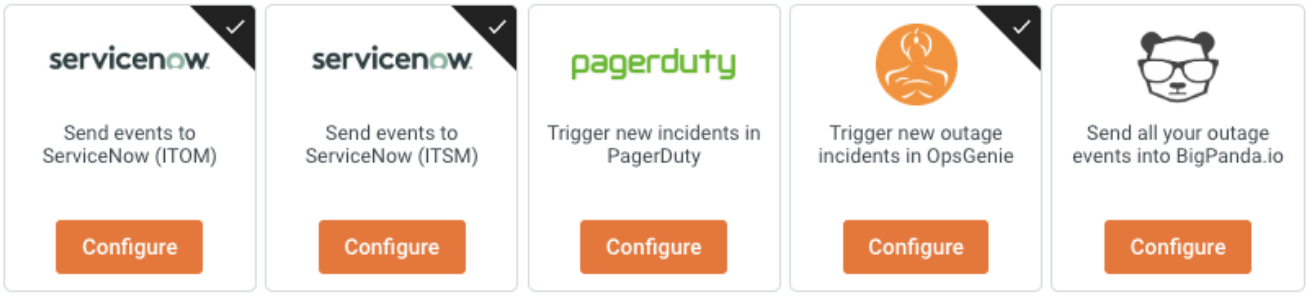
We are constantly adding new services to help you and your team get notifications in the most seamless way possible. Go through the following sections to learn more.

ServiceNow

FortiMonitor's ServiceNow integration enables you to send incident and clear events to your ServiceNow instance.

There are two ServiceNow integration types you may utilize, depending on your ServiceNow workflow. If you want incidents to be sent as events, using the ITOM functionality, select the generic **ServiceNow ITOM** integration. If you want FortiMonitor to directly create incidents in your incident table, select the **ServiceNow ITSM** integration. Both integrations require the same parameters.

The integration can be set up via the **Integrations** page, which is found under the **Teams & Activity**menu item.



Configuration

ServiceNow user roles and permissions

To integrate ServiceNow with FortiMonitor, the ServiceNow user account that you will use to authenticate with FortiMonitor should have the proper permissions. A good starting point is the ServiceNow base system role **ITIL**. The **ITIL** role has the necessary roles assigned to it that will allow FortiMonitor to send events to your ServiceNow environment.

If you have custom roles, the **ITIL** role can still be used to see which permissions are required.

For more information on the **ITIL** role, refer to the ServiceNow documentation.

1. From the main navigation header. select **Settings** > **Integrations**.
2. Locate the **ServiceNow** card and click **Configure**.
3. Within the configuration modal, you'll need to configure the following items:

- **Title:** the name of the integration as it will appear on Alert Timelines
- **ServiceNow Instance Name:** the name of your ServiceNow installation. Typically the subdomain portion of your login URL. For example, the instance name for **FortiMonitor.service-now.com** would be **FortiMonitor**.
- **Username:** the username that FortiMonitor should utilize to authenticate when sending the incident details
- **Password:** the ServiceNow password that FortiMonitor should utilize to authenticate when sending the incident details

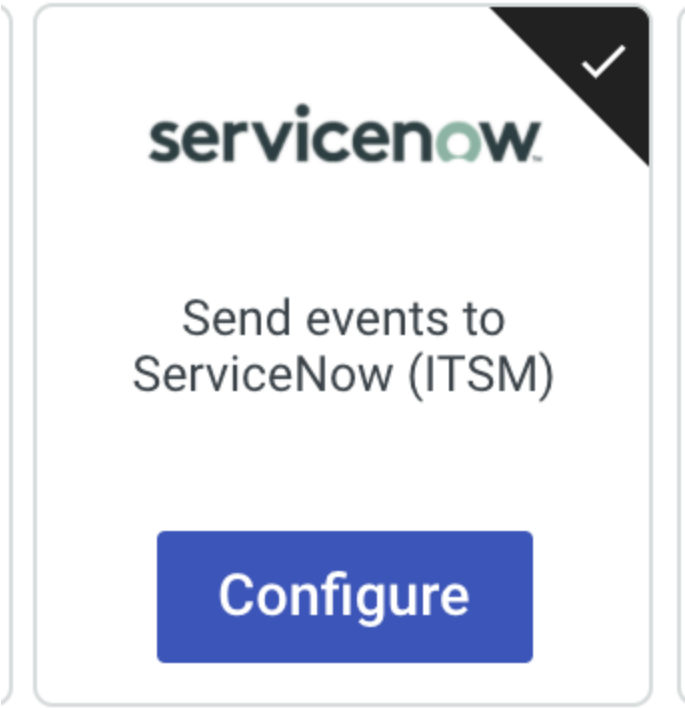
4. Upon saving, you should now see your ServiceNow integration listed in the **Configured Integrations** table

Now that your ServiceNow integration is configured, you can add it to any of your Alert Timelines so that ServiceNow receives Incident events accordingly.

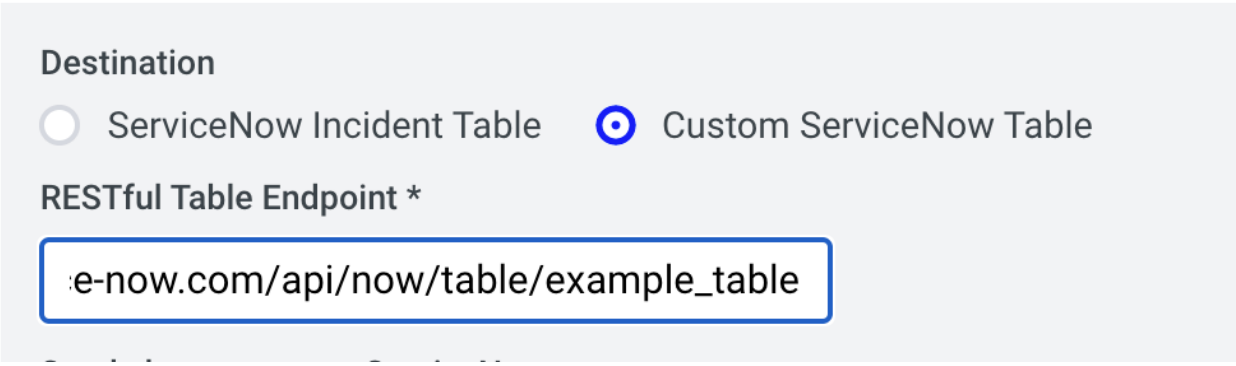
ServiceNow endpoint override

To utilize a staging table in ServiceNow instead of inserting FortiMonitor incidents directly into the SNOW Incident table, we've introduced the ability to POST to a manually-specified endpoint. You can see an example below.

1. Ensure that you've defined your table with the desired columns in ServiceNow.
2. Obtain the **endpoint URL** of your table.
3. In the FortiMonitor Control Panel, go to **Settings** > **Integrations**.
4. Select the **ServiceNow ITSM** card.

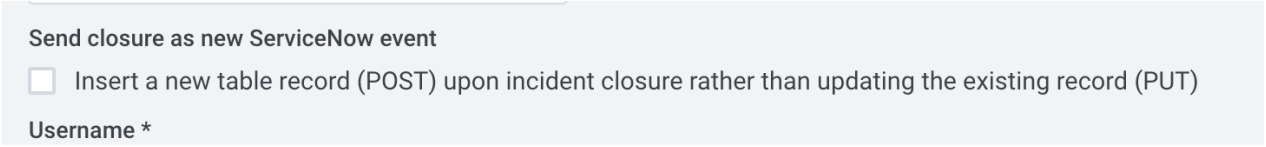


5. Provide a name for the integration.
6. Under **Destination**, select **Custom ServiceNow Table**.
   In the field that appears, enter the **endpoint URL** for your table from step 2.



7.
8. If you want a *new* record created when the FortiMonitor incident resolves, select the **Send closure as new ServiceNow event** option. If you leave it unchecked, upon incident closure, we'll make a PUT call on the previously

created record in your custom table.



9. Enter the appropriate username and password.
10. We provide a basic payload example but you can edit it as needed to match your table schema - it just needs to be valid JSON. As you update the payload, an example will populate below using data from a recent incident on your account.
11. Click **Save**.
    We recommend testing the integration with our incident simulation functionality.

Parameters

The following template parameters may be used in the JSON Payload fields.

Parameter	Description
$alert_label	Alert label of the incident/anomaly.
$compound_service_id	The ID number of the compound metric affected.
$custom_attribute	You can pass custom server attributes that are set on your servers. Use the attribute type as the key.
$duration	The duration of incidents/anomalies which will be filled in on-clear.
$event	The type of event, either **incident event** or **clear event**.
$fqdn	The Fully qualified domain name of the server experiencing the incident/clear.
$incident_summary	The summary of the incident.
$incident_tags	The tags for the incident.
$incident_timeline	The entire timeline output for the incident.
$item_type	The service type textkeys of the services experiencing the incident/clear, or plugin_textkey/resource_textkey combinations of the resources the experiencing the anomaly/clear, or the OID name of SNMP resources experiencing the outage/clear.
$items	Services experiencing the incident/clear or resources experiencing the anomaly/clear.
$location	A comma-separated list of the primary monitoring probe names for all network services affected.
$metric_tags	The tags for all of the metrics involved in the outage.
$name	Name of the server experiencing the incident/clear.
$network_service_id	The ID number of the network service affected.
$outage_id	The ID number of the associated incident.

$partner_server_id	The partner server id for the server.
$reasons	The reasons for network service incidents or the details for anomalies.
$resource	For resource anomalies: resources experiencing the anomaly/clear.
$server_id	The ID number of the server experiencing the incident/clear.
$server_key	The server key for the server.
$server_resource_id	The ID number corresponding to the resource affected.
$services	For service incident: services experiencing the incident/clear.
$severity	The severity of the outage/anomaly, either **critical** or **warning**.
$tags	The tags for the server.
$severity_number	The severity of the incident. • **1** for a critical incident • **2** for a warning
$timestamp	UTC timestamp of when the incident/clear occurred.
$trigger	The type of event that triggered this payload (outage, ack, broadcast, clear).
Server Attributes	$account_id $alertmanager $app $app_kubernetes_io/instance $app_kubernetes_io/managed-by $app_kubernetes_io/name $aws_az $aws_image_description $aws_image_id $aws_instance_id $aws_instance_type $beta_kubernetes_io/arch $beta_kubernetes_io/instance-type $beta_kubernetes_io/os $chart $cloud_aws_account_id $cloud_aws_az $cloud_aws_image_id $cloud_aws_instance_id $cloud_aws_instance_size $cloud_aws_os $cloud_aws_platform $cloud_aws_region $cloud_aws_service

$cloud_azure_az
$cloud_azure_instance_size
$cloud_azure_os
$cloud_azure_region
$cloud_azure_service
$cloud_provider
$cloud_provider
$cluster
$company_name
$component
$container_id
$container_image
$container_name
$container_platform
$controller-revision-hash
$database_engine
$database_version
$doks_digitalocean_com/node-id
$doks_digitalocean_com/node-pool
$doks_digitalocean_com/node-pool-id
$doks_digitalocean_com/version
$environment
$failure-domain_beta_kubernetes_io/region
$helm_sh/chart
$heritage
$io_cilium/app
$joblabel
$k8s-app
$kubernetes kind
$kubernetes_cluster_name
$kubernetes_io/arch
$kubernetes_io/cluster-service
$kubernetes_io/hostname
$kubernetes_io/name
$kubernetes_io/os
$kubernetes_kind
$kubernetes_namespace
$kubernetes_node
$kubernetes_pod_name
$launch_time
$location
$my_attribute

$name
$namespace
$ncm_unimus_activated
$ncm_unimus_deactivated
$node
$operated-alertmanager
$operated-prometheus
$os
$os
$platform
$pod
$pod-template-generation
$pod-template-hash
$prometheus
$provider
$region
$release
$role
$run
$server_cpu_architecture
$server_cpu_core_count
$server_instance_source
$server_kernel_version
$server_location
$server_origin
$server_os
$server_os_distro
$server_os_distro_version
$service_id
$snmp_model
$snmp_serial_num
$snmp_syscontact
$snmp_sysdescr
$snmp_syslocation
$snmp_sysname
$snmp_sysobjectid
$snmp_vendor
$snmp_device_model
$snmp_device_type
$snmp_sysdescr
$statefulset_kubernetes_io/pod-name
$testing

$tier
$ui_performance_tabs
$username
$version
$vmware_hardware_uuid
$vmware_mac_address
$vmware_moref
$vmware_uuid
$vm_id

Slack

Installing FortiMonitor's Slack application in your workspace ensures your team receives real-time FortiMonitor incident notifications. As well, you can respond to incidents alerts by Acknowledging the incident or Delaying, Escalating, and Cancelling future notifications.

Configuring the Integration

1. Visit the integrations page by going to **Teams & Activity > Integrations**
2. On the **Slack** card, click **Configure**
3. If this is your first Slack integration, you'll see a single input. Enter the Slack channel you'd like to receive the notifications.
   3.1 If you've previously configured a Slack integration and just want to add a new channel, you can choose to use an existing authorized integration or add a new one. Either way, you'll enter the new Slack channel you'd like to receive the notifications
4. Hit **Done**. You'll be taken to Slack where you'll be asked to authorize FortiMonitor's Slack app.
5. After authorization, you'll arrive back at the FortiMonitor control panel. The integration is now complete. To begin receiving FortiMonitor notifications in your Slack channel, make sure to add your new Slack integration onto your Alert Timelines.

FortiMonitor Bot Permissions

If you are leveraging CounterMeasures remediation and are sending the output to a private slack channel, you will have to further enable the FortiMonitor bot to post the rich content to your channel by doing **/invite @FortiMonitor** in the channel.

User Authorization

One of the most powerful parts of the FortiMonitor Slack integration is the ability to respond to FortiMonitor incident notifications directly from Slack. To do that, you need to tie your Slack user to your FortiMonitor user. If you attempt to perform one of these actions before authenticating, the action will fail.

1. Make sure you've configured the FortiMonitor Slack app as described above
2. Open the **My Account** modal in FortiMonitor by clicking the user avatar in the top right corner of the page and then click **My Account**.
3. Under the **Slack Integration** module, click **Link Slack User**. The **Link Slack User** modal will appear.
4. Select your **Slack Team** and enter the display name associated with your Slack account then select **Proceed**.

5. You should have received an authentication code via direct message in Slack. Copy the code and enter it in the confirmation input and hit save.
6. Any action you take in Slack will now be tied to your FortiMonitor account

Microsoft Teams

Microsoft Teams is a group collaboration tool that Microsoft recent released that is similar in nature to Slack and HipChat. Like other collaboration tools, it is most effective to have all of your alerts sent to Teams in order to stay up to date on infrastructure performance.

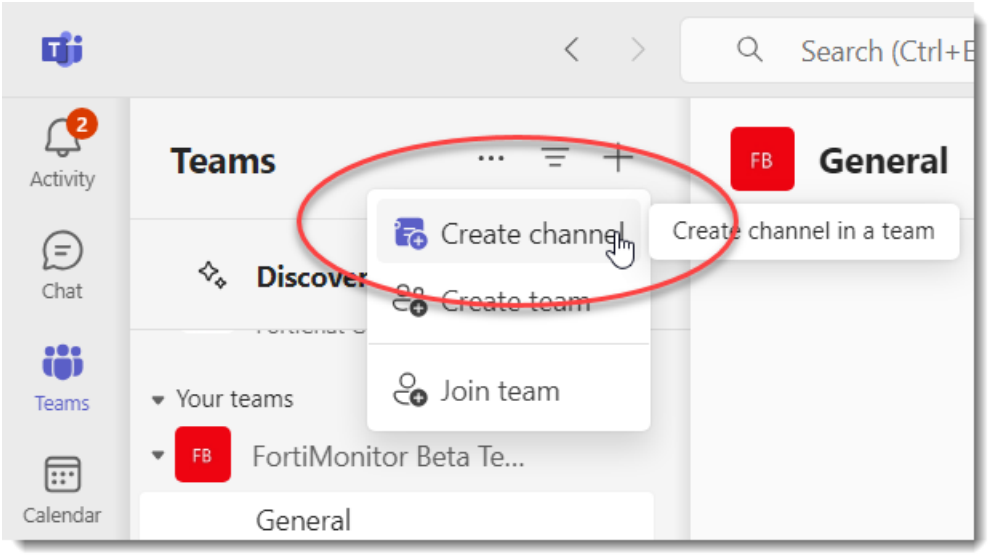
To set up the integration, you need to do the following:

1. At the minimum, a Microsoft Teams Essential package.
2. Add a **Workflow** to a Teams channel.
3. Set up the integration in FortiMonitor.

These procedures are detailed in their respective sections below.

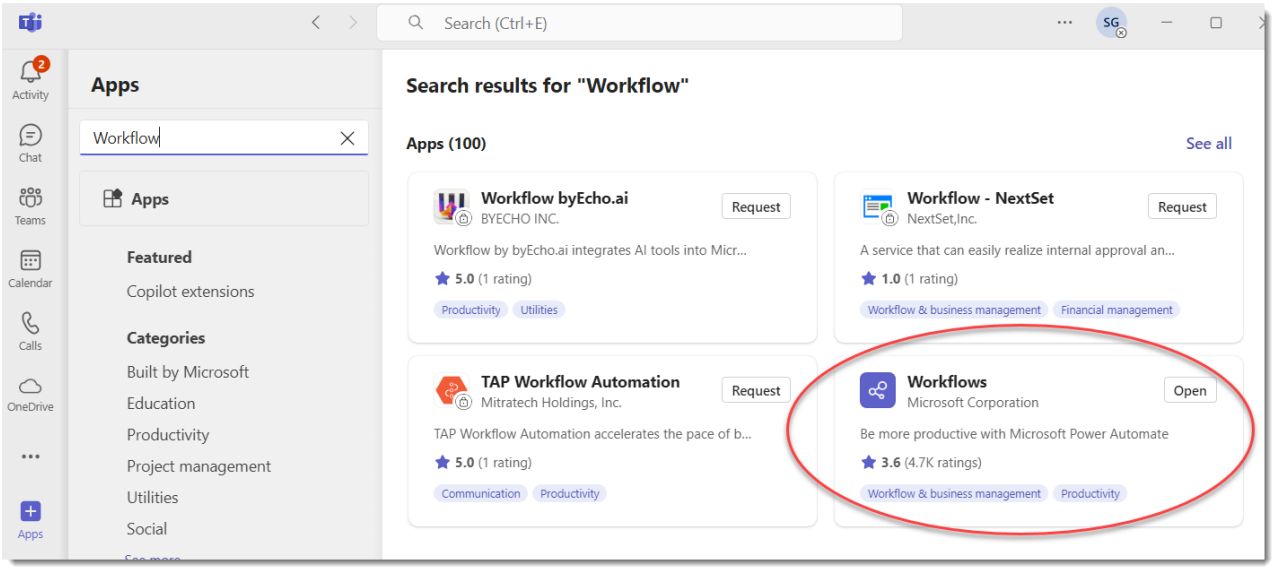
Add a Workflow to a Teams channel

1. Login to your **Microsoft Teams** application.
2. Create a new **Channel** where you want to receive notifications.



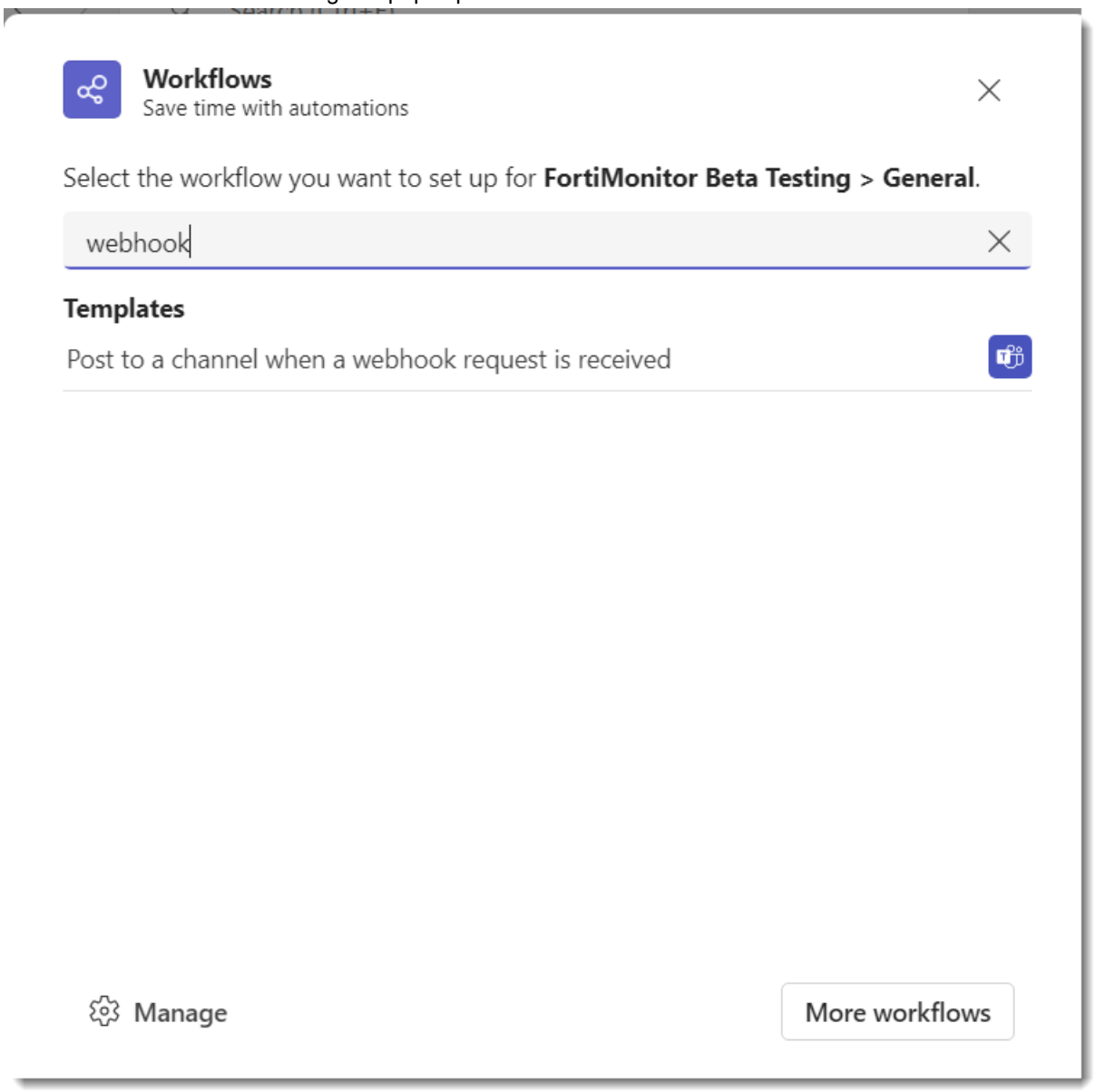
3. **Go to Apps** and search for **Workflow.**

4. Click the **Workflows** app card and select **Add**. If is already added, the button on the card will be **Open**.



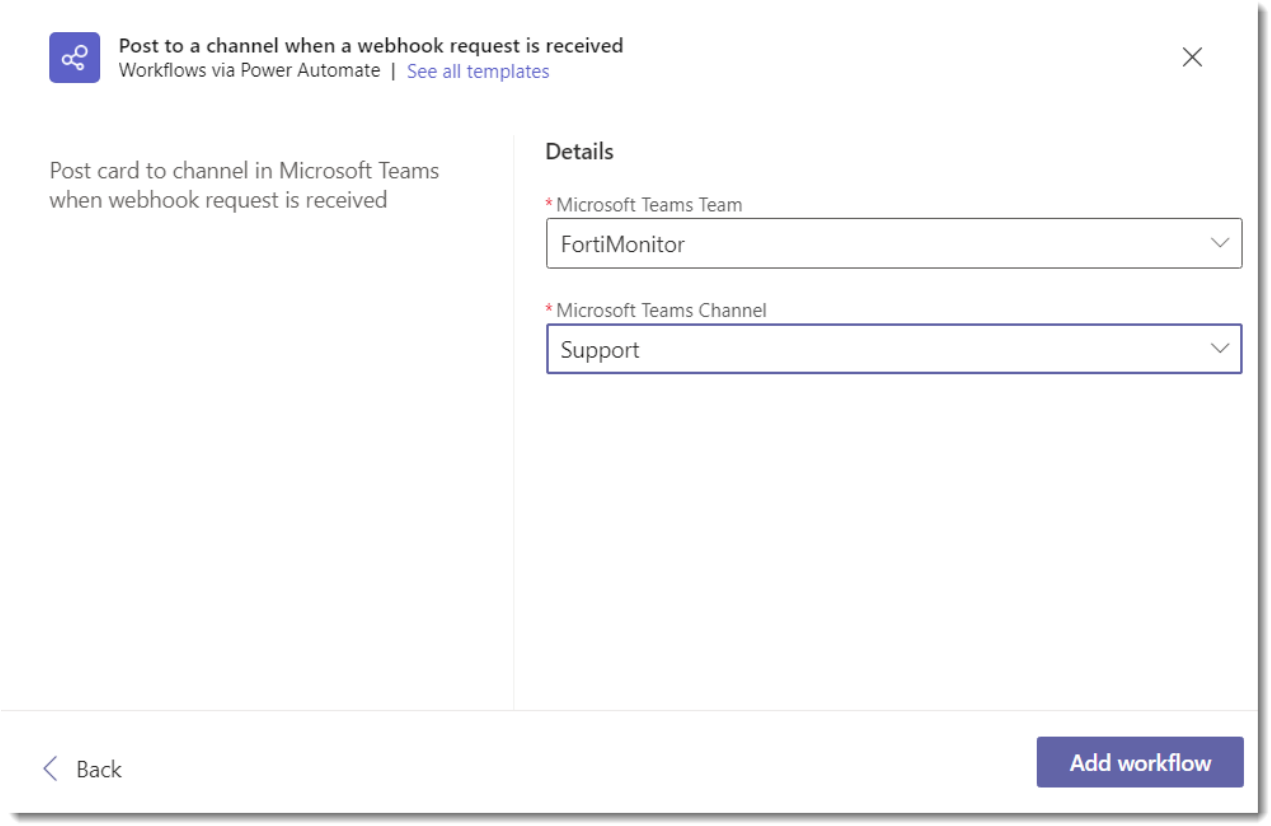
5. Go back to the new channel then click on the 3-dot menu.

6. Search for **webhook** in the dialog that pops up.

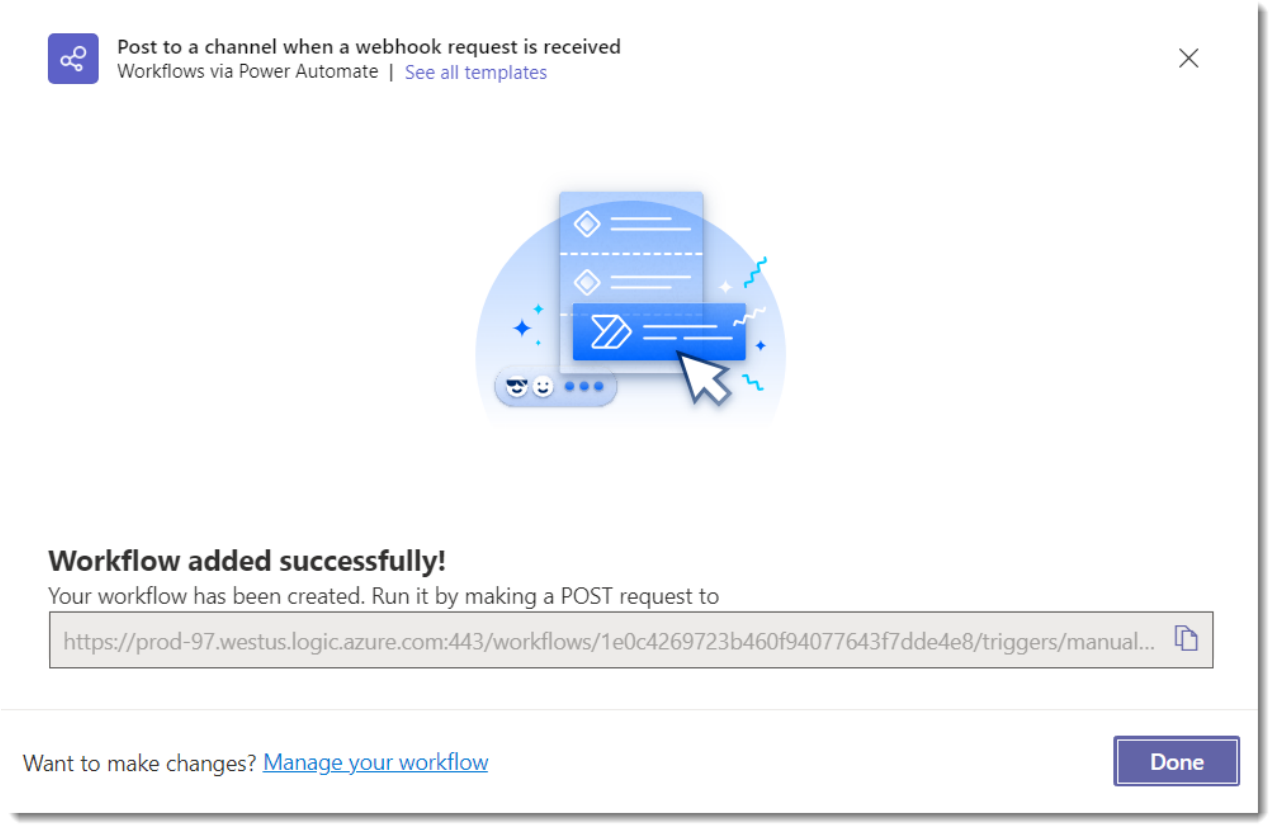


7. Select **Post to a channel when a webhook request is received**.
8. Click **Next** on the dialog. You now have the option to select where you want to receive notification.

9. Click the **Add Workflow** button.



10. After adding the **Workflow**, a new URL is displayed. Save this URL for later.

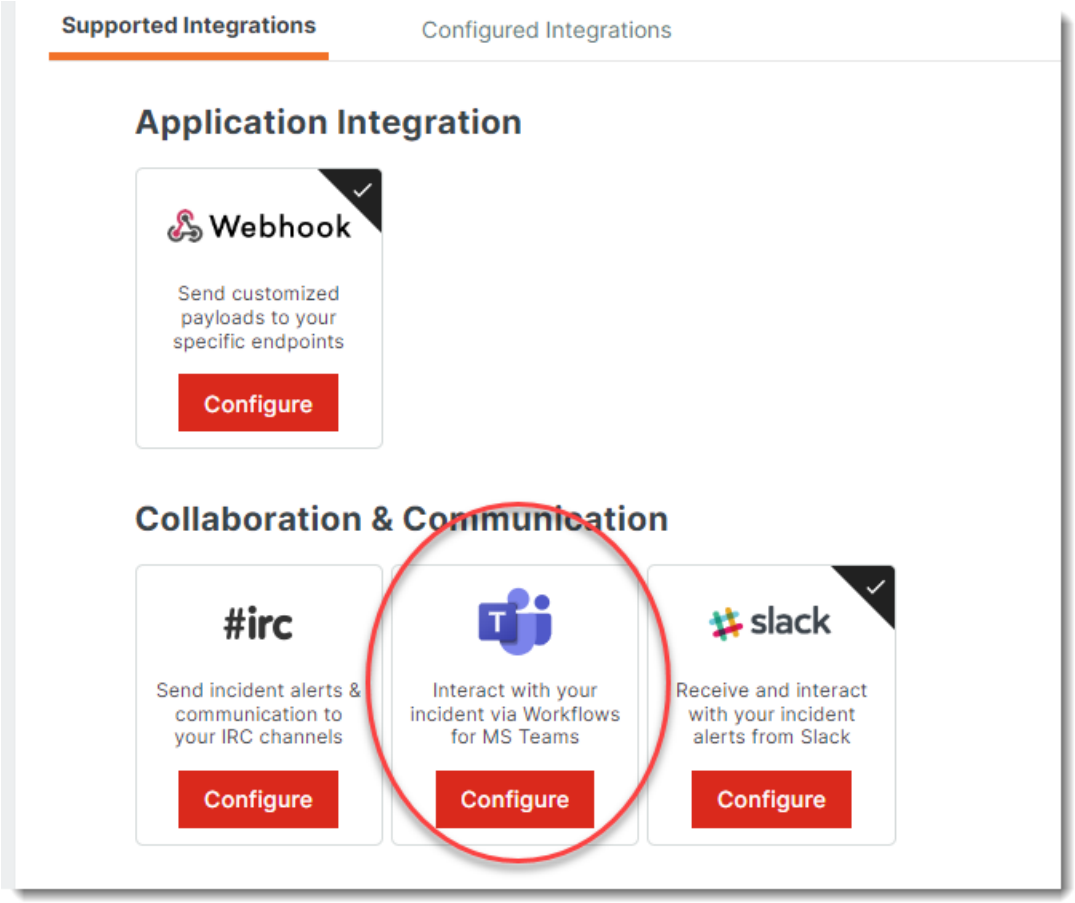


Set up the integration in FortiMonitor

After configuring **Teams**, continue setting up the integration in **FortiMonitor**.

1. Log in to FortiMonitor.
2. From the navigation menu, select **Teams & Activity**> **Integrations**.

3. Locate the **Teams** card then click **Configure**.



4. Configure your Teams integration. The configuration options are detailed in the following table:

Field	Description
Title	Enter a name for the MS Teams integration.
Webhook URL	URL generated from MS Teams that will allow you to send alerts to an MS Teams channel. Enter the copied URL from Teams in this field.

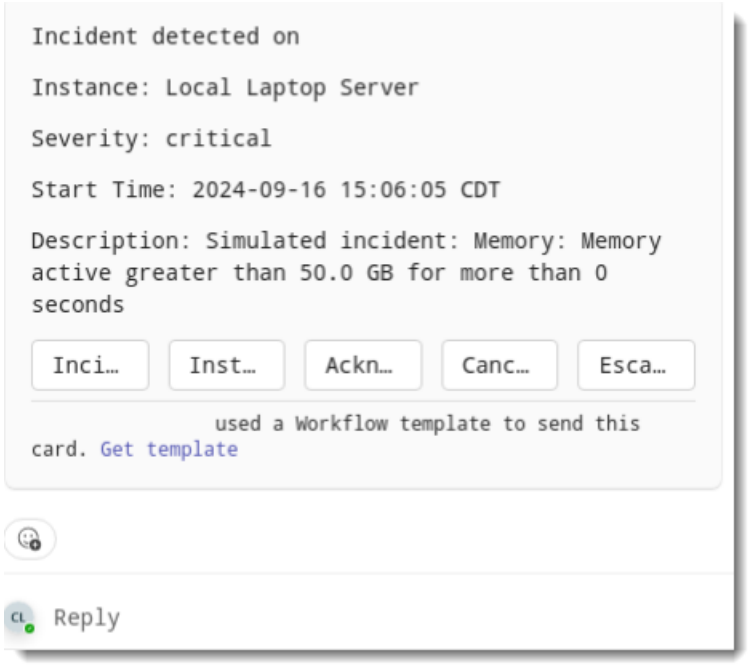
5. Customize the MS Teams **Outage** and **Clear** incident cards. These are the messages that will be posted to the channel when an incident is detected or is resolved. The parameters you can use are detailed in the following sections. The following parameters are preconfigured by default:



6. Select **Save**.
7. Go to the **Configured Integrations** tab to verify your Teams integration.
8. To set up notifications, configure the integration's Alert Timeline and add MS Teams to the timeline.

MS Teams incident card examples

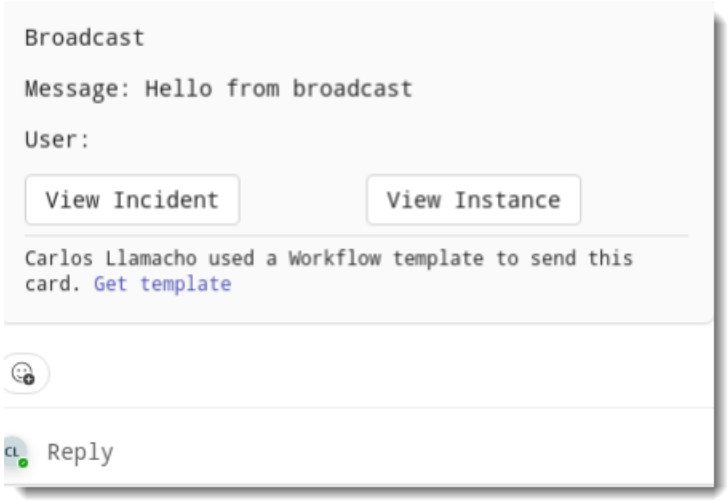
Incident is detected



Incident is resolved



Broadcast



Parameters

Parameter	Description
$trigger	The type of event that triggered this payload (outage, ack, broadcast, clear).
$fqdn	The Fully qualified domain name of the server experiencing the incident/clear.
$name	Name of the server experiencing the incident/clear.
$timestamp	UTC timestamp of when the incident/clear occurred.
$event	The type of event, either **incident event** or **clear event**.
$outage_id	The ID number of the associated incident.
$services	For service incident: services experiencing the incident/clear.
$compound_service_id	The ID number of the compound metric affected.
$network_service_id	The ID number of the network service affected.
$resource	For resource anomalies: resources experiencing the anomaly/clear.
$server_resource_id	The ID number corresponding to the resource affected.
$items	Services experiencing the incident/clear or resources experiencing the anomaly/clear.
$reasons	The reasons for network service incidents or the details for anomalies.
$duration	The duration of incidents/anomalies which will be filled in on-clear.
$tags	The tags for the server.
$server_key	The server key for the server.
$server_id	The ID number of the server experiencing the incident/clear.

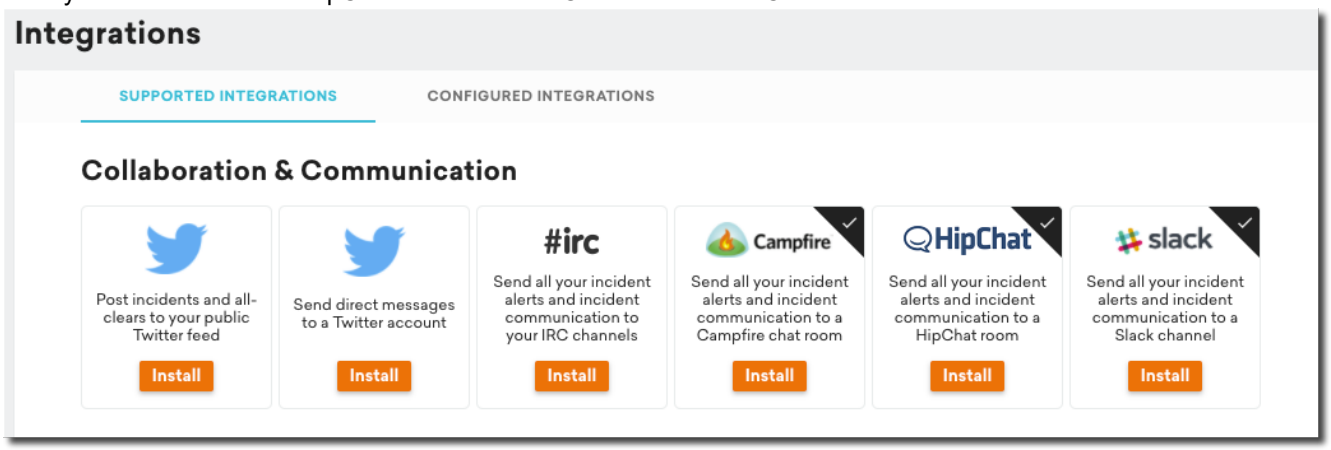
$partner_server_id	The partner server id for the server.
$item_type	The service type textkeys of the services experiencing the incident/clear, or plugin_textkey/resource_textkey combinations of the resources the experiencing the anomaly/clear, or the OID name of SNMP resources experiencing the outage/clear.
$custom_attribute	You can pass custom server attributes that are set on your servers. Use the attribute type as the key.
$alert_label	Alert label of the incident/anomaly.
$severity	The severity of the outage/anomaly, either **critical** or **warning**.
$metric_tags	The tags for all of the metrics involved in the outage.
$location	A comma-separated list of the primary monitoring probe names for all network services affected.

HipChat

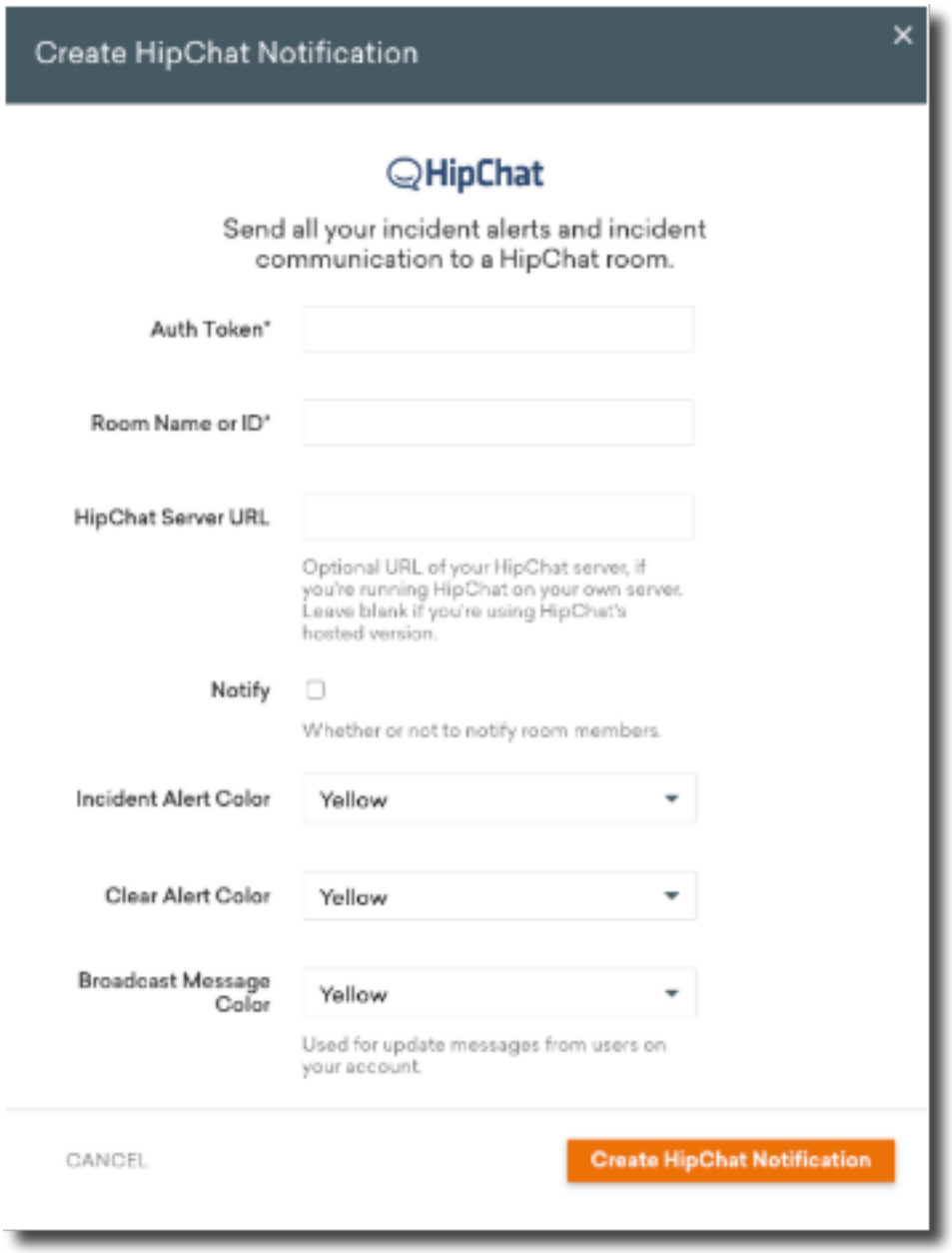
Adding HipChat alerts is a great way to keep your entire team informed of outages and server statuses as events arise.

First go to **Teams & Activity** > **Integrations**.

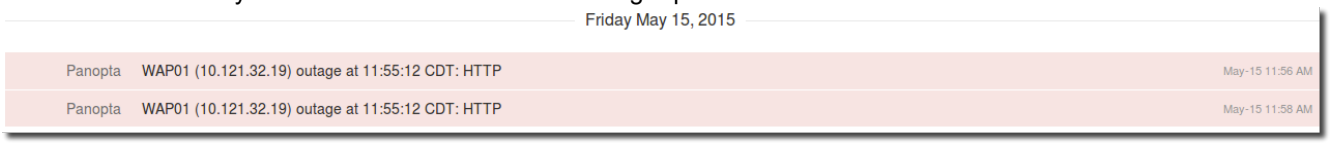
Here you should select the HipChat card from the "Collaboration and Communication" section.



A window will open up that will allow you to enter your HipChat information. You should notice that this page also provides the appropriate links used to find this information.



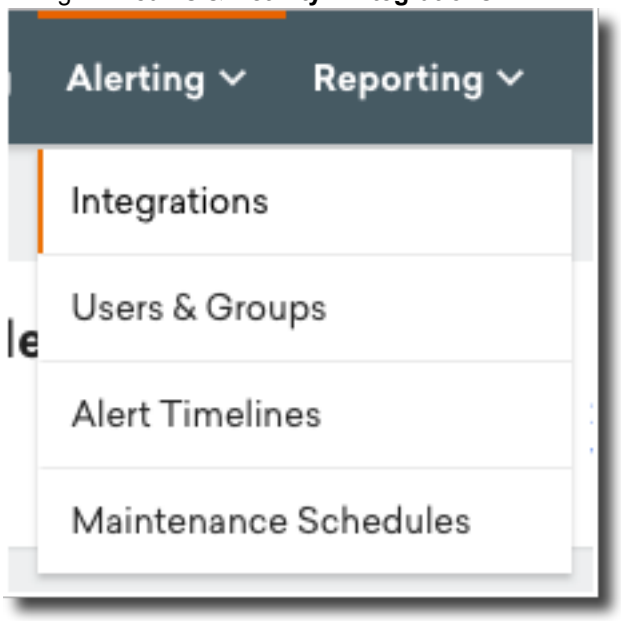
You can also set the color of the HipChat background depending on if the alert is an outage, all clear, or broadcast. Once you have finished configuring these settings click Create HipChat Notification. Now you will need to add this contact information to one of your Alert Timelines to start receiving HipChat alerts.



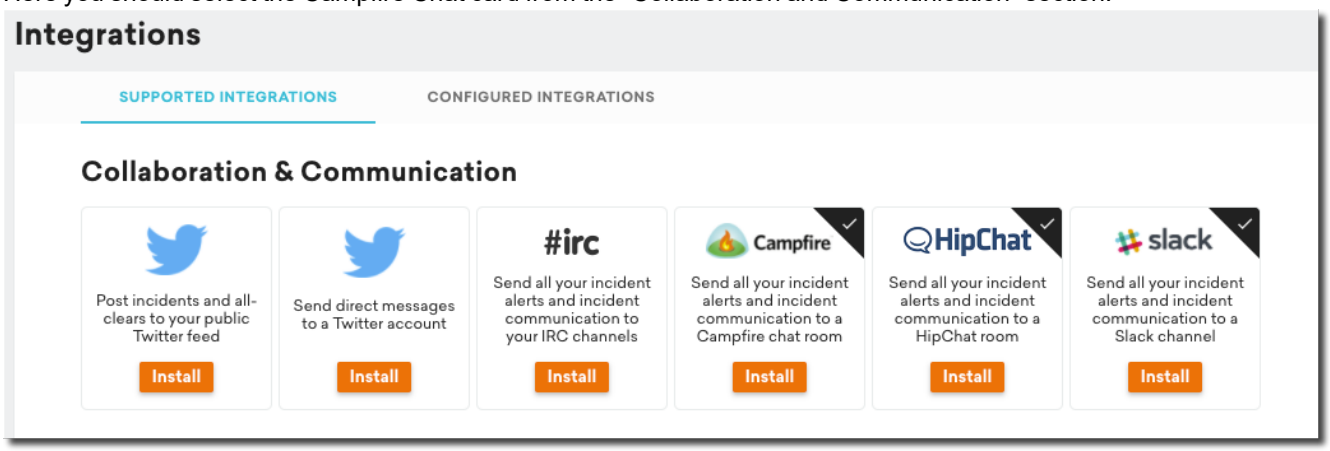
Campfire Chat

Campfire Chat is a chat room service created to help businesses communicate internally. Adding Campfire Chat alerts is a great way to keep your entire team informed of outages and server statuses as events arise.

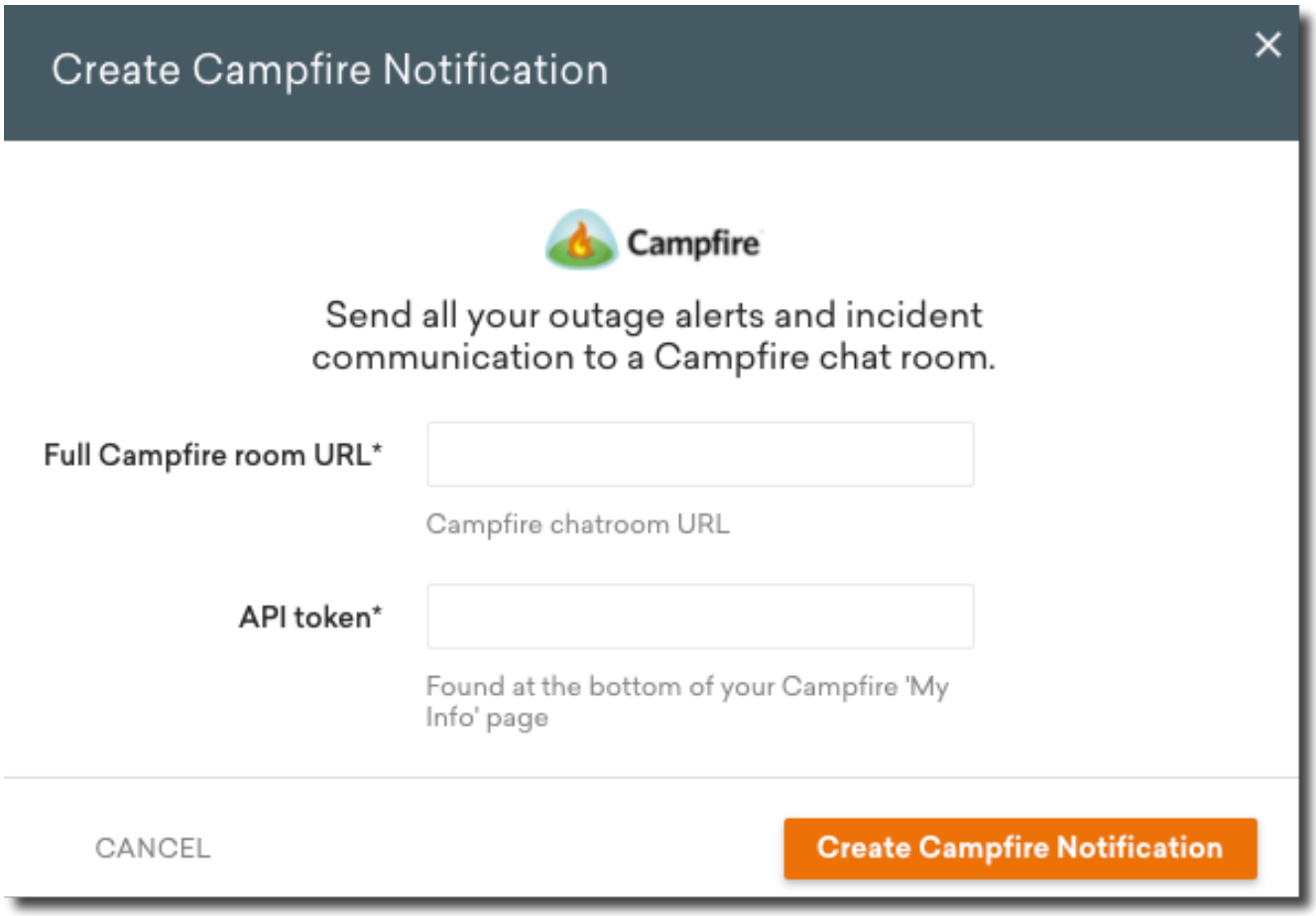
First go to **Teams & Activity** > **Integrations**.



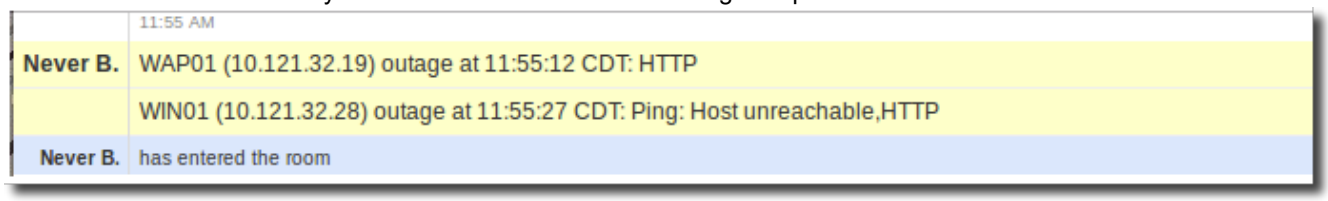
Here you should select the Campfire Chat card from the "Collaboration and Communication" section.



A window will open up that will allow you to enter your chat room URL and API token.



Once you have finished configuring these settings click Create Campfire Notification. Now you will need to add this contact information to one of your Alert Timelines to start receiving Campfire alerts.

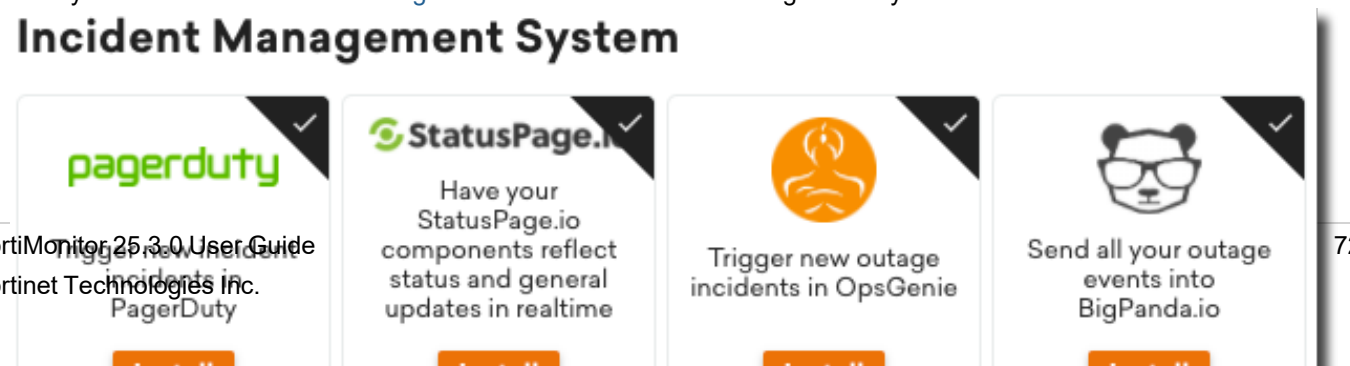


StatusPage.io

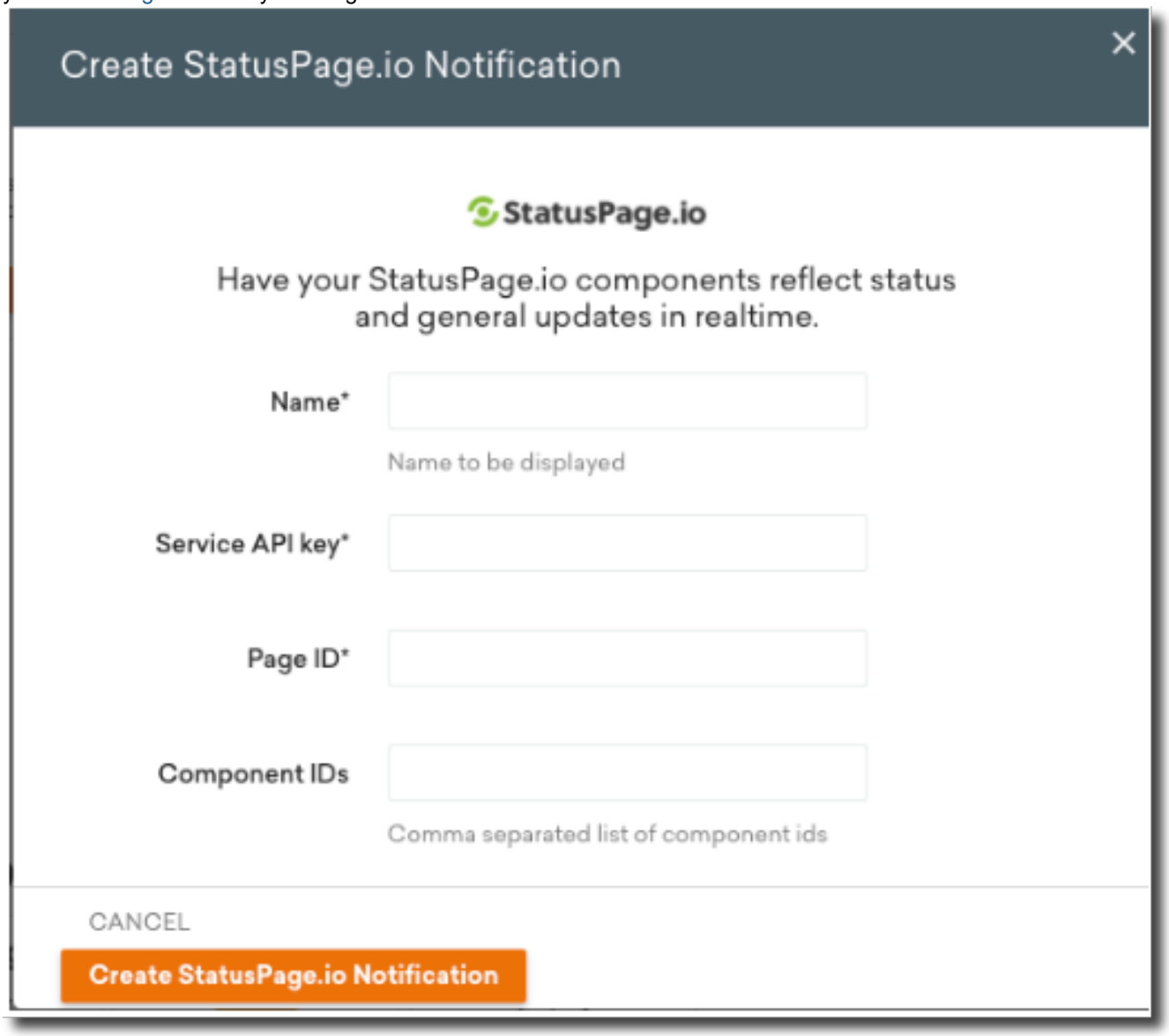
If you're not familiar with them already, StatusPage.io is a service that provides public status pages for its customers. Using StatusPage.io with FortiMonitor is a great way to create a public status page for your infrastructure which is independent of your system and always available.

To send your outage information to StatusPage.io, go to **Teams & Activity** > **Integrations**.

Here you should select the StatusPage.io card from the Incident Management System section.

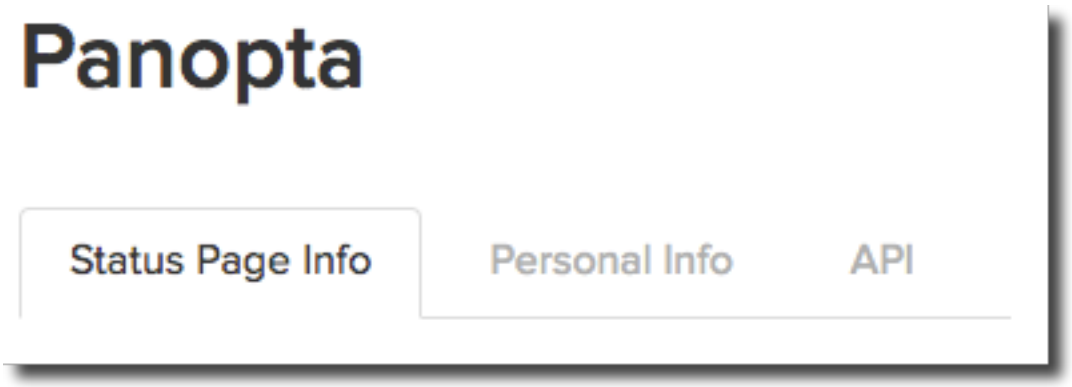


A window will open up that will allow you to enter your StatusPage.io information. As you can see below, you will need your StatusPage.io API key and Page ID.

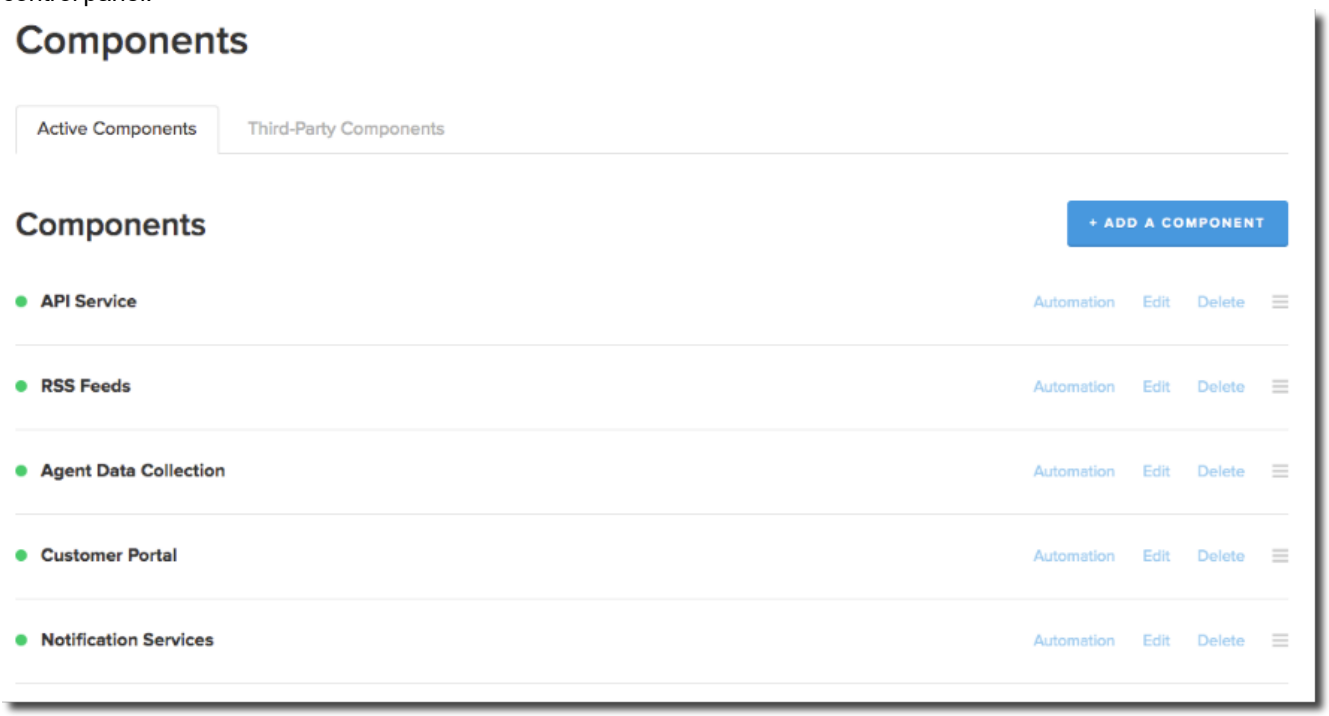


You can find this information by logging into your StatusPage.io and selecting "Account" from your user name's drop-down.

On this new page you can fin your API and Page ID under the API tab.



You can also add component IDs from your StatusPage.io account if you would like. First select "components" in the control panel.



On this new page select the "Automate Component" tab.

The Component ID's will be listed below, and they consist of the string between the "+" and the "@." You can add several of these if you would like. This will allow your outages to affect specific component in your StatusPage.

Once you have finished configuring these settings click Create StatusPage Notification. Now you will need to add this contact information to one of your Alert Timelines so that your StatusPage account will start receiving outage information.

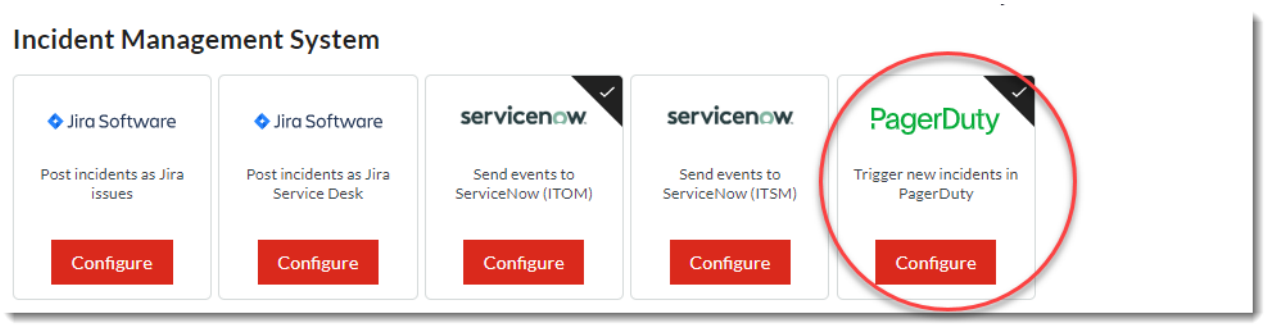
PagerDuty

If you are using multiple monitoring services with their own notification configurations, you may want a tool that can integrate with these different monitoring tools to streamline your Alert Timeline. PagerDuty is the perfect tool for doing exactly that. By integrating your monitoring with PagerDuty you can ensure that all of your systems alarms can be organized within a single framework.

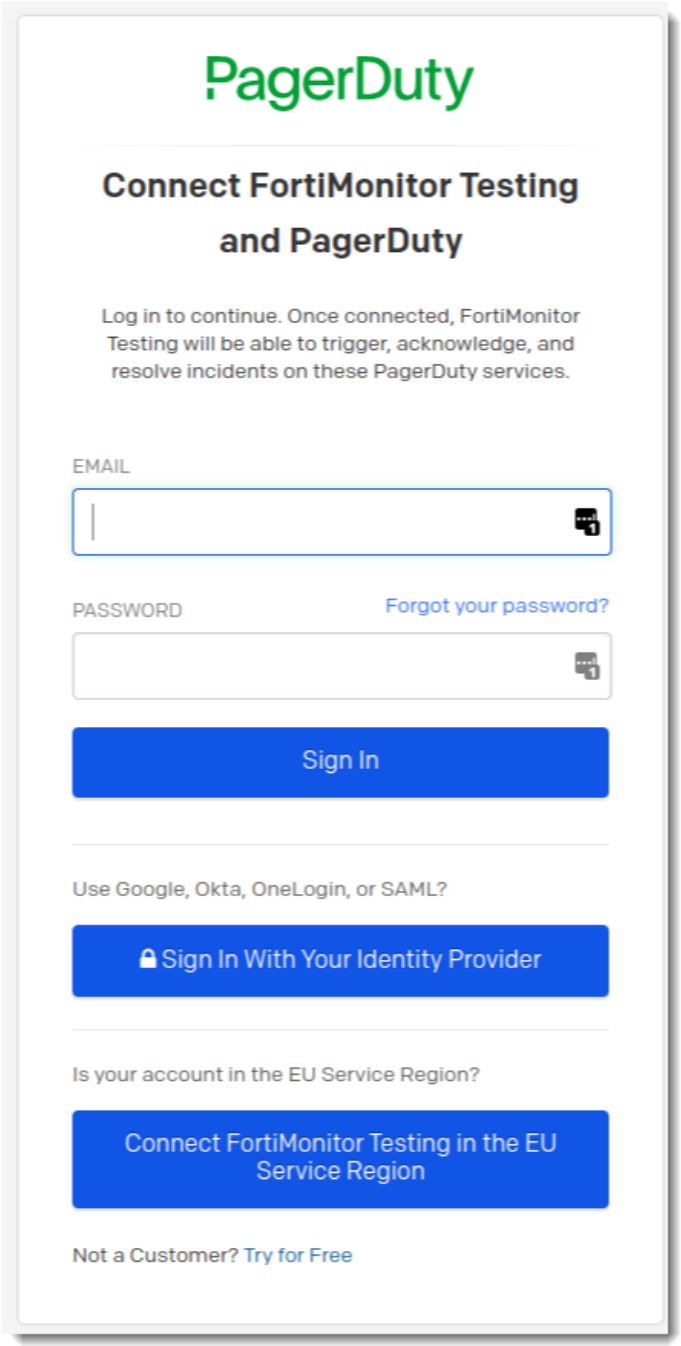
Set up the integration in FortiMonitor

To set up the integration, perform the following steps:

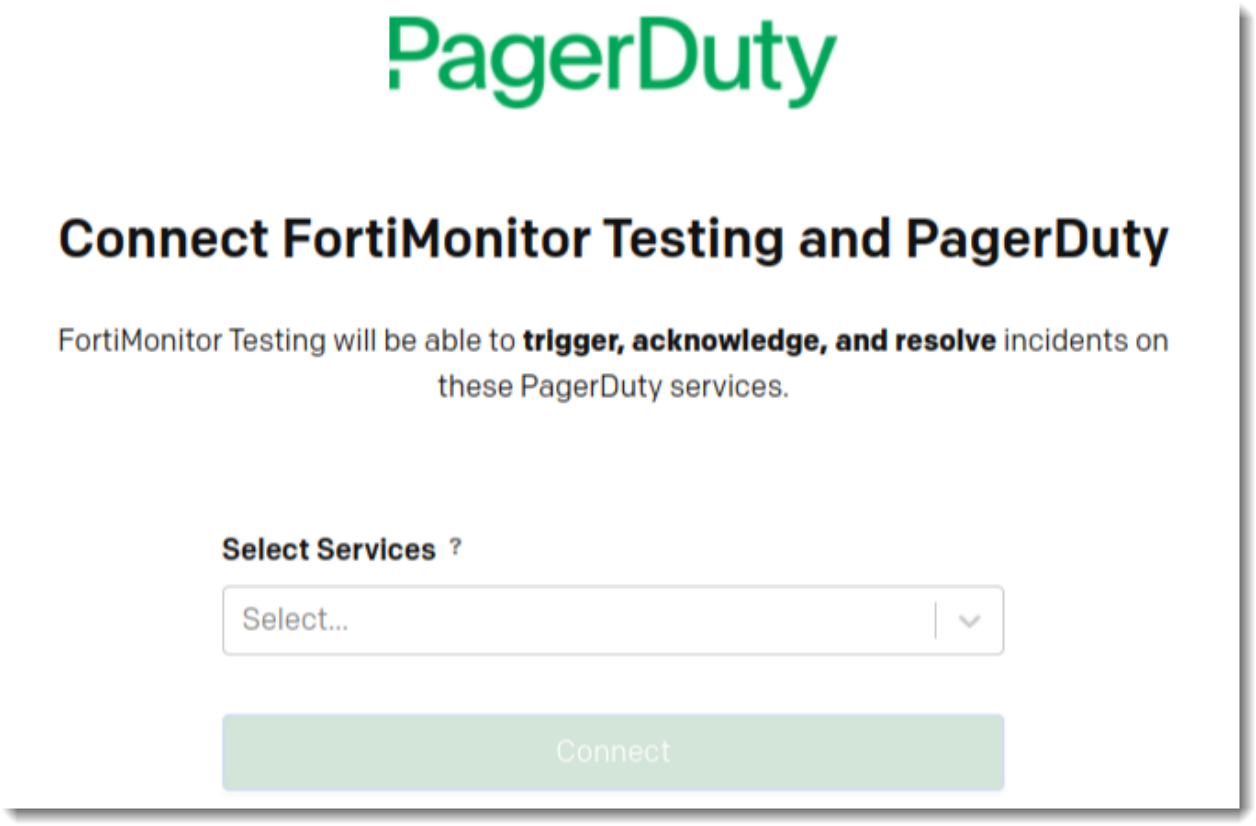
1. In FortiMonitor, go to **Teams & Activity** > **Integrations**.
2. Under the **Incident Management System** section, select the **PagerDuty** card.



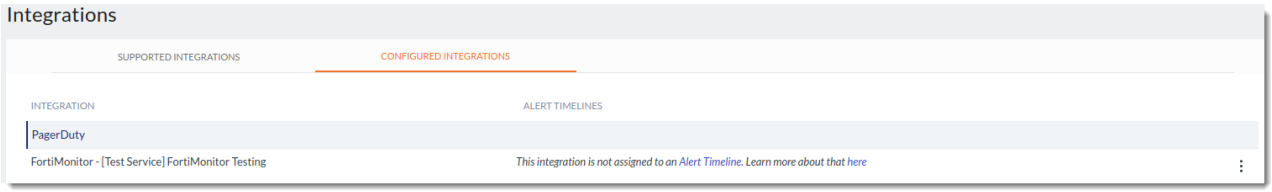
3. The PagerDuty login tab is displayed. Enter your email and password then select **Authorize Integration**.



4. PagerDuty will then ask you to select one or multiple services to complete the integration.



5. After a moment your new PagerDuty integration, complete with your network key, will show up under your configured integrations tab. If this does not happen, then this means that the integration was not successful and you should contact our support team. Do not forget to add user contact information to whichever Alert Timeline you would like to pass through PagerDuty.



Additional support

For further help with setting up the integration, contact our support team.

OpsGenie

OpsGenie is an incident response orchestration platform for teams to manage alerts & the incident resolution process. See https://docs.opsgenie.com/docs/panopta-integration for details on how to prepare OpsGenie for this integration.

To integrate OpsGenie with FortiMonitor, perform the following steps:

1. Go to **Teams & Activity** > **Integrations**.
2. Locate **OpsGenie** under the **Incident Management System** grouping.

3. Click **Install**.
4. In the modal that appears, enter your API key obtained from OpsGenie.
   - Obtaining your API Key
     - In OpsGenie, in the left menu, select **Integrations**.
       - Search for and select **Panopta**.
       - Copy the API key from the **Settings** block.
       - Make any other appropriate changes and hit **Save**.
5. You most likely don't need to change the URL unless you're overriding the geographic location. This is an OpsGenie setting - if you have questions, please reach out to them.

To leverage this new integration, you can add it to any Alert Timeline. Learn more about Alert Timelines to start receiving OpsGenie alerts.

Severities

The table below describes how FortiMonitor severities - **Critical**, **Warning**, and **Simulated** - map to the **Priority** field in OpsGenie.

FortiMonitor Severity	OpsGenie Priority
Critical	P1
Warning	P3
Simulated	P5

Webhooks

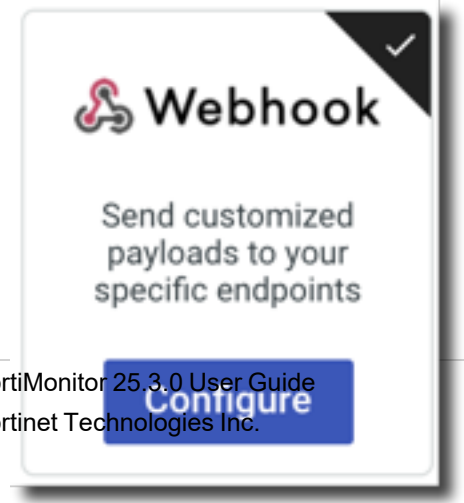
The Webhook Notification integration allows you to receive an HTTP request whenever your servers experience an outage or return from an outage. If you have multiple outages at the same time, you will receive a separate HTTP request per outage, as opposed to them being grouped together, like they are in other notification types.

All webhook requests originate from the following IP addresses:

- 104.197.35.194/32
- 35.185.29.9/32

Configure your firewall to allow inbound webhook traffic. Note that these IP addresses are subject to change and you will be notified prior to any such change.

The Postback can be set up via the **Integrations** page, which is found in **Teams & Activity** > **Integrations**.



Configuration

1. Go to **Teams & Activity > Integration**
2. Locate the **Webhook** card then click **Configure**.
3. Configure the following fields:

Field	Description
Title	Enter a title for the Webhook. This title will be referred to in the Alert Timelines.
Trigger Event	Select the type of event that will trigger the payload.
Request Method	Alerts can be sent via HTTP POST, PUT, DELETE, or GET operations, choose whichever your system is set up to receive.
Postback URL	The URL you wish to send the alert to.
HTTP Headers	If you need HTTP headers, select **Add New Header**.
Authentication Method	Select **None** or **Basic**. If you select basic, enter your credentials in the **HTTP Auth Username** and **HTTP Auth Password** fields.
Payload Type	Select **Raw Payload** or **Form Variables**. If you select **Form Variables**, you can directly input the variable's name and value. The values can be set up to include macros that expand to the actual details of the outage, for example including **$fqdn** in a value will get replaced with the actual domain name of the outage. Details on all of the parameters are shown on the right side of the form. For more information, see Raw payload example.

4. Select **Save**.

To add another Webhook event, select **Add Event** then repeat steps 3 and 4.

The webhook functionality is quite useful for integrating monitoring results into the rest of your infrastructure. Some of the uses that we have seen so far are:

- Automatically open tickets in your ticketing system when an outage begins.
- Display alerts for currently down servers in an internal company dashboard/panic board.
- Pipe messages into internal chat systems (although we now support sending alerts directly to Google Chat and IRC chat rooms as well).

If you have other interesting integrations that you set up with this, we'd love to hear about them. Also, if you find that the webhook functionality is missing something that is preventing you from using it, please let us know about that too so we can see if it can be extended.

Raw payload example

Incident Detection webhook

```
{
"cards": [
{
"header": {
```

```
"title": "Panopta - $trigger",
"imageUrl": "https://www.panopta.com/wp-content/uploads/2017/01/Favicon.png"
},
"sections": [
{
"widgets": [
{
"keyValue": {
"topLabel": "Severity",
"content": "$severity"
}
},
{
"keyValue": {
"topLabel": "Instance",
"content": "<a href='https://my.panopta.com/report/InstanceDetails?server_id=$server_
id#panel_1'>$name</a>"
}
},
{
"keyValue": {
"topLabel": "Start Time",
"content": "$timestamp"
}
},
{
"keyValue": {
"topLabel": "Impacted Metric(s)",
"content": "$items"
}
}
]
},
{
"widgets": [
{
```

```
"buttons": [
{
"textButton": {
"text": "View in Panopta",
"onClick": {
"openLink": {
"url": "https://my.panopta.com/outage/IncidentDetails?incident_id=$outage_id#panel_1"
}
}
}
}
]
}
]
}
]
}
```

Parameters

Parameter	Description
$trigger	The type of event that triggered this payload (outage, ack, broadcast, clear)
$fqdn	The Fully qualified domain name of the server experiencing the incident/clear.
$name	Name of the server experiencing the incident/clear.
$timestamp	UTC timestamp of when the incident/clear occurred.
$event	The type of event, either **incident event** or **clear event**.
$outage_id	The ID number of the associated incident.
$services	For service incident: services experiencing the incident/clear.
$compound_service_id	The ID number of the compound metric affected.
$network_service_id	The ID number of the network service affected.
$resource	For resource anomalies: resources experiencing the anomaly/clear.
$server_resource_id	The ID number corresponding to the resource affected.

$items	Services experiencing the incident/clear or resources experiencing the anomaly/clear.
$reasons	The reasons for network service incidents or the details for anomalies.
$duration	The duration of incidents/anomalies which will be filled in on-clear.
$tags	The tags for the server.
$server_key	The server key for the server.
$server_id	The ID number of the server experiencing the incident/clear.
$partner_server_id	The partner server id for the server.
$item_type	The service type textkeys of the services experiencing the incident/clear, or plugin_textkey/resource_textkey combinations of the resources the experiencing the anomaly/clear, or the OID name of SNMP resources experiencing the outage/clear.
$custom_attribute	You can pass custom server attributes that are set on your servers. Use the attribute type as the key.
$alert_label	Alert label of the incident/anomaly.
$severity	The severity of the outage/anomaly, either **critical** or **warning**.
$metric_tags	The tags for all of the metrics involved in the outage.
$location	A comma-separated list of the primary monitoring probe names for all network services affected.
$message	For ack and broadcast payloads, the message that was sent.
$incident_lead	The incident lead's email address, if set.
$alert_timeline	The name of the alert timeline used for the incident.
$alert_timeline_id	The ID of the alert timeline used for the incident.

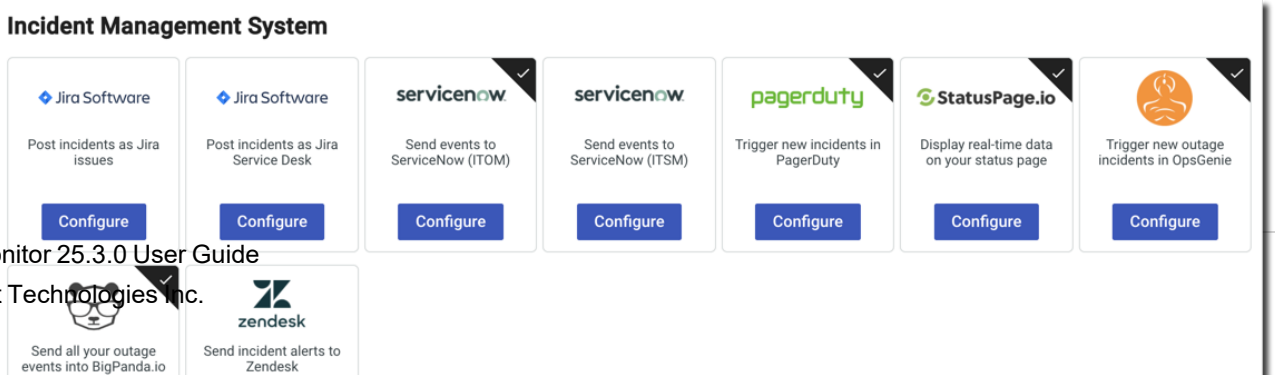
Jira and Jira Service Desk

FortiMonitor's Jira and Jira Service Desk integrations allow you to post incidents as issues to your Jira and Jira Service Desk environments.

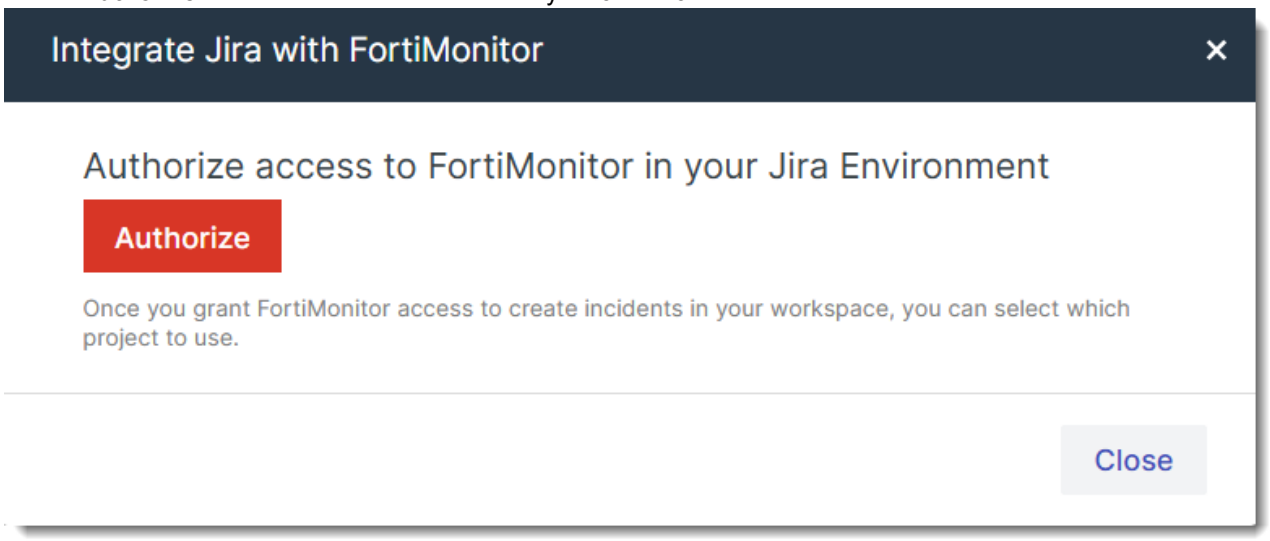
Set up the integration

The integration can be setup via the **Integrations** page.

1. From the navigation menu, select **Teams & Activity** > **Integrations**.
   Choose whether you want to post issues in Jira or Jira Service desk then select **Configure**.



Select **Authorize** to allow FortiMonitor access to your Jira or Jira service desk environment.



3.
4. On the next screen, select **Accept**.
5. Configure your Jira or Jira service desk integration. The configuration options are detailed in the following table:

Field	Description
Projects	Select which project the Jira or Jira Service Desk will be created when an incident occurs.
Issue type	Select the type of issue to be created. **Values**: • New feature • Story • Task • Support • Bug
Close incident in Jira upon incident resolution	Select this option if you want Jira or Jira Service Desk to automatically close issues once an incident is resolved.
Close Type	Enter the name of the transition to apply when an incident is resolved.
JSON Payload	The customizable JSON payload sent to the Jira API. You can create a payload for the following actions: • Incident creation – A ticket is created in Jira whenever an incident is detected. • Incident closure – The ticket is closed once the incident has been resolved.

6. Select **Save Integration**.
7. Go to the **Configured Integrations** tab to verify your Jira or Jira Service Desk integration.
8. To set up notifications, configure the integration's Alert Timeline.

JSON payload

Incident creation

The following JSON payload is sent to Jira whenever an incident is detected. See Parameters for details on how to customize the payload.

Incident closure

The following JSON payload is sent to Jira once an incident is resolved. See Parameters for details on how to customize the payload.

Parameters

Parameter	Description
$trigger	The type of event that triggered this payload (outage, ack, broadcast, clear)
$fqdn	The Fully qualified domain name of the server experiencing the incident/clear.
$name	Name of the server experiencing the incident/clear.
$timestamp	UTC timestamp of when the incident/clear occurred.
$event	The type of event, either **incident event** or **clear event**.
$outage_id	The ID number of the associated incident.
$services	For service incident: services experiencing the incident/clear.
$compound_service_id	The ID number of the compound metric affected.
$network_service_id	The ID number of the network service affected.
$resource	For resource anomalies: resources experiencing the anomaly/clear.
$server_resource_id	The ID number corresponding to the resource affected.
$items	Services experiencing the incident/clear or resources experiencing the anomaly/clear.
$reasons	The reasons for network service incidents or the details for anomalies.
$duration	The duration of incidents/anomalies which will be filled in on-clear.
$tags	The tags for the server.
$server_key	The server key for the server.
$server_id	The ID number of the server experiencing the incident/clear.
$partner_server_id	The partner server id for the server.

$item_type	The service type textkeys of the services experiencing the incident/clear, or plugin_textkey/resource_textkey combinations of the resources the experiencing the anomaly/clear, or the OID name of SNMP resources experiencing the outage/clear.
$custom_attribute	You can pass custom server attributes that are set on your servers. Use the attribute type as the key.
$alert_label	Alert label of the incident/anomaly.
$severity	The severity of the outage/anomaly, either **critical** or **warning**.
$metric_tags	The tags for all of the metrics involved in the outage.
$location	A comma-separated list of the primary monitoring probe names for all network services affected.

Zendesk

FortiMonitor's Zendesk integration allows you to automatically create Zendesk tickets when an incident is detected by FortiMonitor. When the incident resolves, the Zendesk ticket will also be updated with the appropriate state. You can customize the close state while configuring the integration.

Installation

The integration can be setup via the **Integrations** page.

1. From the navigation menu, select **Teams & Activity** > **Integrations**.
   Look for the Zendesk card then select **Configure**.



2.
3. Enter the following information:

Field	Description
Title	A name for the Zendesk integration.
Zendesk Instance Name	The part after **https://** and before **.zendesk.com.**

Username	The username/email address you use to log in to Zendesk. **Note**: You must include **/token** after your email address.
Token	Zendesk API token
Close state	When the incident resolves in FortiMonitor, you can choose which state to set the ticket to in Zendesk. • Closed (Default) • Pending • Hold • Solved
JSON Payload	The customizable JSON payload sent to Zendesk API.

JSON payload

The following JSON payload is sent to Zendesk whenever an incident is detected. See Parameters for details on how to customize the payload.

```
{
	"ticket": {
		"subject":  "$severity incident on $name",
		"type":"incident",
		"tags":"$tags",
		"comment":  { "html_body": "<strong>Instance: </strong>$name
</br><strong>Reason: </strong> $reasons </br> <strong>Control Panel Link: </strong><a
href='https://my.panopta.com/outage/IncidentDetails?incident_id=$outage_
id'>my.panopta.com/outage/IncidentDetails?incident_id=$outage_id</a></br><strong>Start Time:
</strong> $timestamp" },
		"priority": "urgent"
	}
}
```

Payload example

```
{
	"ticket": {
		"subject":  "critical incident on SQL01",
		"type":"incident",
		"tags":"IIS,activedirectory,dfs,exchange,sql,sqlserver,windows",
		"comment":  { "html_body": "<strong>Instance: </strong>SQL01
</br><strong>Reason: </strong> Memory: % Used greater than 75% for more than 5 minutes </br>
<strong>Control Panel Link: </strong><a
href='https://my.panopta.com/outage/IncidentDetails?incident_id=-
```

```
126783267'>my.panopta.com/outage/IncidentDetails?incident_id=-
126783267</a></br><strong>Start Time: </strong> 2020-10-05 04:49:36 UTC" },
            "priority": "urgent"
      }
}
```

Parameters

Parameter	Description
$trigger	The type of event that triggered this payload (outage, ack, broadcast, clear)
$fqdn	The Fully qualified domain name of the server experiencing the incident/clear.
$name	Name of the server experiencing the incident/clear.
$timestamp	UTC timestamp of when the incident/clear occurred.
$event	The type of event, either **incident event** or **clear event**.
$outage_id	The ID number of the associated incident.
$services	For service incident: services experiencing the incident/clear.
$compound_service_id	The ID number of the compound metric affected.
$network_service_id	The ID number of the network service affected.
$resource	For resource anomalies: resources experiencing the anomaly/clear.
$server_resource_id	The ID number corresponding to the resource affected.
$items	Services experiencing the incident/clear or resources experiencing the anomaly/clear.
$reasons	The reasons for network service incidents or the details for anomalies.
$duration	The duration of incidents/anomalies which will be filled in on-clear.
$tags	The tags for the server.
$server_key	The server key for the server.
$server_id	The ID number of the server experiencing the incident/clear.
$partner_server_id	The partner server id for the server.
$item_type	The service type textkeys of the services experiencing the incident/clear, or plugin_textkey/resource_textkey combinations of the resources the experiencing the anomaly/clear, or the OID name of SNMP resources experiencing the outage/clear.
$custom_attribute	You can pass custom server attributes that are set on your servers. Use the attribute type as the key.
$alert_label	Alert label of the incident/anomaly.

$severity	The severity of the outage/anomaly, either **critical** or **warning**.
$metric_tags	The tags for all of the metrics involved in the outage.
$location	A comma-separated list of the primary monitoring probe names for all network services affected.
$message	For ack and broadcast payloads, the message that was sent.

Salesforce

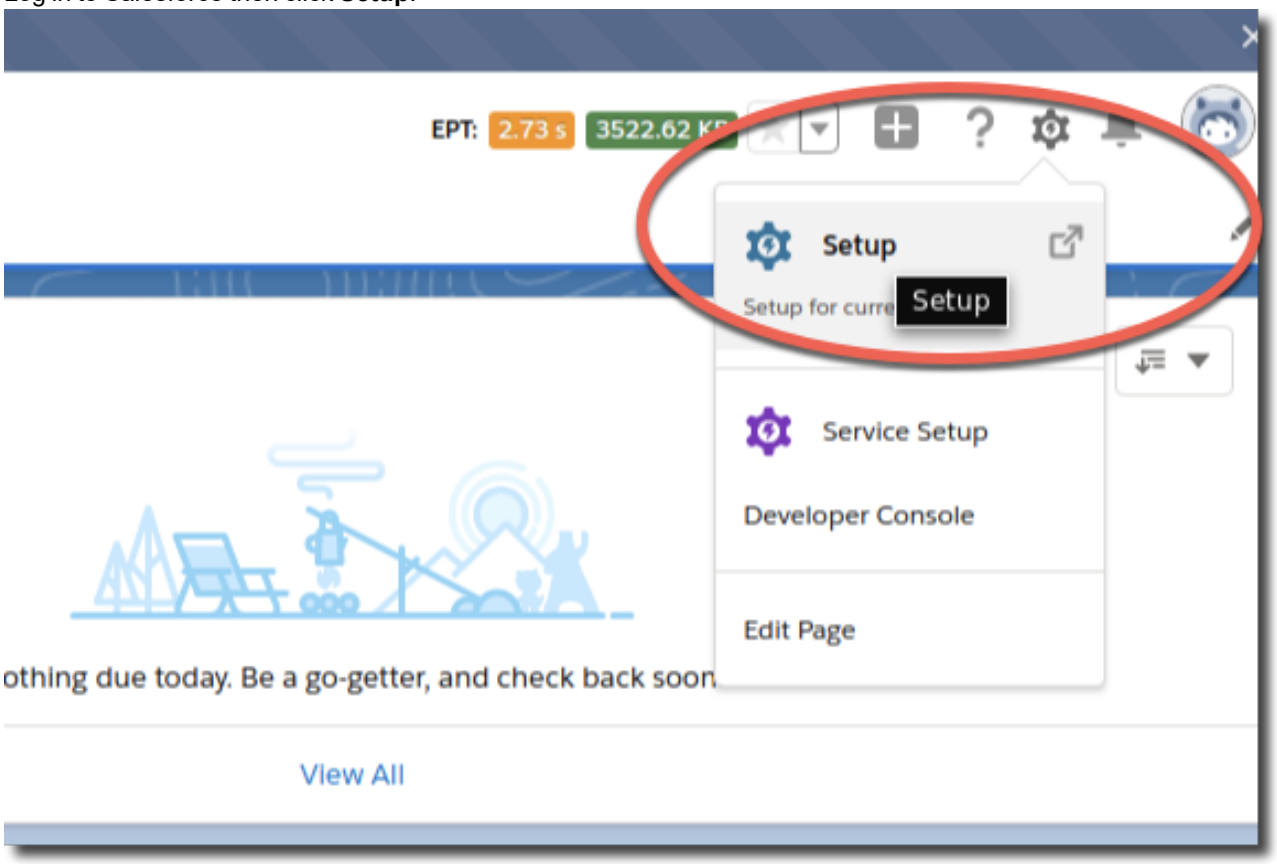
FortiMonitor's Salesforce integration allows you to post incidents detected by FortiMonitor as cases to your Salesforce environment. To set up the integration, you need to do the following:

1. Configure a new connected app in Salesforce.
2. Reset your security token.
3. Set up the integration in FortiMonitor.

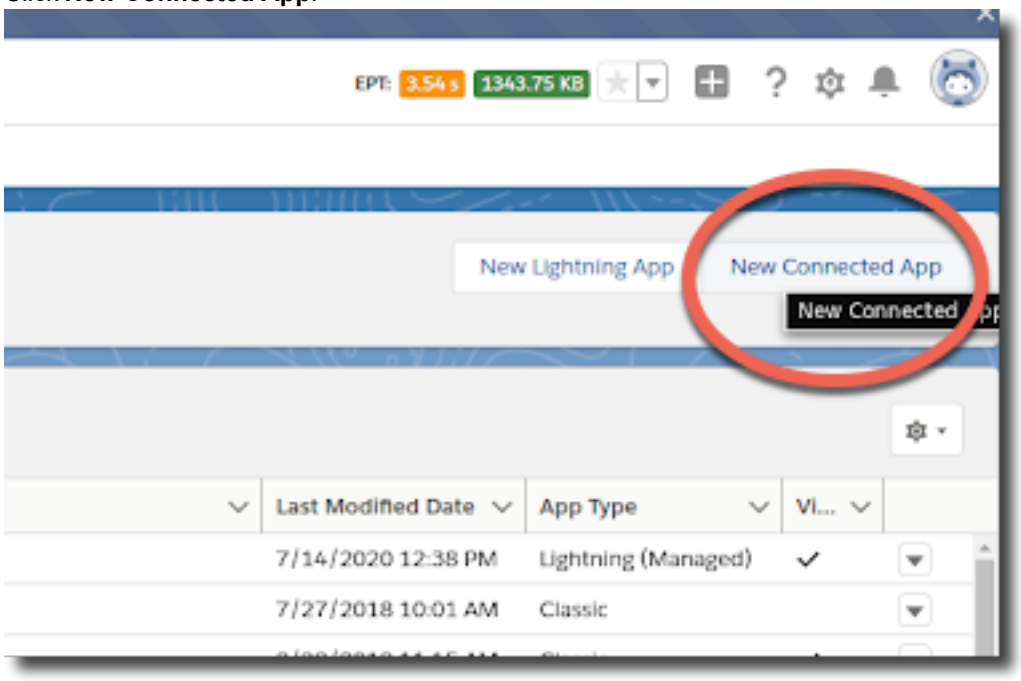
These procedures are detailed in their respective sections below.

Configure a new connected app in Salesforce

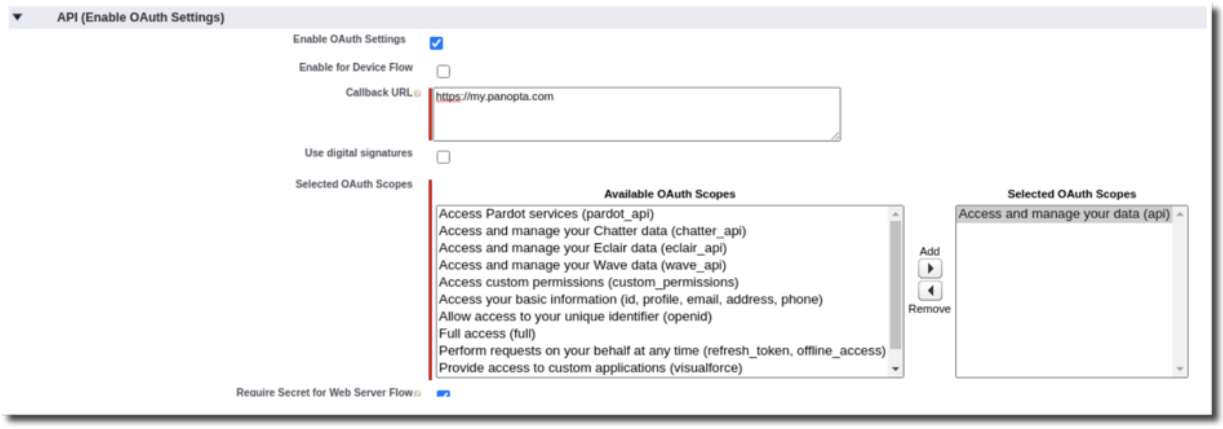
1. Log in to Salesforce then click **Setup**.



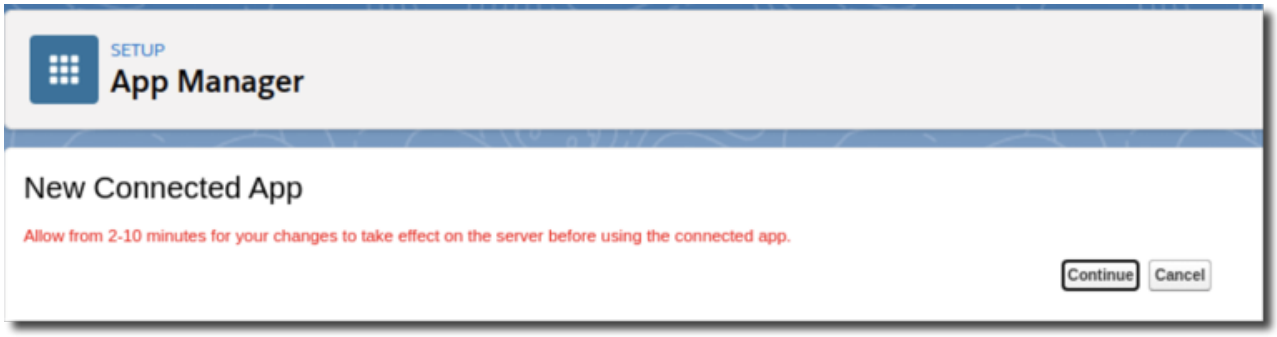
2. Click **New Connected App**.



3. In the **Basic Information** area of the page, specify the following information:
   a. Connected App Name – Specify a name for the connected app.
   b. API Name – A unique identifier used when referring to the app. You can hit **[Tab]** to automatically populate the field. The **API Name** defaults to the App Name, replacing spaces with underscores.
   c. Contact Email - Specify an email address to associate with the connected app.
4. In the **API (Enable OAuth Settings)** area, configure the following:
   a. Enable OAuth Settings – Click the checkbox to enable the option.
   b. Callback URL – Enter **https://fortimonitor.forticloud.com**.
   Note: New users should use **fortimonitor.forticloud.com** for the SSO URL. For long time customers of FortiMonitor, continue using **my.panopta.com**.
   c. Selected OAuth Scopes – Select **Access and manage your data (API)**.
   Require Secret for Web Server Flow – Click the checkbox to enable the option.
   d.

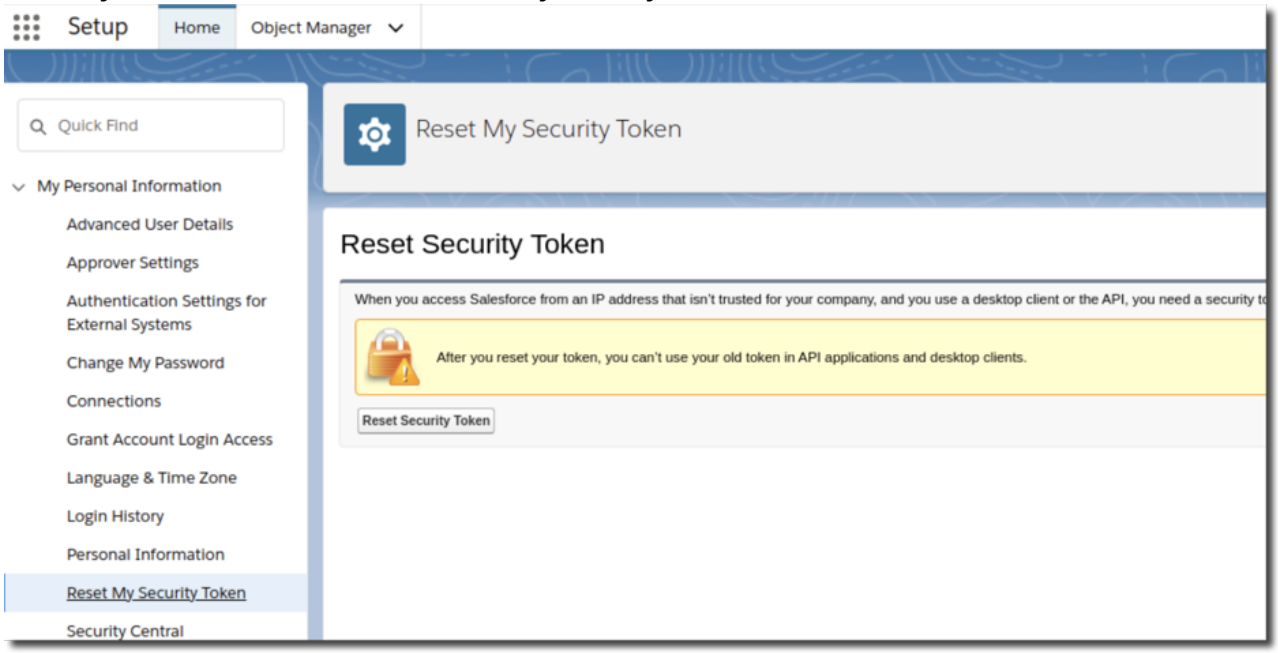


5. Click **Save**.
   Click **Continue**.
6.



7. On the next page, take note of your **Consumer Key** and **Consumer Secret**. You will need to specify these when configuring your Salesforce integration in the FortiMonitor Control Panel.

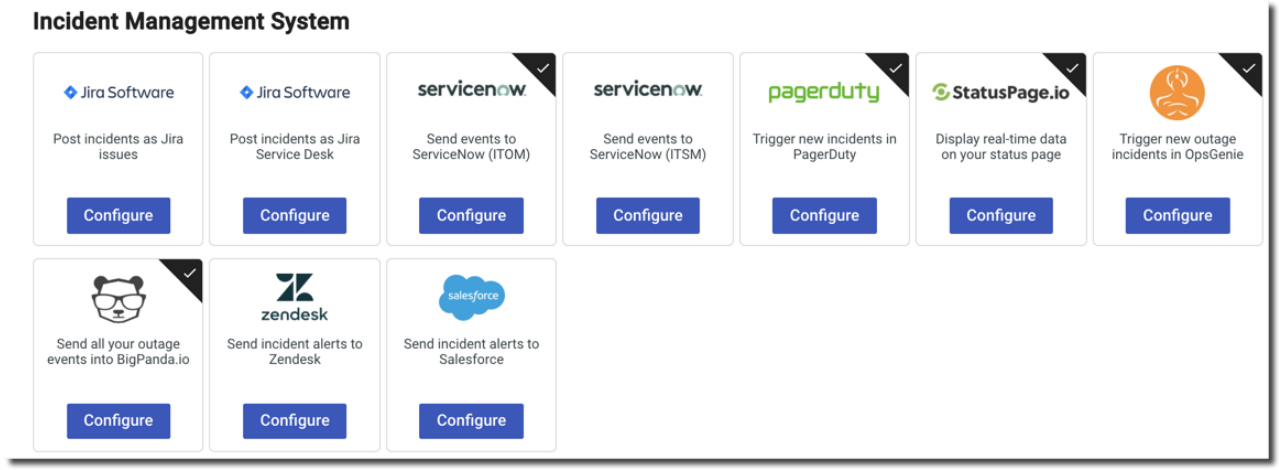
Reset your security token

1. Click your avatar then click **Settings** to open your profile settings.
   Under **My Personal Information**, click **Reset My Security Token**.
2. 

3. Click **Reset Security Token**.
4. You will receive an email with your security token. Copy the token. You will need to specify this when configuring your Salesforce integration in the FortiMonitor Control Panel.

Set up the integration

The Salesforce integration can be set up via the **Integrations** page.

1. Log in to FortiMonitor.
2. From the main navigation header, select **Teams & Activity** > **Integrations**.
   Locate the **Salesforce** card then click **Configure**.
3. 

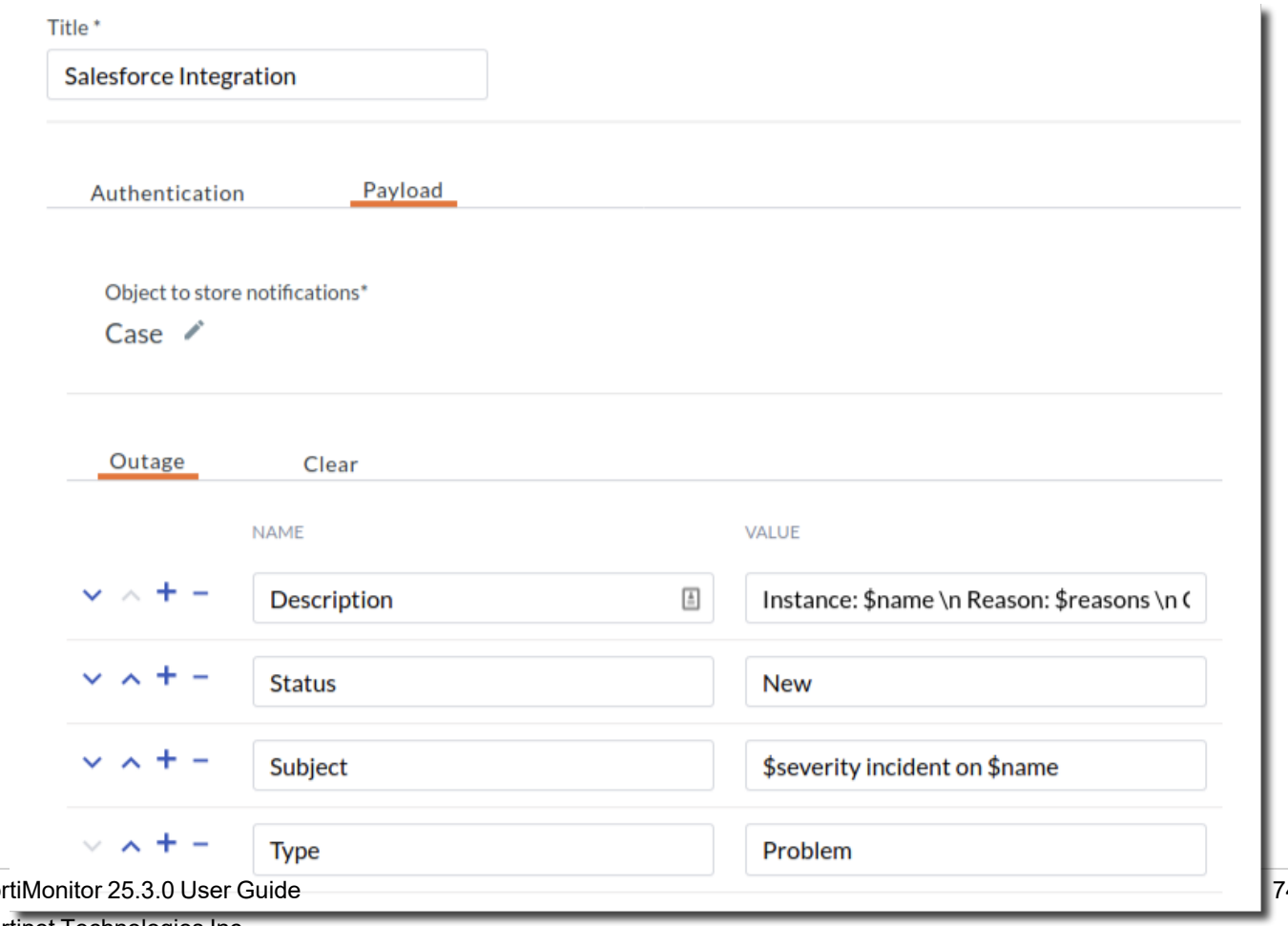
4. Configure your Salesforce integration. The configuration options are detailed in the following table:

Field	Description
Title	Enter a name for the Salesforce integration.
User Email Address and User Password	The email address and password you use to log in to Salesforce.
Consumer Key	The **Consumer Key** from the Salesforce connected app you created. See Configure a new connected app in Salesforce.
Consumer Secret	The **Consumer Secret** from the Salesforce connected app you created. See Configure a new connected app in Salesforce.
Security Token	Security token from Salesforce. See Reset your security token.
Close State	Enter the name of the transition to apply when an incident is resolved. Make sure your Close State value corresponds to a valid status in your Salesforce instance.
Sandbox	Enable this option if you are testing inside a Sandbox instance.

5. Select **Save**.

6. Click the **Payload** tab.

7. Customize the **Outage** and **Clear** incident parameters. These parameters are used for cases that will be created in Salesforce when an incident is detected or is resolved. The parameters you can use are detailed in the following sections.



8. Select **Save Payload**.
9. Go to the **Configured Integrations** tab to verify your Salesforce integration.
10. To set up notifications, configure the integration's Alert Timeline.

Parameters

Parameter	Description
$trigger	The type of event that triggered this payload (outage, ack, broadcast, clear)
$fqdn	The Fully qualified domain name of the server experiencing the incident/clear.
$name	Name of the server experiencing the incident/clear.
$timestamp	UTC timestamp of when the incident/clear occurred.
$event	The type of event, either **incident event** or **clear event**.
$outage_id	The ID number of the associated incident.
$services	For service incident: services experiencing the incident/clear.
$compound_service_id	The ID number of the compound metric affected.
$network_service_id	The ID number of the network service affected.
$resource	For resource anomalies: resources experiencing the anomaly/clear.
$server_resource_id	The ID number corresponding to the resource affected.
$items	Services experiencing the incident/clear or resources experiencing the anomaly/clear.
$reasons	The reasons for network service incidents or the details for anomalies.
$duration	The duration of incidents/anomalies which will be filled in on-clear.
$tags	The tags for the server.
$server_key	The server key for the server.
$server_id	The ID number of the server experiencing the incident/clear.
$partner_server_id	The partner server id for the server.
$item_type	The service type textkeys of the services experiencing the incident/clear, or plugin_textkey/resource_textkey combinations of the resources the experiencing the anomaly/clear, or the OID name of SNMP resources experiencing the outage/clear.
$custom_attribute	You can pass custom server attributes that are set on your servers. Use the attribute type as the key.
$alert_label	Alert label of the incident/anomaly.
$severity	The severity of the outage/anomaly, either **critical** or **warning**.
$metric_tags	The tags for all of the metrics involved in the outage.

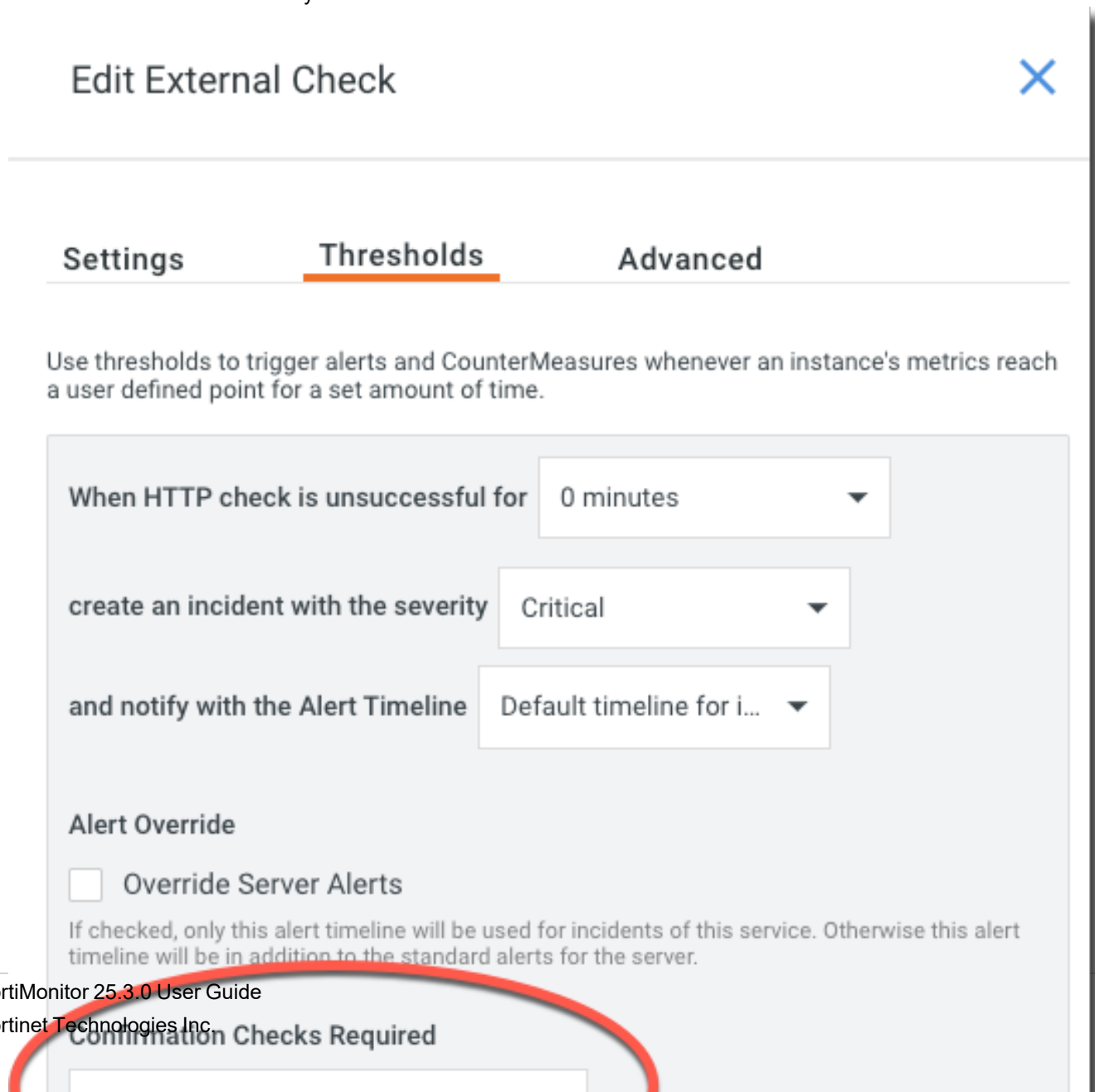
$location	A comma-separated list of the primary monitoring probe names for all network services affected.

Limit the number of confirmation checks for an outage

By default, whenever an outage is discovered, our system will run five confirmation checks from different locations to ensure that this outage is accurate. You can lower the number of confirmation checks for any of your network services checks. You may want to do this if you would like to be alerted for localized network connection failures.

Accounting for AnyCast IP is a good reason to use this feature. If you are monitoring a service that uses an AnyCast IP, it is possible that the confirmation check can actually reach a different POP which is up, thus cancelling a real outage. To prevent this, you can reduce the number of confirmation checks needed and whitelist the other nodes which you are certain will reach the same POP. This is a very effective way of monitoring AnyCast infrastructure.

In order to have access to this feature, you will need to e-mail our support team and request it. Once it is enabled, edit the network service check on which you would like to limit the number of confirmation checks.



On the **Thresholds** tab, you will see a **Confirmation Required** dropdown. Select however many confirmation checks you would like, and then click **Save**. From now on, this network service check will only perform the number of confirmation checks you have specified.

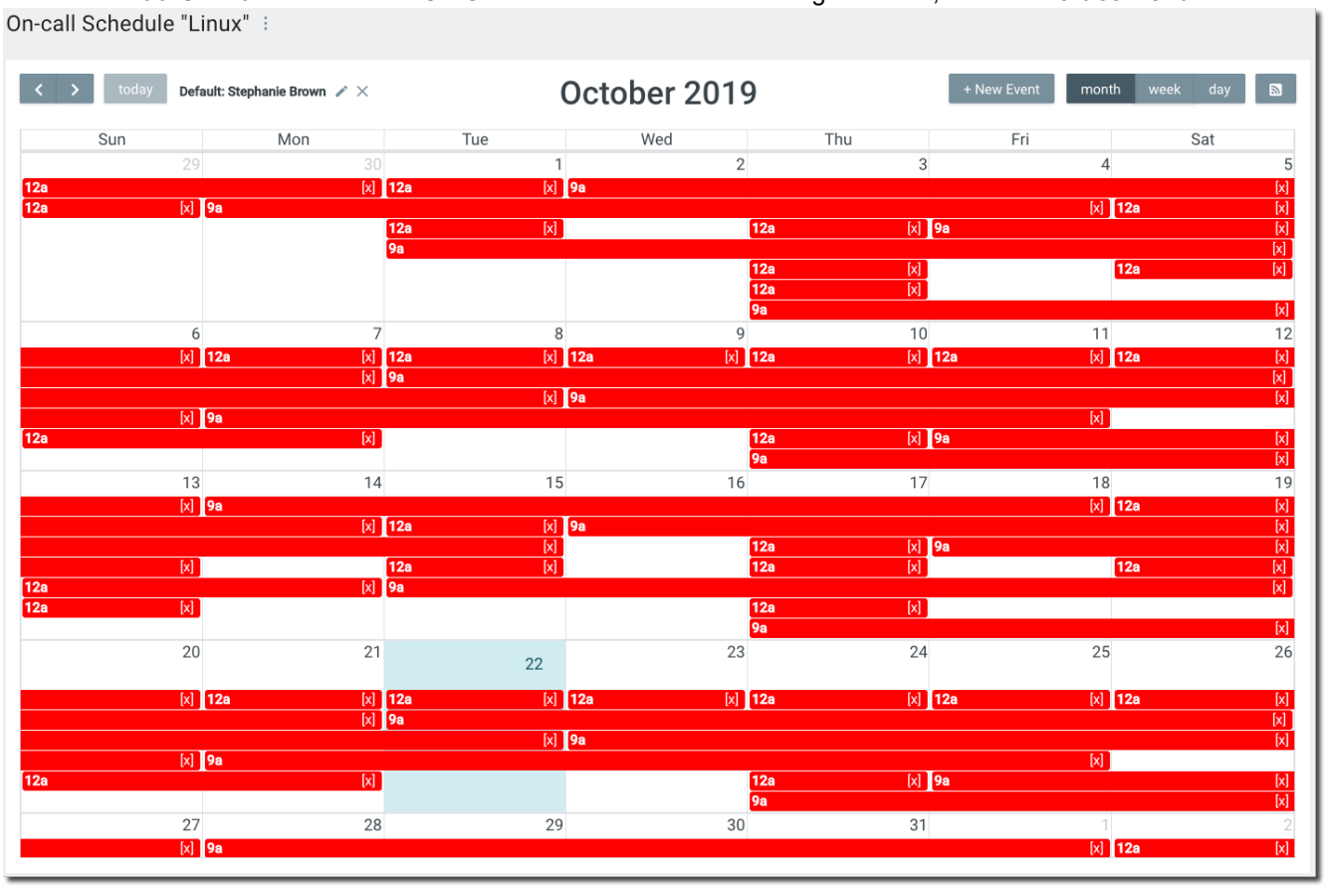
Set up an On-Call schedule

On-Call schedules allow you to set dynamic alerting behavior based on your specific on-call rotations within your organization. Schedules can be set to any level of granularity.

Create an On-Call schedule

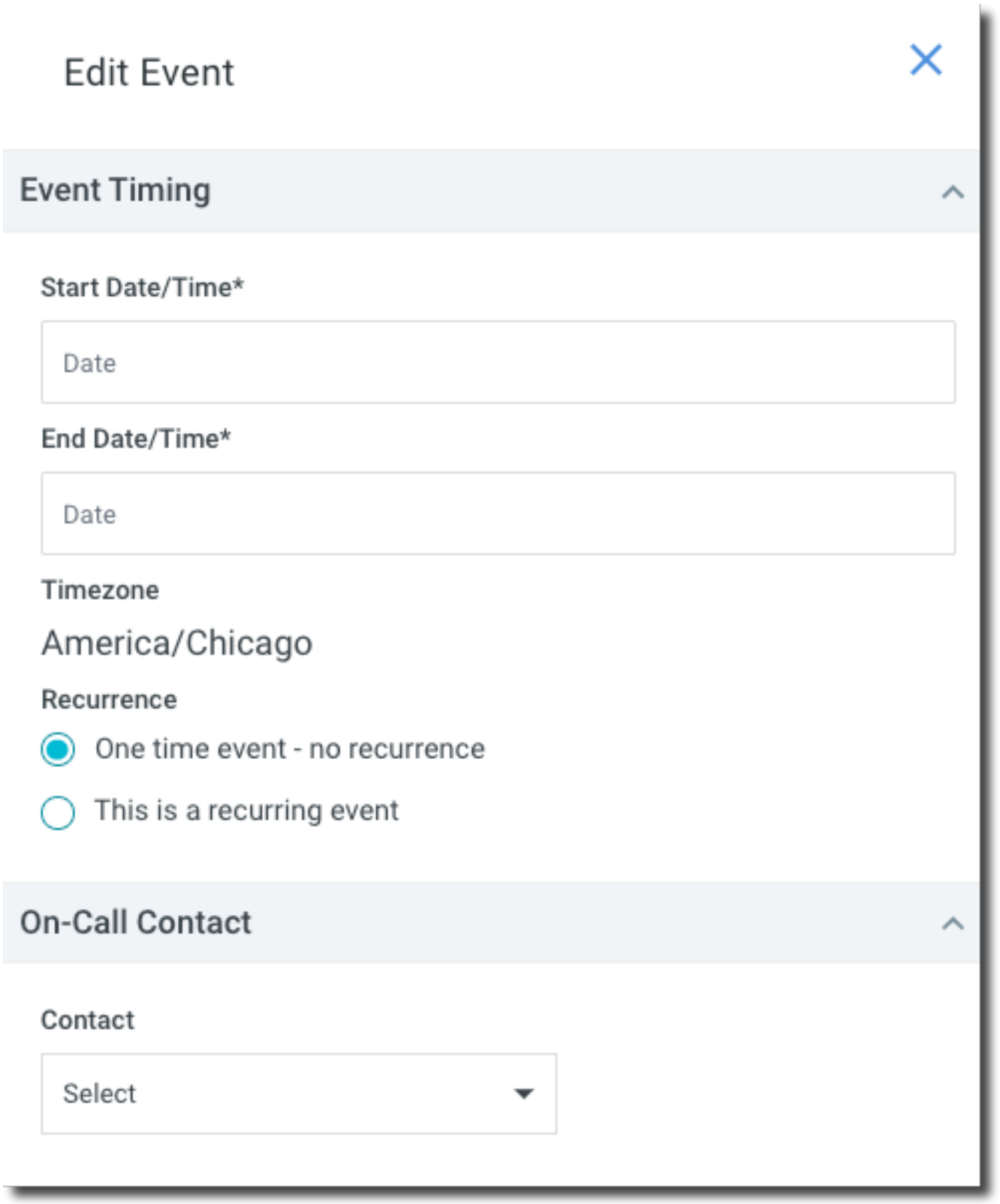
To create or edit an On-Call schedule:

1. From the navigation menu, select **Teams & Activity** > **Users, Groups & On-Call**.
2. Select the **On-Call schedules** tab.
3. Select **Add On-Call** to add a new On-Call schedule. To edit an existing schedule, select the **3-dot menu**.



Add On-Call events

You can add On-Call events by clicking and dragging on your desired date range or by clicking **New Event**. The **Edit Event** drawer will slide out where you can configure your event.


Edit Event
Event Timing
Start Date/Time*
Date
End Date/Time*
Date
Timezone
America/Chicago
Recurrence
One time event - no recurrence
This is a recurring event
On-Call Contact
Contact
Select

Recurring events

By default, events are scheduled to occur a single time. The more common situation, however, is to configure recurring events.

When the create event pane opens, set your event start and end times. This will define how long each recurring event lasts. Next, change the `Recurrence` option from `One Time` to `This is a recurring event`. From there, you can choose a number of recurrence options:

- **Every** `n` days: beginning on the day of your event start date, an event will occur every `n` days from that point
- **Every** `n` weeks: beginning on the day of your event start date, an event will occur on the selected days of the week, every `n` number of weeks. For example, use `1` for every week, `2` for every other week
- **Every** `n` months on the `nth` day of the month: every `n` number of months, on the date of the month that coincides with the date of your `Start Date`. For example, entering a start date of May 1st every `1` month will schedule an event on the first day of every month
- **Every** `n` months on the `nth` day of the week of the month: Every month, an event will be created on the `n` day of the `nth` week. For example, setting a start date of May 1, 2018, which is the first Tuesday of May, will schedule an event for the first Tuesday of every month.

From there, you merely need to select a user and their contact methods to be utilized for alerting.

Default On-Call contact

At the top of the On-Call schedule, there's a button for `Set Default Contact`. This allows you to select a fallback contact to be utilized in when an incident is identified utilizing an Alert Timeline with your On-Call schedule but no On-Call event is present.

Using On-Call schedules

To utilize your On-Call schedule, you still need to add it to an `Alert Timeline`. Navigate to **Monitoring > Alert Timelines**. From there, either create a new Alert Timeline or edit an existing one. Add a new event on the timeline, and then when configuring it, select your desired On-Call Schedule from the `On-Call Schedules` dropdown.

Example

You added an event in your `Alert Timeline` to trigger after four minutes. Once an incident on an instance utilizing this timeline has been ongoing for four minutes, we'll look at the On-Call schedule to see who is on-call. We'll alert them according to the configured contact methods on the on-call event. If no event is present, the `Default Contact` will be used.

Notifications when rolling on and off

To remind team members that they are rolling on or off-call, an email or SMS message can be sent.

Configuring notifications

1. Visit any On-Call schedule.
2. Next to the name of the On-Call schedule, select the **3-dot menu**.
3. Select `Edit Details`.
4. In the slide-out pane, select the notification types you'd like sent - rolling-on, rolling-off, or both.

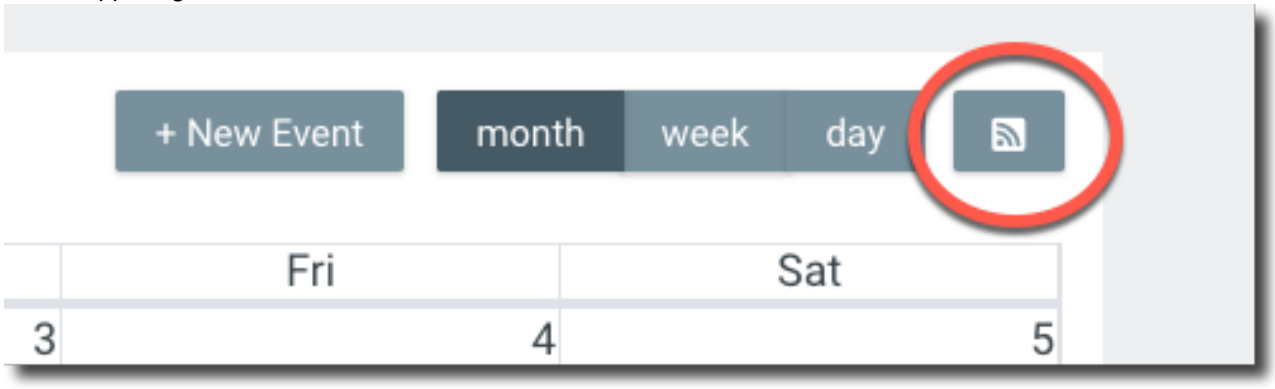
Notification options

Option	Description
Notification Type	SMS or Email. If a user doesn't have one of these contact methods, they will be skipped.
Amount of time...	**Starting**: The amount of time before the user is to start their on-call shift that the message should be sent. Use 0 minutes to notify the user when their shift starts **Ending**: The amount of time before the user is to roll-off on-call that a reminder should be sent. Use 0 minutes to send a message to the user exactly when their shift ends.
Message	The notification content

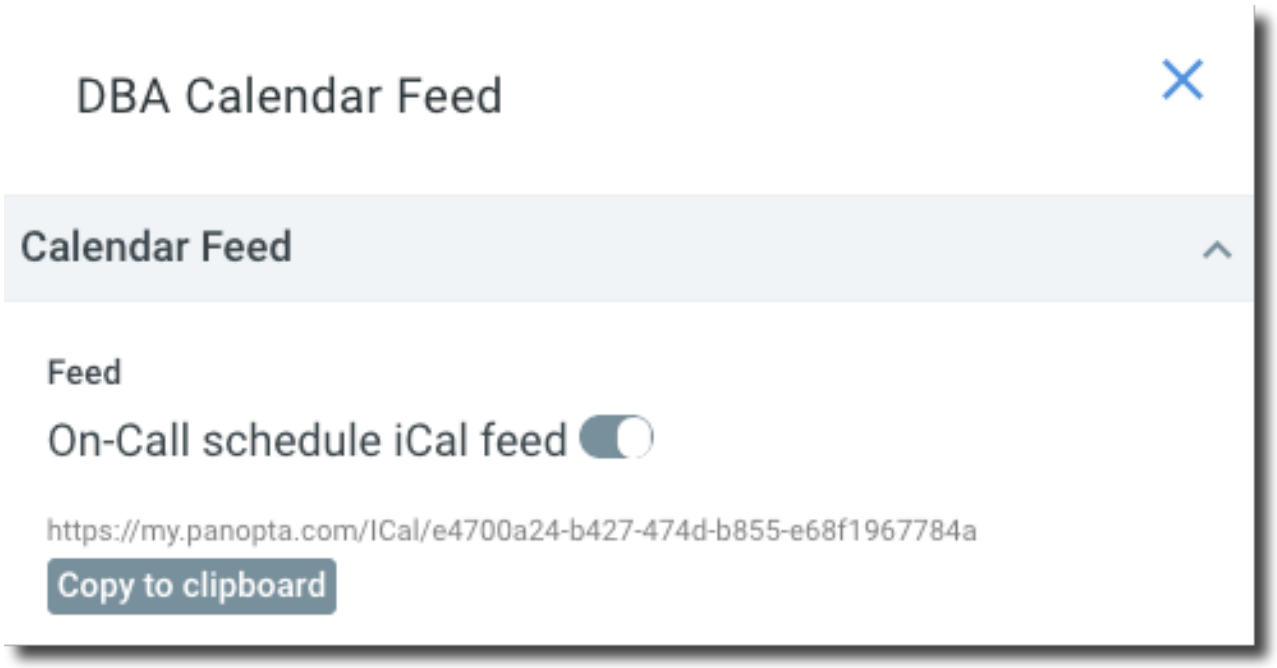
Optionally, you can also customize the email template the notification uses. Under **Settings > Email templates**, you will find **OnCall Notification Conclusion** and **OnCall Notification Intro** - this will allow you to add extra content in addition to the message that is configured above - items such as links to help docs, run-books, wiki's, etc.

Set up iCal integration

On-Call schedules can be viewed in your own calendar by enabling the **On-Call schedule iCal feed** option.

1. Open an On-Call schedule.
   On the upper-right corner of the calendar, select the **RSS feed** icon.
2. 

   The **Calendar Feed** drawer will slide out.



3. Enable the **On-Call schedule iCal feed** option then select **Copy to clipboard.**
4. Paste the copied URL to your own calender application to start subscribing to the feed.

Multiple server outages

When you have multiple servers and network devices that all go down at the same time, the worst scenario that can happen is to have an avalanche of outage emails, forcing you to wade through all of them in order to find and fix the underlying cause.

We try to help you avoid this situation in several ways. By default, our notification system will merge outage alerts for servers that begin at roughly the same time, provided they're using the same Alert Timeline. You'll get one email, SMS, or voice call that lists each of the servers that is down, with full details of each outage.

Also, you can reflect connectivity relationships by setting parent/child dependencies between servers. Whenever an outage for a child server is detected, alerts will only be sent if the parent server is not currently experiencing an outage. Alerts for the parent will include a note about how many children are also down at the same time.

With this functionality, you can setup dependencies such as individual web servers being children of their immediate upstream switch. This way, if your switch fails you'll get an alert saying the switch is unreachable, and that the 20 servers connected to it also are down. Much easier than getting 21 separate alerts and having to dig through them before you find the root cause.

To setup parent/child dependencies, edit each server in the control panel and you can set the parent. If you have lots of dependencies to setup contact us and we can help with a bulk configuration.

Can I notify someone separately just for one service on a server?

In addition to the Alert Timeline attached to a server, you can configure per-service Alert Timelines that are only used for an outage involving the specified service. You can configure the service-level timeline to either override the normal notification that is performed for the server, or to be used in addition to the normal notification.

Many customers use this to bring developers and others responsible for only part of the services provided by a server into the outage response process.

Long-running incidents

Long-running incidents are incidents with a duration of 7 days or more. The confirmation check frequency for these incidents are automatically scaled back after a certain amount of time. The following table describes the changes to confirmation check frequency schedule for long-running incidents.

Incident Duration	Change to Schedule
7 days	Set to 60 minutes
30 days	Set to 1440 minutes
60 days	Pause check
90 days	The underlying metric or metrics will automatically be paused and the incident will be resolved. For Agent, OnSight, Fabric, and SNMP heartbeat incidents, the corresponding metrics that depend on those will also be paused.

For example, if an incident has remained active for 7 days, the confirmation check frequency will be changed to 60 minutes, overriding the configured check frequency for the metric. Once the duration reaches 30 days, the frequency will be changed to 1440 minutes. After 60 days, confirmation checks will be paused.

Note: The frequency changes only apply to confirmation checks. Original checks will still be ran until the 60-day mark where further checks will be paused.

Setting aside alerts

When a large number of alert notifications are generated in a short time, a background process is triggered to set aside these notifications, preventing end users from being flooded with notifications. This is primarily due to throttling limitations that can occur with services like SMS, Slack, and Teams. Additionally, generating emails at scale may cause temporary interruptions from the service provider, as the volume could be flagged as spam-like behavior. This process ensures that new incidents still trigger notifications. Note that this only affects alert notifications, not the incidents themselves.

Parent-child alerting for Fabric and SNMP devices

The following logic changes have been implemented to improve alert visibility for child devices:

- Child incident alerts are triggered normally, except in the following cases:
  - If the parent device is a Fabric or SNMP device and has an active Heartbeat incident, alerts for child incidents are silenced.
  - If the parent device is not a Fabric or SNMP device, but has an active Ping incident, alerts for child incidents are silenced.

Note: Parent/child relationships do not affect the creation of incidents on child devices. Those will always be generated. The only impact is on whether alert notifications for child incidents are silenced based on the conditions above.

Example:

If a FortiGate device has a high CPU usage alert (a non-heartbeat condition) but is otherwise reachable, it will not suppress notifications for incidents on its child devices. Those alerts will be sent as usual.

Previous Behavior

Previously, if any metric on a parent device was in an alert state, all child device incidents were still created, but all alert notifications were suppressed. This may cause critical child alerts to be missed, especially in SNMP and Fabric environments.

Incident Hub

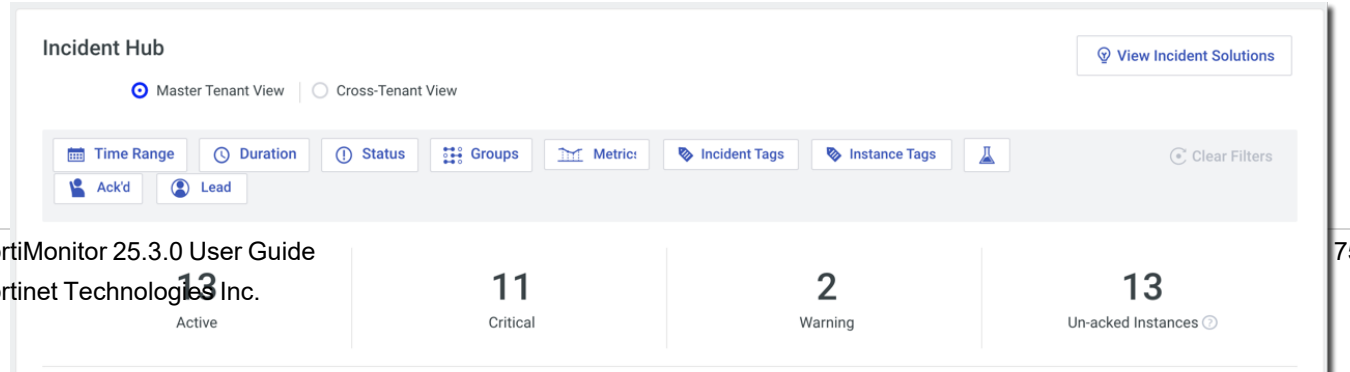
The **Incident Hub** provides teams with a single location to view and manage incidents. A main component of the **Incident Hub**, by default, shows all of the active incidents in your infrastructure. This can be further refined by using filters.

From this page, you can:

- Filter incidents
- View incident details
- Acknowledge an incident
- Set an Incident Lead
- Start a maintenance
- View Incident Solutions
- Switch between Master tenant and Cross-tenant views
- Perform other actions

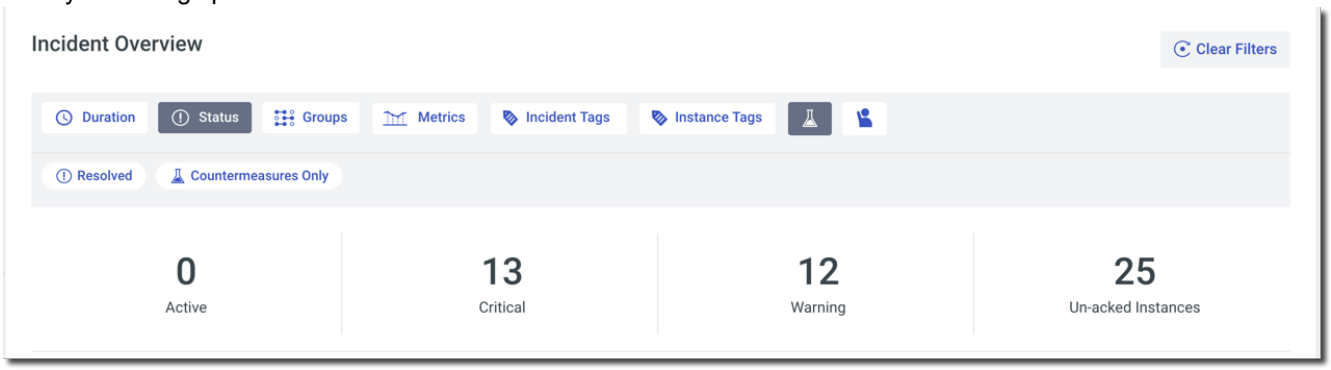
These are further detailed in their respective sections below.

To go to the **Incident Hub**, select **Incidents** > **All Incidents** from the navigation menu.



Filter incidents

When you open the page, only the active incidents are displayed. You can scope the results to specific incidents using a variety of filtering options.



These filters are described in the following table.

Filter	Description
Time Range	Filter the results by time range. **Values**: • All Incidents • Last 15 minutes • Last hour • Last 24 hours • Last 30 days • Custom
Duration	Filter the results by how long an incident has been occurring. **Values**: • > 2 minutes • > 5 minutes • > 15 minutes • > 60 minutes
Status	Filter the results by the current status of an incident. **Values**: • Active • Resolved • Both
Groups	Filter the results by instance group. Clicking the **Groups** filter will provide you with a list of all the instance groups in your infrastructure. Select a single group or **All servers**.
Metrics	Filter the results by metric. Click the **Metrics** filter to select or more metrics to include in the results.

Incident Tags	Filter the results by **Incident Tags**.
Instance Tags	Filter the results by **Instance Tags**.
Severity	Filter the results by severity: **Values:** • Critical • Warning
CounterMeasures Only	Clicking the **CounterMeasures** filter will only include incidents with active CounterMeasures.
Ack'd	Filter by acknowledging user or by acknowledge status. **Values**: • Is • Is not • Is Ack'd • Is Unack'd
Lead	Filter by incident lead or by lead assignment status. **Values**: • Is • Is not • Is Assigned • Is Unassigned

To remove a filter, click **x** on the name of the applied filter. You can also clear all the filters you applied and revert back to the default (show only the active incidents) by clicking **Clear Filters**.

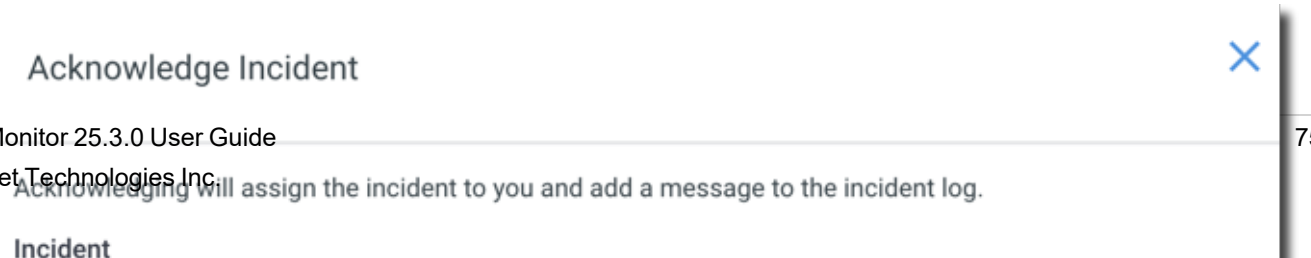
View incident details

Opening or clicking an incident ID opens the incident's details page which shows relevant information that can help you find the root cause of an incident and track its status. See View incident details for more information.

Acknowledge an incident

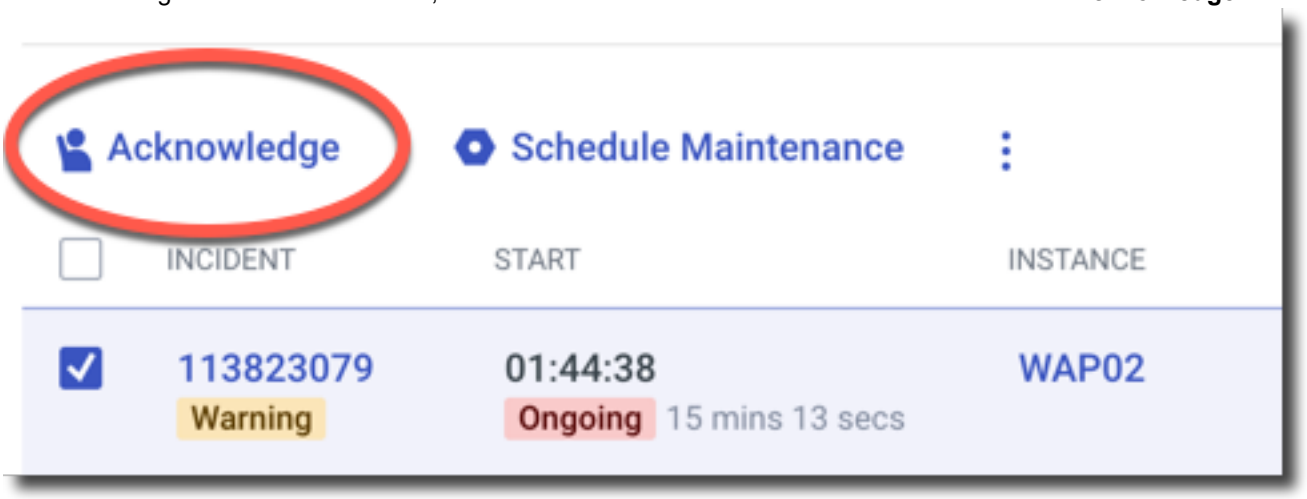
Acknowledging an incident marks the incident with your username and also sets you as the Incident Lead. The following additional options are available:

- Change the Incident Lead
- Add a message to the incident log
- Send a message to all configured users and integrations in the Alert Timeline
- Delay further alerts for a set amount of time
- Cancel all alerts



For more information, see Respond to an incident and delay further alerts.

To acknowledge one or more incidents, select the checkbox of the incident or incidents then click **Acknowledge**.



You can also acknowledge a single incident by clicking the **Acknowledge** icon of an incident as shown below.



Set an Incident Lead

The **Incident Hub** makes it easier for you to see which incidents a team member is working on and which incidents are currently unassigned. Assigning an incident to a team member designates that team member as the Incident Lead. Also, note that the user who acknowledges an incident is automatically assigned as the Incident Lead.

To set an Incident Lead, perform the following steps:

1. Select the **Incident Lead** icon of the incident that you want to assign.

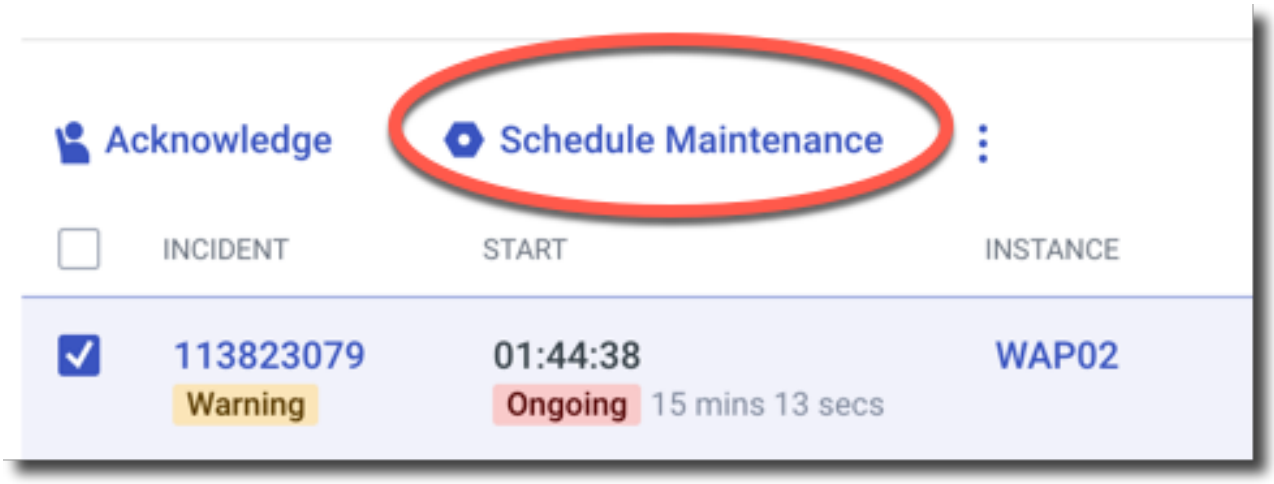


2. Select a FortiMonitor user to designate as the Incident Lead.

Start a maintenance

To start maintenance on one or more affected instances, perform the following:

Select the checkbox of the incident or incidents then click **Schedule Maintenance**.



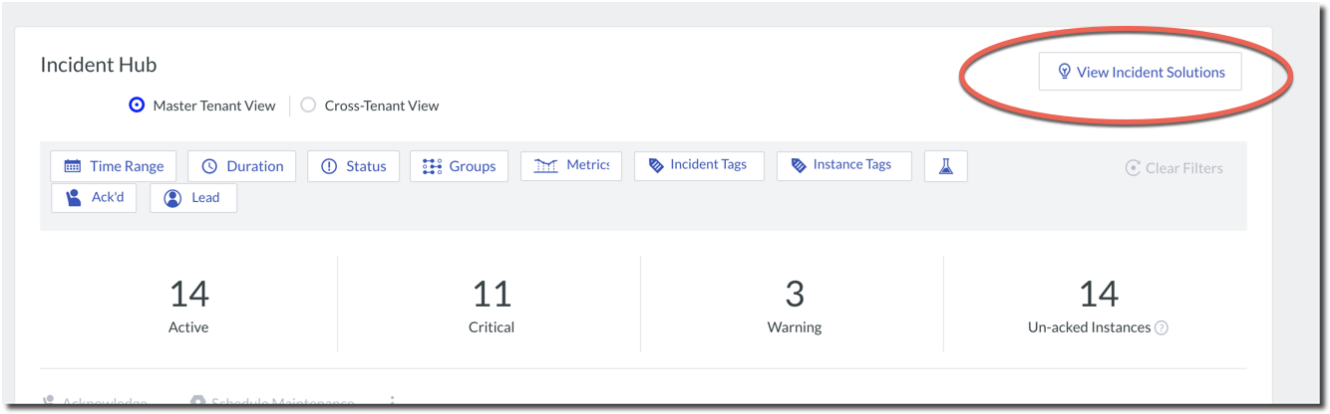
1.
2. From the options, select **Immediate** or **Custom**.

You can also start maintenance on a single incident by clicking the **Schedule Maintenance** icon of an incident as shown below.



View Incident Solutions

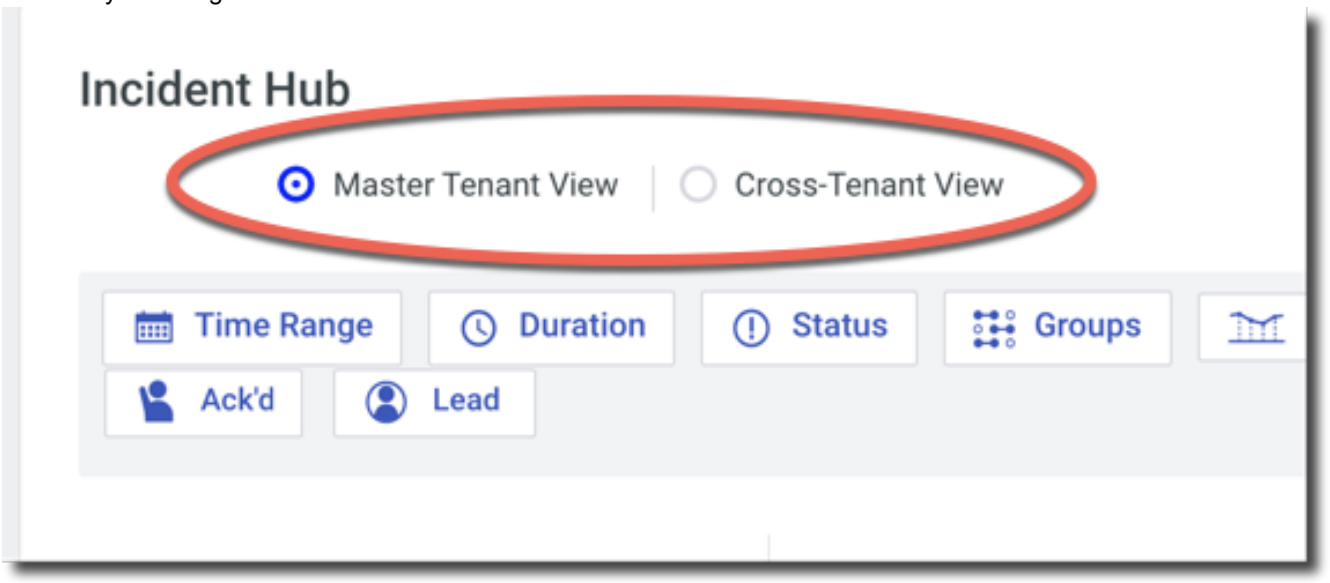
Incident Solutions provide a mechanism to document and disseminate the steps taken to respond to an incident, including details captured during triage and diagnosis phases and the steps to ultimately resolve the incident. For more information, see Incident Solutions.



Switch between Master tenant and Cross-tenant views

This option is for Multi-tenant accounts only.

If you are on a **Master tenant** account, you have the option to view and act on all of the incidents across your sub-tenants by selecting **Cross-Tenant View**.



Perform other actions

Additional actions can be performed on one or more incidents by clicking the **3-dot menu**. These actions are detailed in the following table.

Action	Description

Escalate CMs	Immediately execute the next CounterMeasure action, bypassing the configured delay.
Delay CMs	Delay the next CounterMeasure action for a specified amount of time.
Escalate alerts	Immediately send an alert based on the next event, overriding the configured amount of time in the Alert Timeline.
Delay alerts	Postpone the next event on the Alert Timeline for a specified length of time.
Cancel alerts	Cancel succeeding alerts. No further alerts will be sent.
Exclude from Availability	Exclude one or more incidents from availability calculation.
Send Alert	Manually send an incident alert to one or more users or integrations.

View incident details

Opening or clicking an incident opens the incident's details page which shows relevant information that can help you find the root cause of an incident and track its status.

Key data points

- Start time
- Duration
- Traceroute information for external and synthetic checks (HTTP, HTTPS, etc)
- Detailed metric results
- CounterMeasure output

Available actions

- View relevant incident data
- Tag incidents
- Acknowledge and set an incident lead
- Start a maintenance
- Add messages to the timeline

Incident Header overview

The **Incident Header** section provides a high-level overview of the incident.



The following table describes the fields and options in the **Incident Header**.

Field	Description
Incident ID	The unique identifier of the incident.
Primary FQDN and instance name	The primary FQDN and instance name (if you named the server) of the server where the incident occurred.
Status	The current status of the incident. • Active • Resolved
Severity	The severity of the incident: • Warning • Critical
Incident lead	The team member responsible for the incident. To set the **Incident Lead** for the incident, see Set Incident Lead.
Start time	The start time, date, and timezone of the incident.
End time	The end time, date, and timezone of the incident.
Duration	The amount of time that the incident has been active.
Add Incident summary	Adding an incident summary can help provide context or additional information about the incident. To add an incident summary, click **Add Incident Summary**. **Note**:A template format can be set for the Incident Summary. To have this enabled, contact Support.
Incident tags	To add tags to the incident, see Add incident tags.

You can also perform several actions from the header. These actions are described in the following sections.

Set Incident Lead

You can assign a team member as the Incident Lead by clicking **Set Incident Lead** and selecting a user from the dropdown.



Acknowledge an incident

Acknowledging an incident marks the incident with your username. While acknowledging an incident, you can add a message to the incident log. You also have the option to:

- Broadcast a message to all users and integrations in the Alert Timeline who've been previously alerted.
- Delay further alerts for a set amount of time.
- Cancel all future alerts related to the incident.



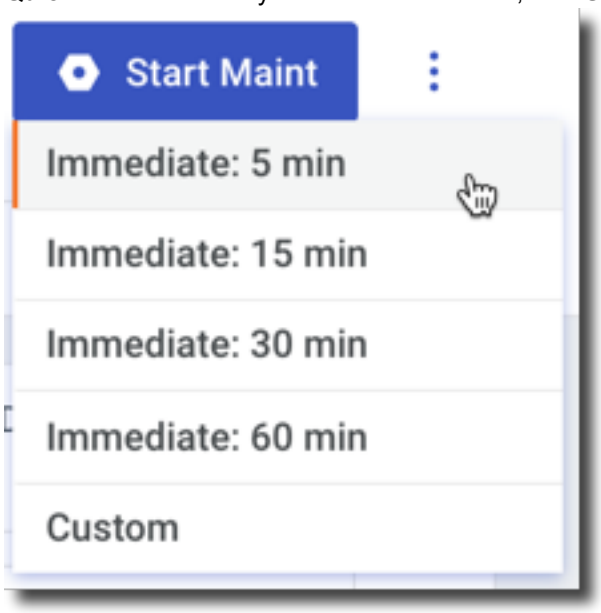
To acknowledge an incident, click **Acknowledge**. The **Acknowledge Incident** drawer, from where you can perform several actions, will slide out.

For more information on acknowledging incidents, see Respond to an incident and delay further alerts.

Start a maintenance

You have two options to put the server affected by the incident under maintenance:

Quick – To immediately start the maintenance, click **Start Maint** then select a duration.

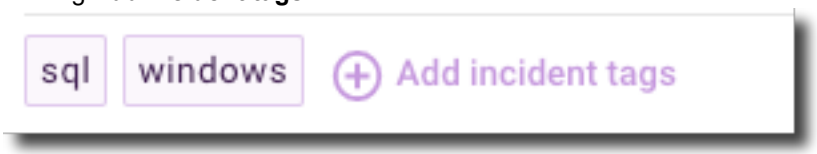
- 

- **Custom** – The custom option allows you to control the handling of the maintenance period, including the timing, scope, and other monitoring options. The options are described in Maintenance Schedules.

To extend/end the maintenance, or if you need more information, see Maintenance Schedules.

Add incident tags

Tagging an incident provides a way to categorize and filter incidents in your infrastructure. You can add tags by clicking **Add incident tags**.



Additional actions

Clicking the **3-dot menu** will show the rest of the actions that can be performed on the incident. The following table describes these actions in detail.

Action	Description
Include in availability	Include this alert to your instances' availability calculation.
Exclude from availability	Exclude this alert from your instances' availability calculation.
Disable all clear	Disable the all-clear alert notification. For more information, see Disable all-clear alerts.
Delay Alerts	Postpone the next event on the Alert Timeline for a specified length of time. • 15 minutes • 30 minutes • 1 hour • 2 hours
Escalate Alerts	Immediately send all scheduled alerts on the Alert Timeline. This means that even if you have configured the Alert Timeline to send an alert after 10 minutes, you can override this and send the alert earlier.
Cancel Alerts	Cancel all succeeding alerts. No further alerts will be sent.
Send Alert	Manually send an incident alert to one or more users or integrations.

Event panel

A single incident may trigger several associated events. The **Event** panel shows the current status of an affected metric as an event. From here, you can see details such as when the event started and for how long the event has been occurring. CounterMeasure action output can also be viewed from this panel.

Each event panel can be expanded (or collapsed) to show more details.

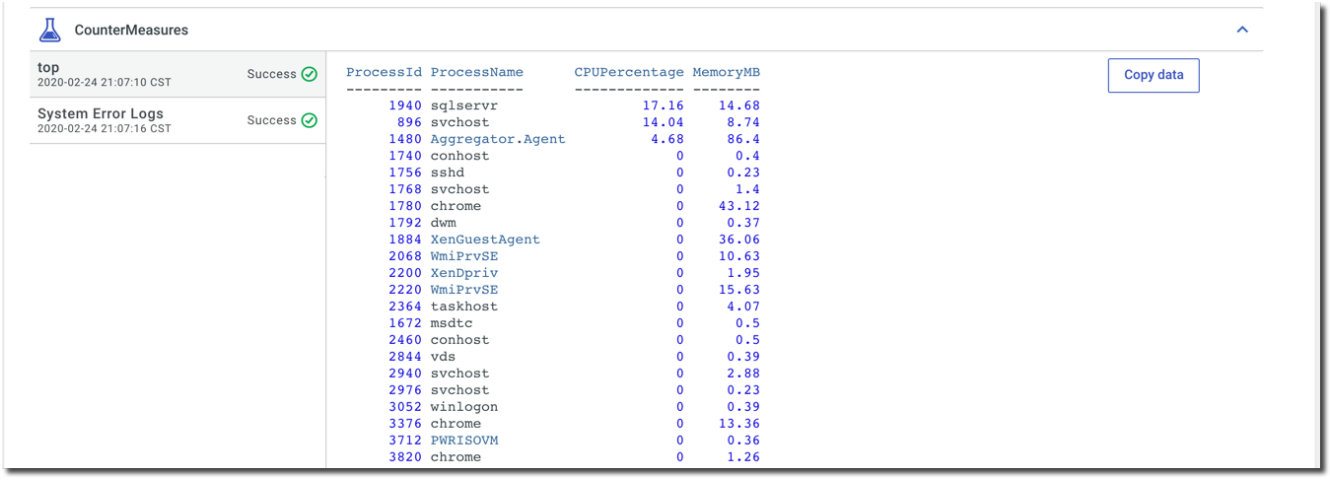


For events associated with network-based services, the check results and traceroute output are shown together with the timeline. The **Traceroute** section also indicates the types of traceroutes ran on the incident.



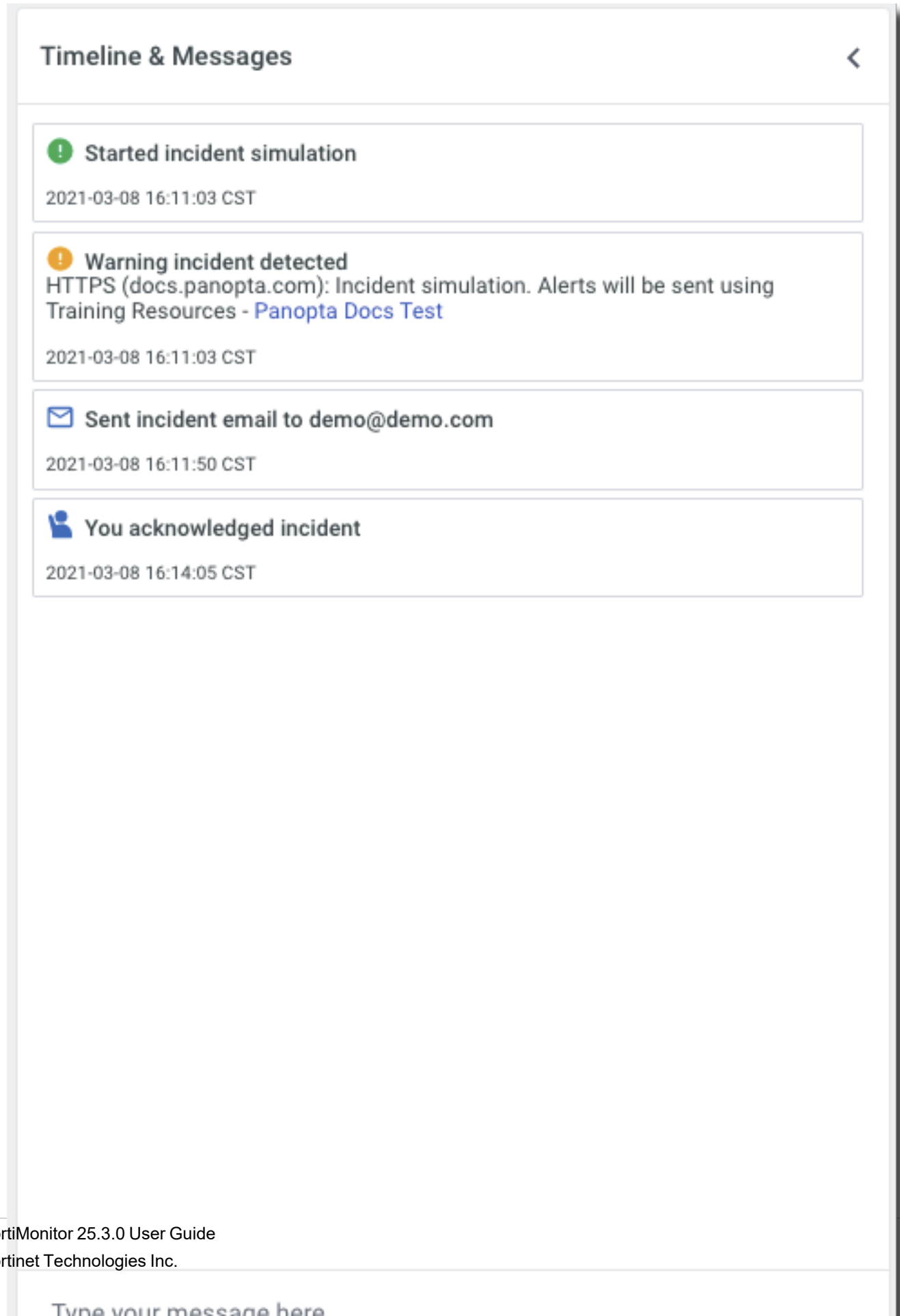
CounterMeasures selection and details

The CounterMeasure is presented in a collapsible panel that details each of the CounterMeasure actions executed on the instance. Click an action to view the output of the CounterMeasure action.



Timeline and Messages

The **Timeline & Messages** section focuses on two things. First is **Upcoming Events**, which lists pending events for the lifetime of the incident. Next is **Messages** which details all of the sent messages or communication related to the incident.


Timeline & Messages
Started incident simulation
2021-03-08 16:11:03 CST
Warning incident detected
HTTPS (docs.panopta.com): Incident simulation. Alerts will be sent using Training Resources - Panopta Docs Test
2021-03-08 16:11:03 CST
Sent incident email to demo@demo.com
2021-03-08 16:11:50 CST
You acknowledged incident
2021-03-08 16:14:05 CST

Comments can also be added to the timeline. Comments can be optionally broadcasted or be made public by enabling the following options:

- Broadcast – The comment is sent to contacts that have previously been alerted.
- Public – The comment is visible in public status pages.

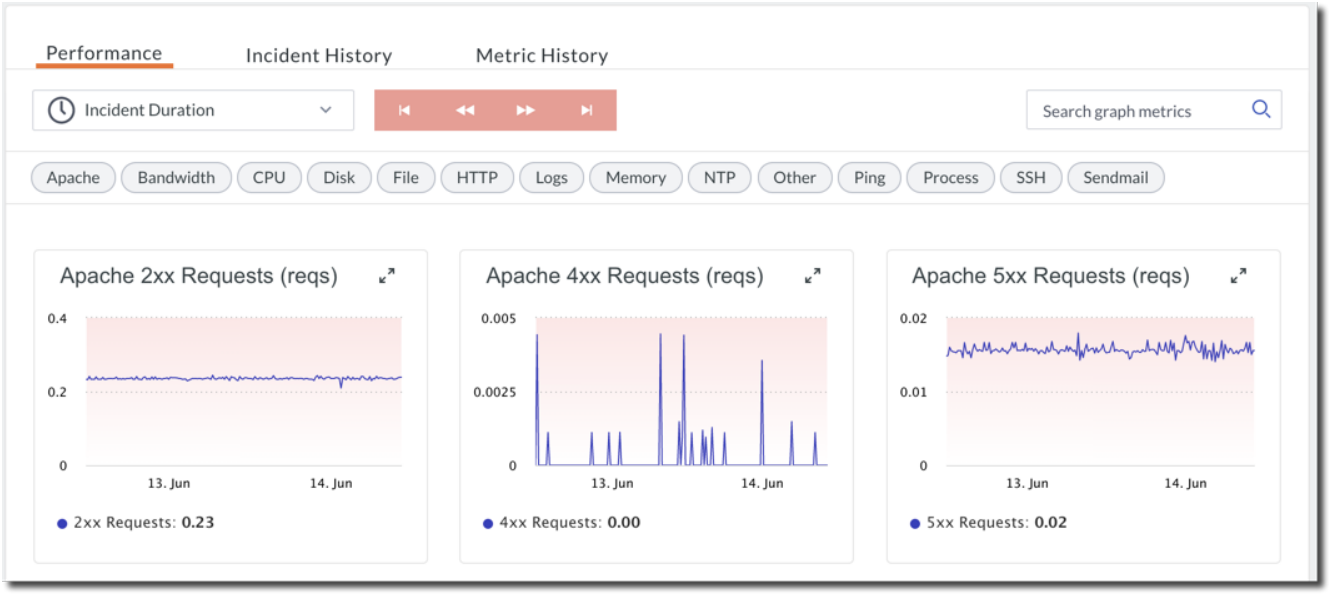
Upcoming Events

The tab also shows a list of events that will be triggered as scheduled. This includes:

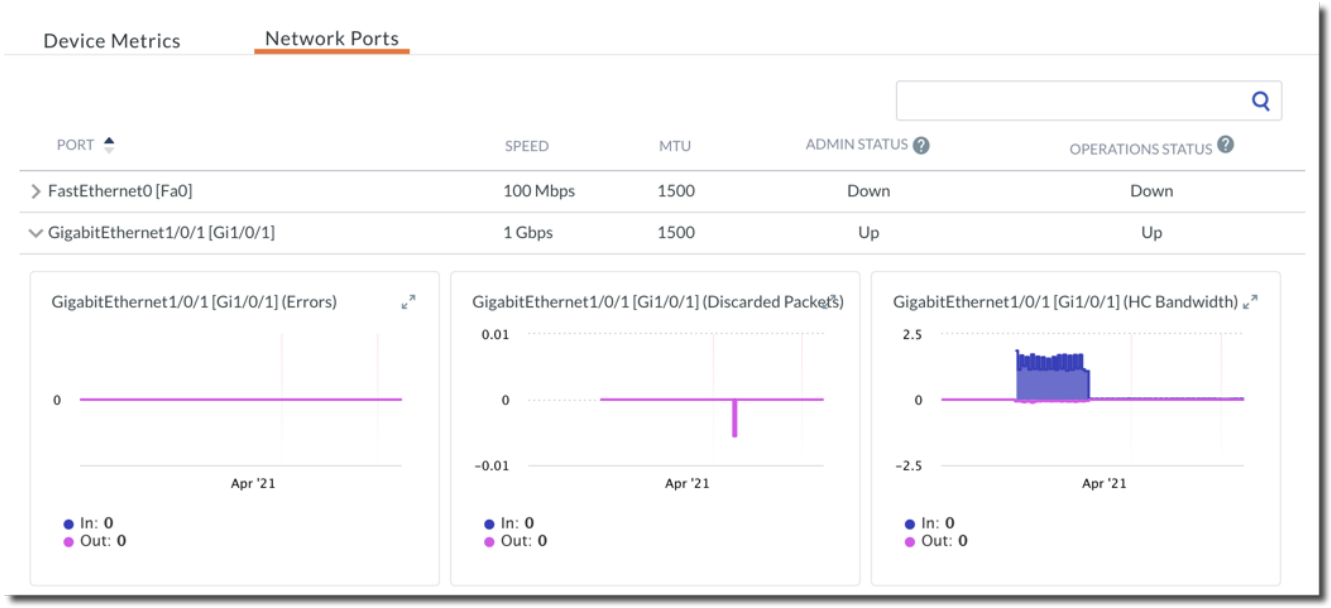
- The next CounterMeasure action to be executed, if any
- The next incident alert based on the instance's Alert Timeline
- Active maintenance
- Upcoming maintenance

Performance

The **Performance** section provides a visual representation of the status of all of the metrics being monitored on the instance. Each metric is represented by a line graph that can be filtered based on the metric type and scoped using a specific time range.

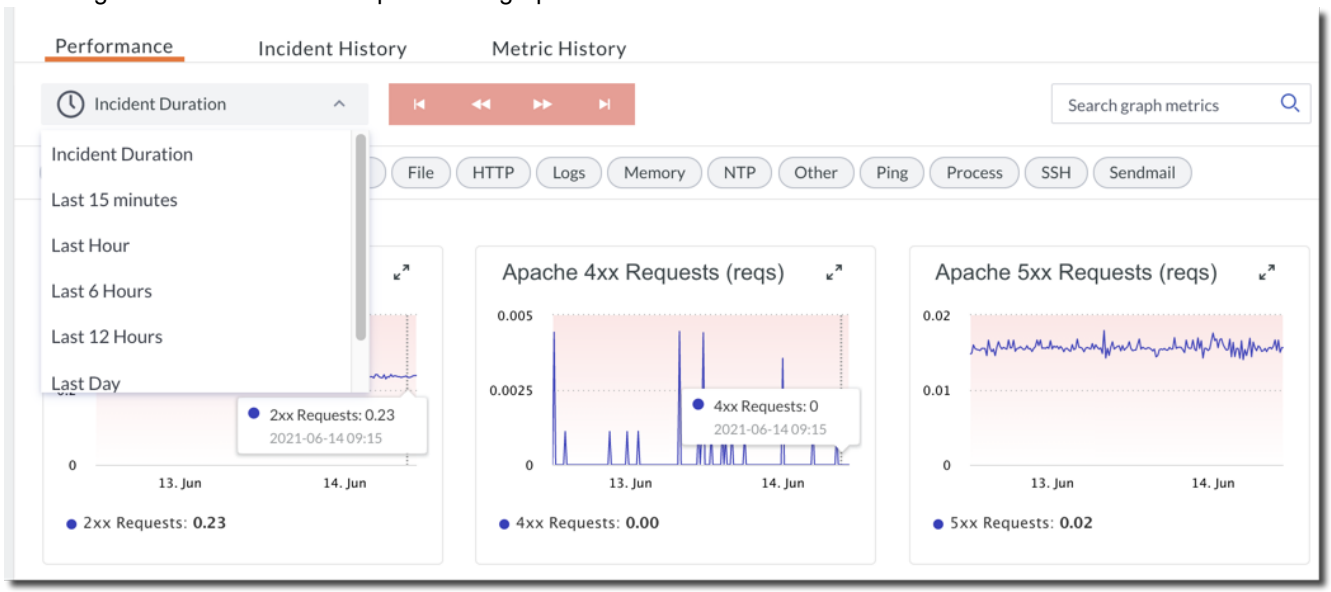


For network devices, the **Network Ports** tab displays the status of each port on the device.



Incident Duration

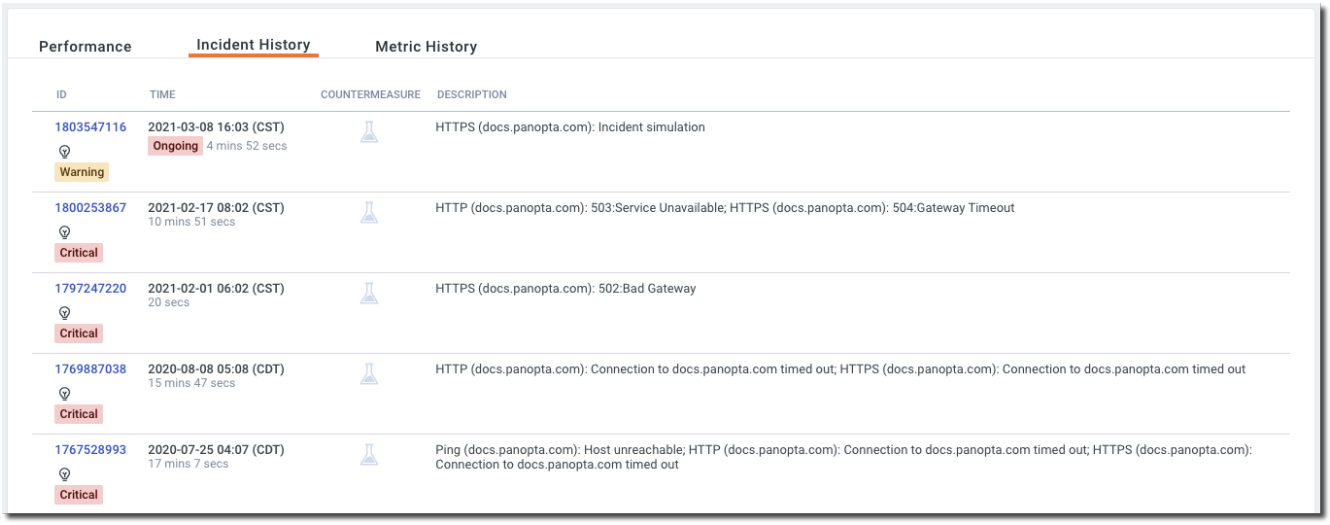
Selecting an **Incident Duration** updates the graphs based on the time selected.



Incident History

All of the incidents that have occurred on an instance are detailed in the **Incident History** section. The Incident ID, the time when the incident occurred, and the description are detailed in the list.

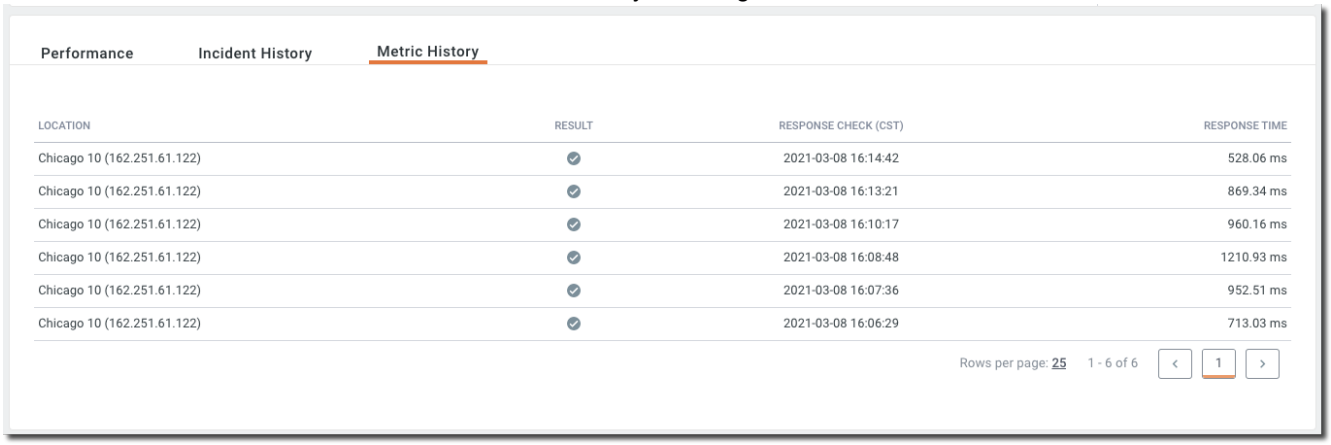
Select **See full history** to view the full list. The list can also be fine-tuned by generating an Incident Report.



Metric History

The **Metric History** tab provides a log of all the metric data collected based on the configured check frequency.

For network service checks, a log of all network service checks from the primary monitoring location and other confirmation check locations. You can refine the results by selecting from the list of available locations.



Incident Solutions

Incident Solutions provide a mechanism to document and disseminate the steps taken to respond to an incident, including details captured during triage and diagnosis phases and the steps to ultimately resolve the incident.

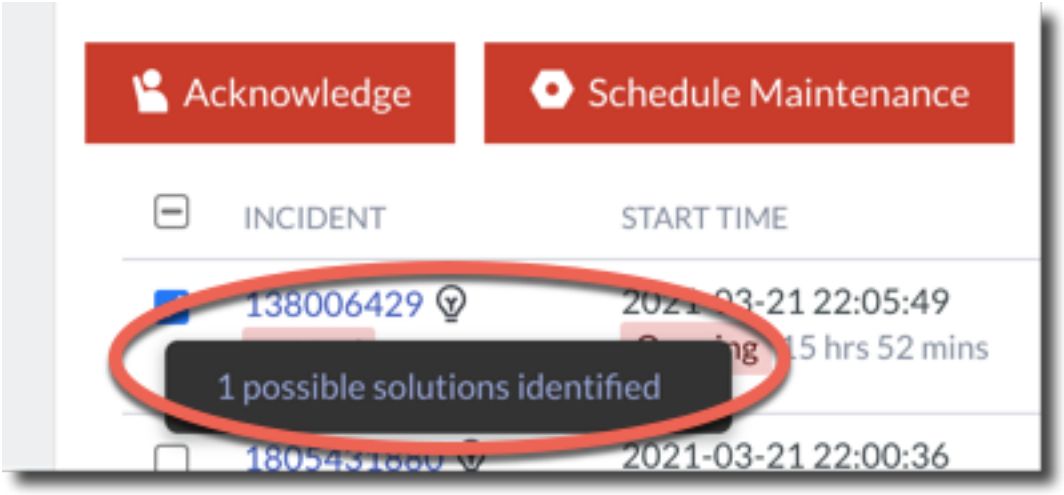
The solution functionality is driven by a collection of content that your team builds as they respond to issues that arise in your infrastructure. As your team captures details related to active incidents, they can be entered as a new solution in the system which will then be analyzed and surfaced as suggestions when similar incidents arise. This allows your team take advantage of previous learnings and to more quickly respond to subsequent problems.

How it works

The Incident Solution functionality is an extension of our existing incident management functionality. Below is a walkthrough of how it works.

1. Incident is detected and alerts are distributed to relevant users based on your alert timelines.

   Throughout the control panel, incidents that have a potentially applicable solution will be shown with a light bulb icon:

2. 

   Clicking the incident opens its details page. A link indicating that a possible solution is available will be shown in the event panel of the metric that triggered the alert.

3. 

4. Clicking the link will display a list of the possible solutions for the incident.
5. Your team can then view more details about any of the solutions and decide which of those to implement.
6. When an incident is resolved:
   a. A user can indicate in the summary that the issue was resolved due to an existing solution.
   b. User can define a new solution.

Apply a solution to an incident

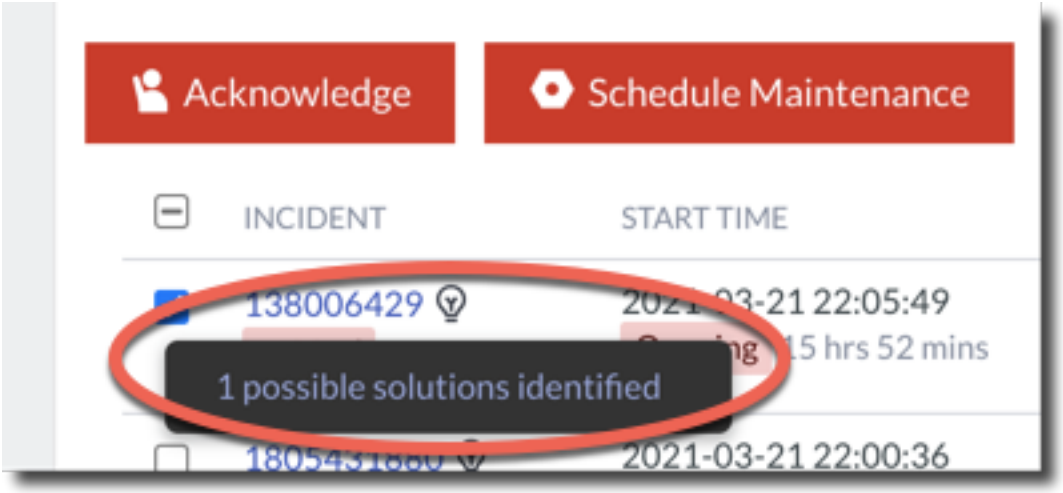
There are three ways to associate a solution with an incident:

- **Use a possible solution** – Use this option to use solutions that we're able to surface based on matching of incident details (instances, metrics, etc.) to previously encountered incidents in the system.
- **Use an existing solution** – You can use this option as a fallback if no suggested incident solution match is available.
- **Create a new solution** – Create a new solution for the incident.

Details on each of these are in the following sections.

Apply a possible solution

1. If a possible solution for an incident is detected, a lightbulb icon will be displayed. Open the incident's details page to review the solution.

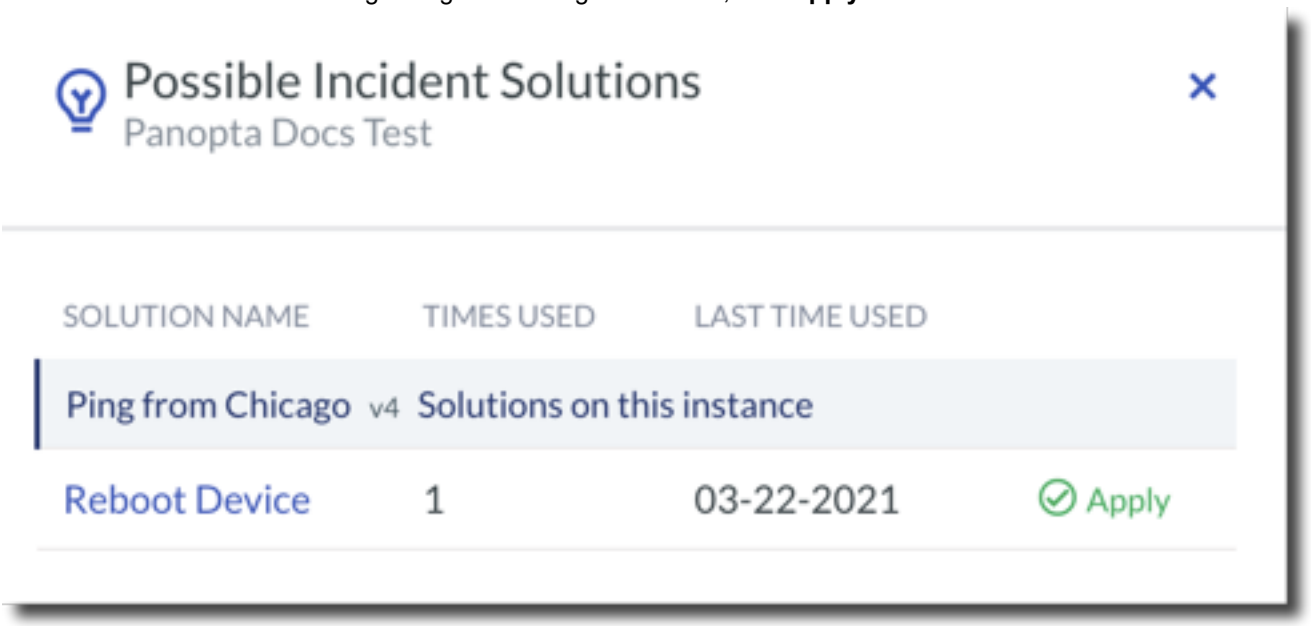


2. Click the **Solutions Available** link from the event panel of the incident to view the solution.



3. Review the solution.

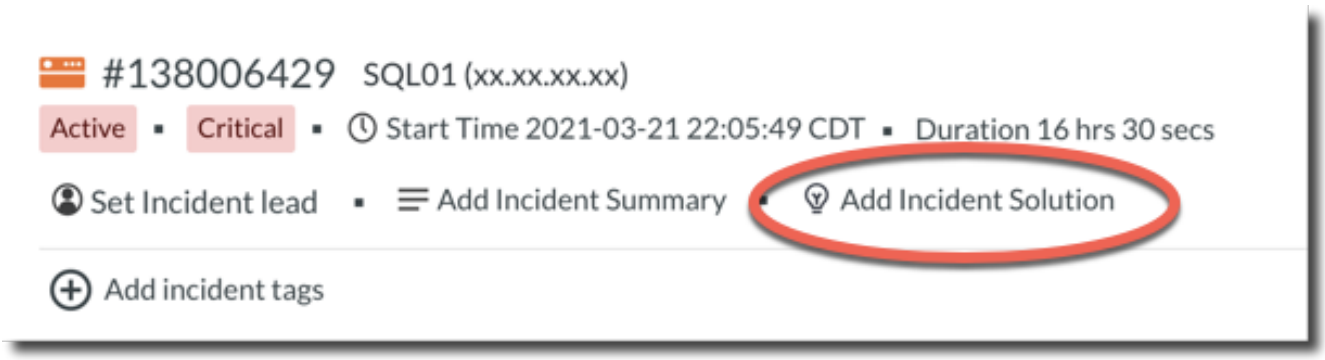
4. If the solution was useful in diagnosing or resolving the incident, click **Apply** to associate the solution with the incident.



Existing solution

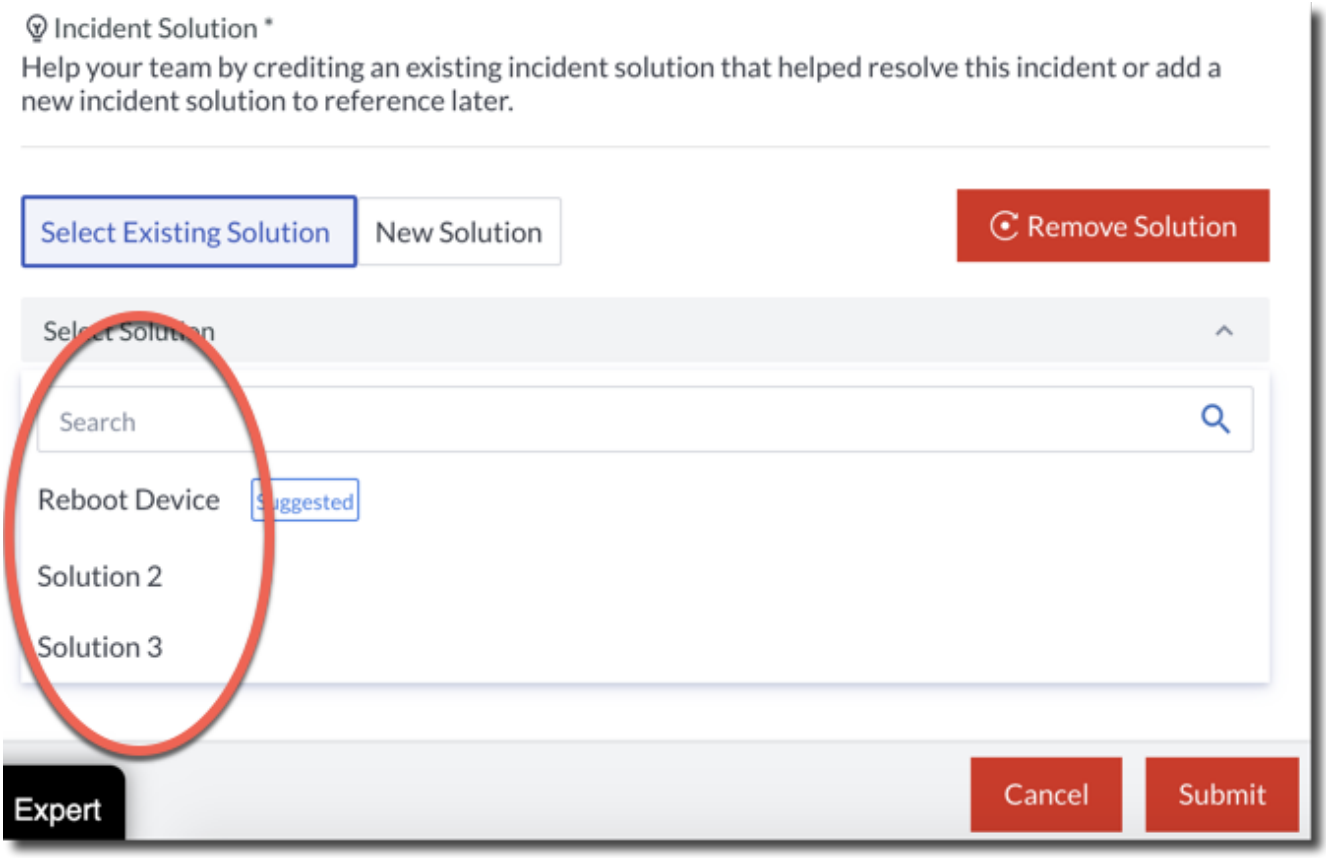
To add an existing Incident Solution to an incident, perform the following steps:

1. Open the details page of the incident where you want to add a solution.
2. Click **Add Incident Solution**.

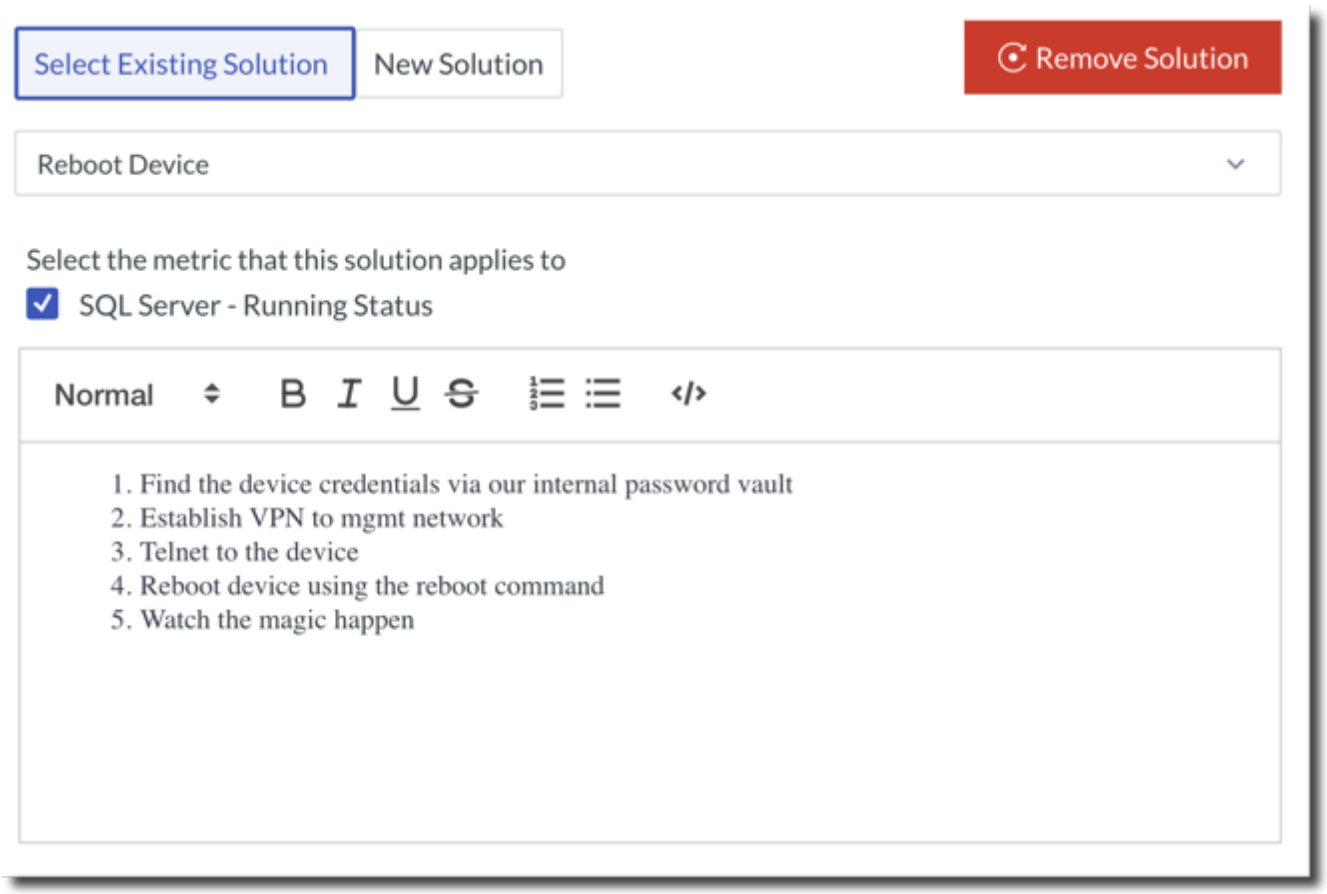


3. Click **Select Existing Solution**.

4. Select the solution that you want to apply to the incident from the **Select Solution** dropdown.

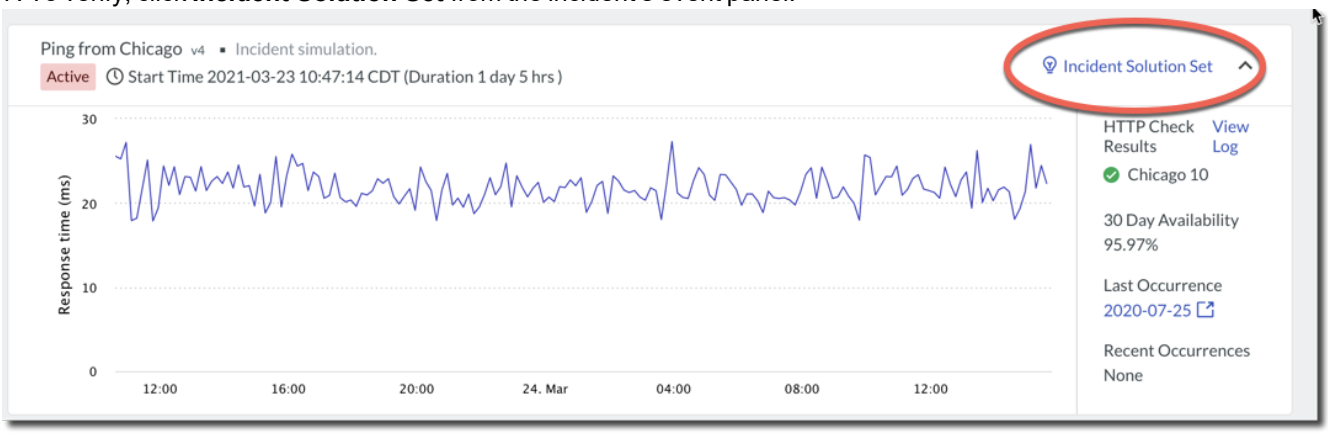


5. Select the metric where you want to apply the solution.



6. Click **Submit**.

7. To verify, click **Incident Solution Set** from the incident's event panel.



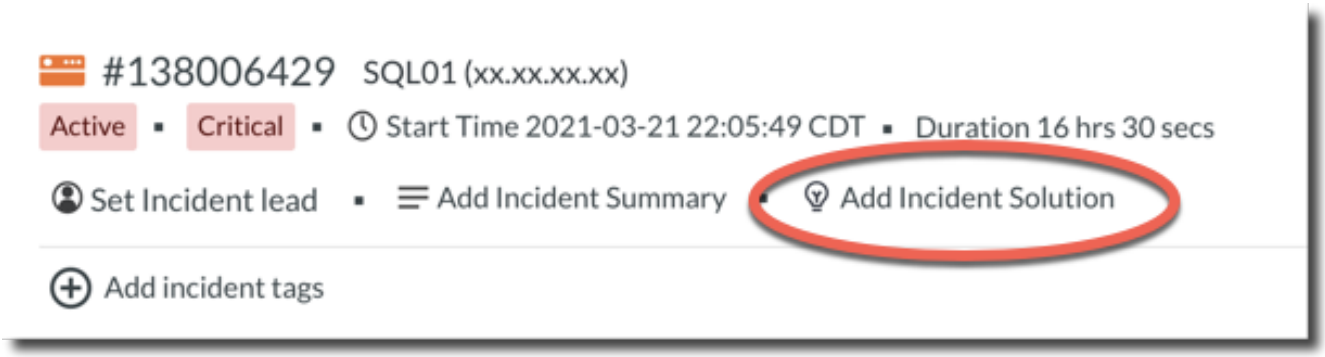
New solution

If the system cannot find an incident solution that matches the parameters of the incident, you can create new incident solution and capture relevant information that can help your team diagnose and solve the incident. Some suggested content for the solution include:

- Steps to diagnose and confirm the incident
- Copy of sample commands and their output
- Link to vendor and third-party documentation
- Other relevant information

To add a new Incident Solution to an incident, perform the following steps:

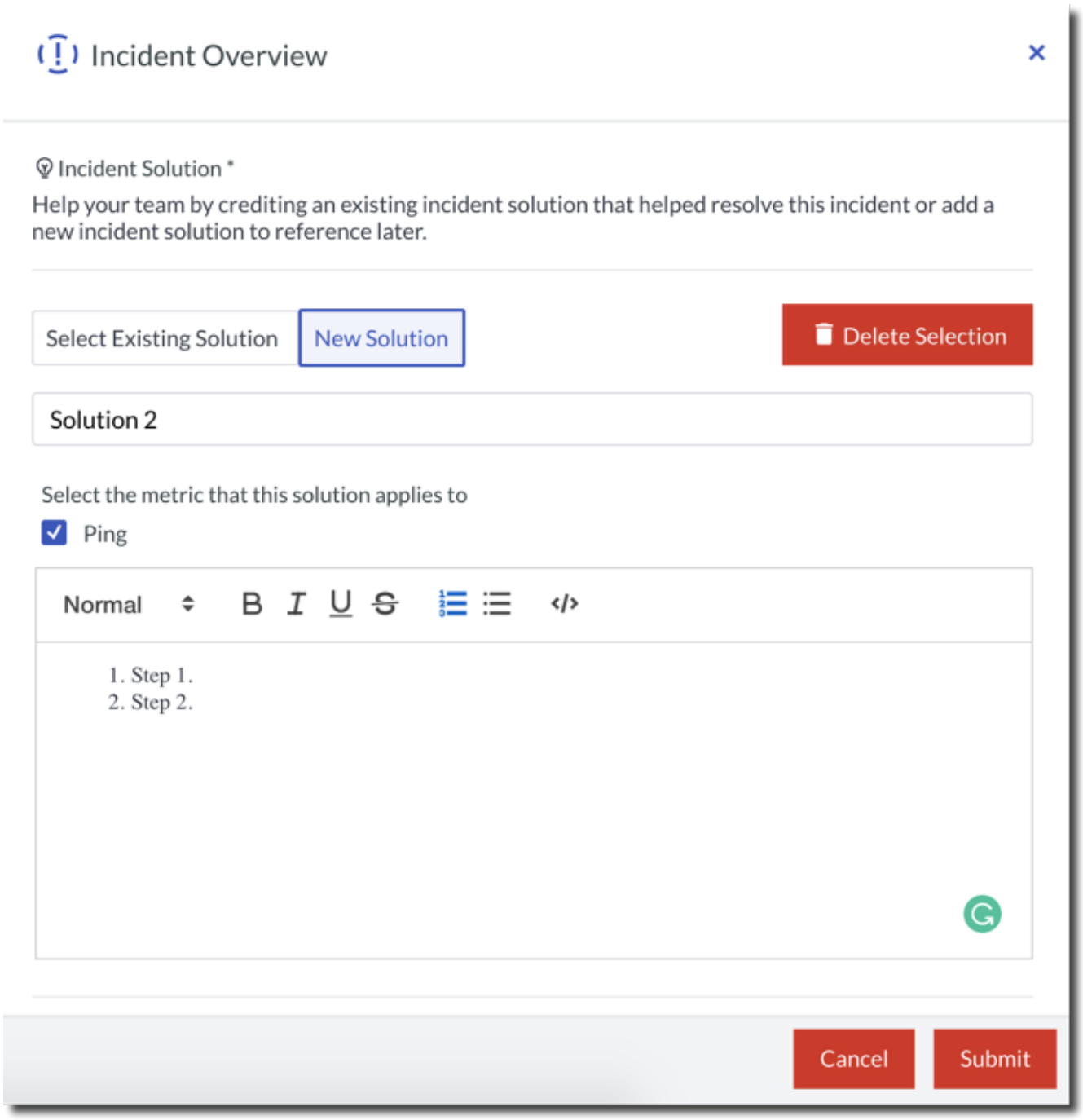
1. Open the details page of the incident where you want to add a solution.
2. Click **Add Incident Solution**.



3. Select **New Solution**.

4. Enter a name for the solution. After providing a name, a text box where you can write the solution will become available.

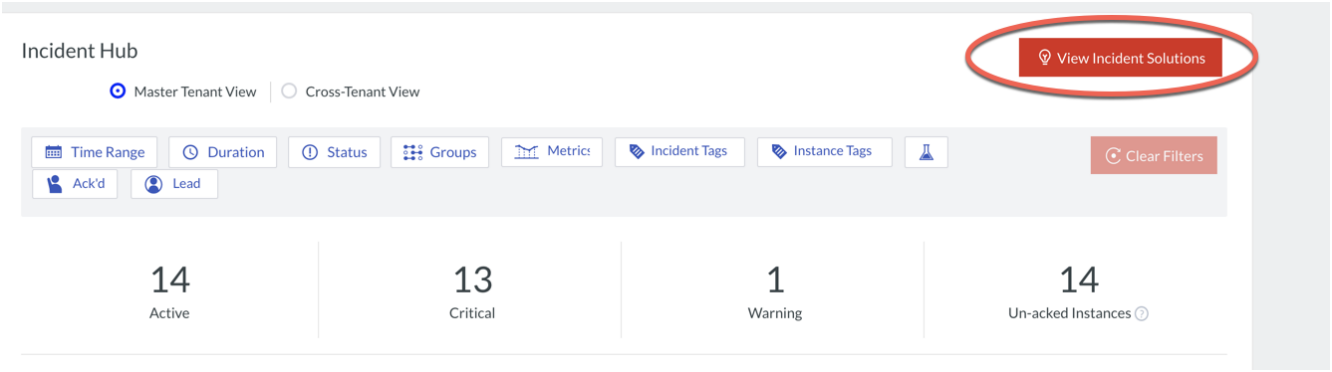
5. Select the metric where you want to apply the solution.



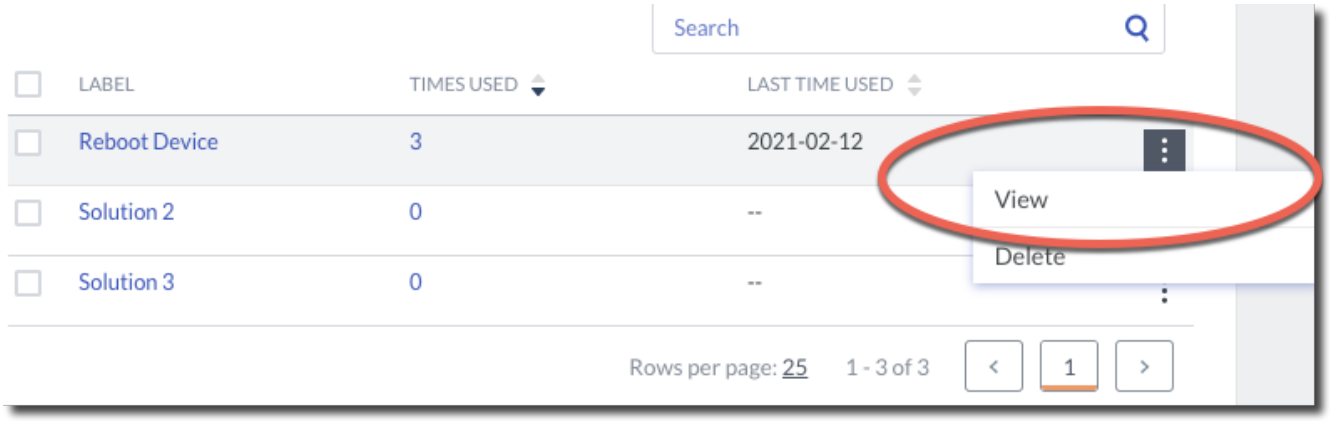
7. Click **Submit**.
8. To verify, click **Incident Solution Set** from the incident's event panel.

Manage Incident Solutions

Incident Solutions can be viewed, edited, or deleted from the Incident Hub. Click View Incident Solutions to show all of the available solutions.

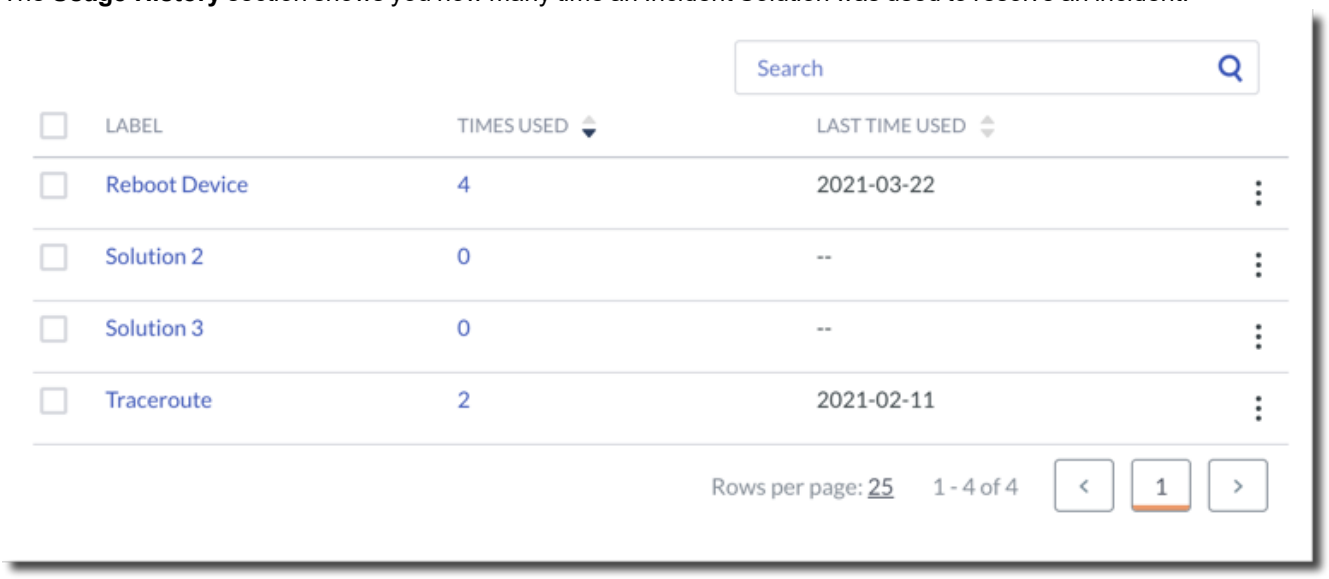


To edit or delete a solution, click the 3-dot icon then click **View** to edit or **Delete** to remove the solution.



Usage history

The **Usage History** section shows you how many time an Incident Solution was used to resolve an incident.



Create an incident using email

Note: This feature is in beta release. Please contact us to have the feature enabled on your account.

You can setup one or more email addresses within FortiMonitor to serve as a gateway for incoming incident emails. Each email gateway will have an auto-generated email address assigned to it, allowing you to target specific instances or different alerting workflows.

Emails with incident information in the header or body can be used to start or end incidents. The email content options are described in Email content options.

Create a custom incident

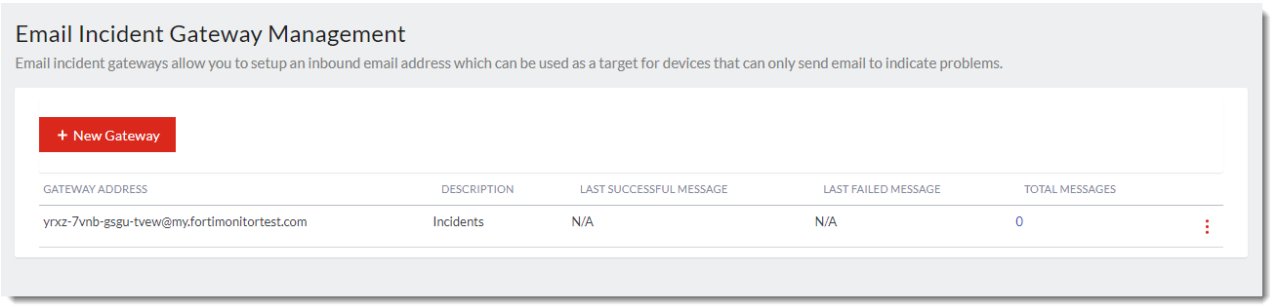
A custom incident is required to be created on each target instance. Once a custom incident is created, take note of the **Plugin Textkey** and **Resource Textkey**. These values are used in the body of incident emails.

To create a custom incident, follow the instructions detailed in Create custom incidents.

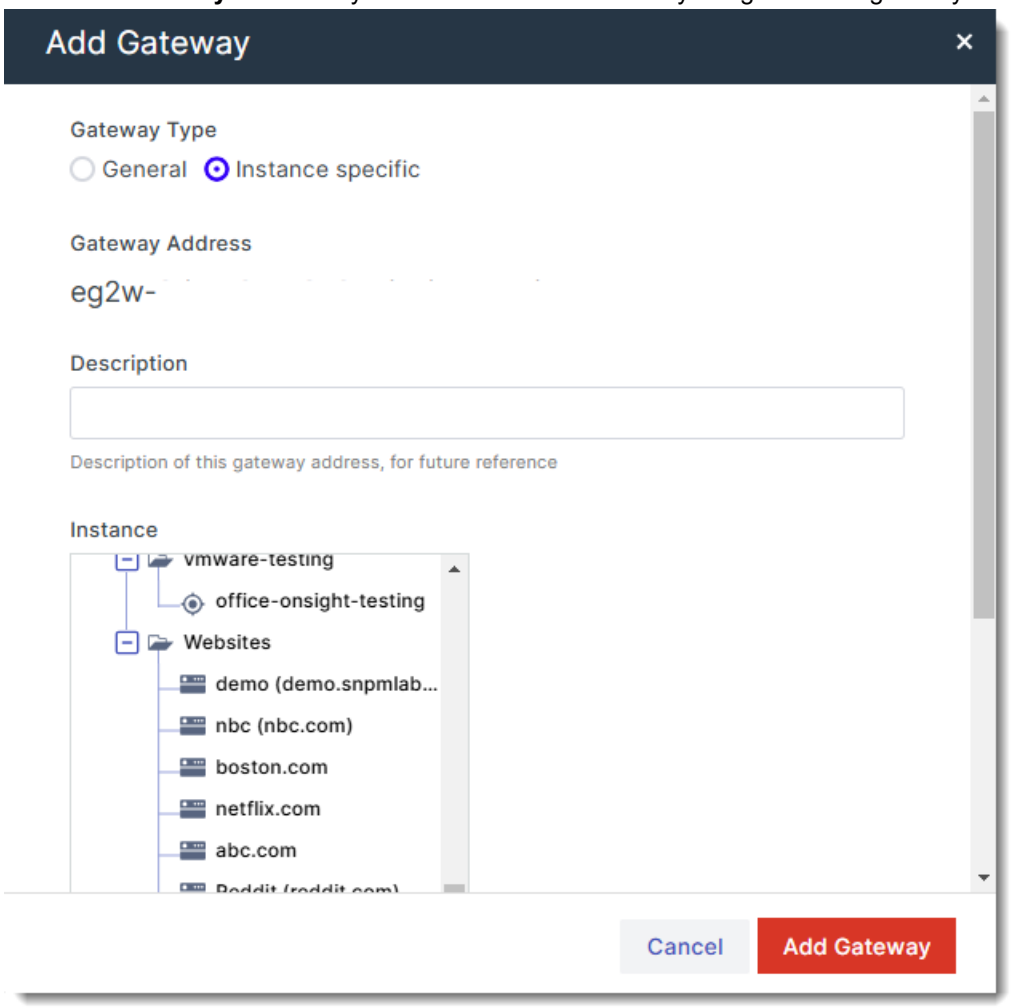
Create an email gateway

To add an email gateway, perform the following steps:

1. From the main navigation header, select **Incidents** > **Email Gateways**. From here you can also see a list of the gateways you have created.



2. Click **New Gateway**. A Gateway Address will be automatically assigned to the gateway.



3. If you select **Instance specific** for the **Gateway Type**, select an instance to associate with the gateway.
4. Add a description for the gateway address.
5. Click **Add Gateway**.

Email content options

Currently, there are two types of supported email content.

- Email headers
- JSON payload

See the following table for more details.

Note: Some proprietary email format (Aviatrix, etc.) can also be processed by the gateway. Contact Support to learn more.

Email Headers	JSON Payload	Description
X-FMN-FQDN	fqdn	The IP or FQDN of the instance to attach the incident to.
X-FMN-Plugin-Textkey	plugin_textkey	The timestamp of when the incident starts or ends. If not provided, defaults to the time when the email message was received.
X-FMN-Resource-Textkey	resource_textkey	The textkeys to use for the custom incident metric that the incident is attached to. Used to tie in to the custom incident filter to pick up alerting criteria. For more information, see Create custom incidents.
X-FMN-Timestamp	timestamp	
X-FMN-Description	description	Free text field to describe the incident
X-FMN-Action	action	Used to indicate if the email is starting or resolving an incident. Available options: • start • end

Usage examples

Email Headers

Email headers can include the following:

```
X-FMN-FQDN: fortinet.com
X-FMN-Plugin-Textkey: com.mysite.email
X-FMN-Resource-Textkey: com.resource.email
X-FMN-Description: "short desc of the issue",
X-FMN-Action: start
```

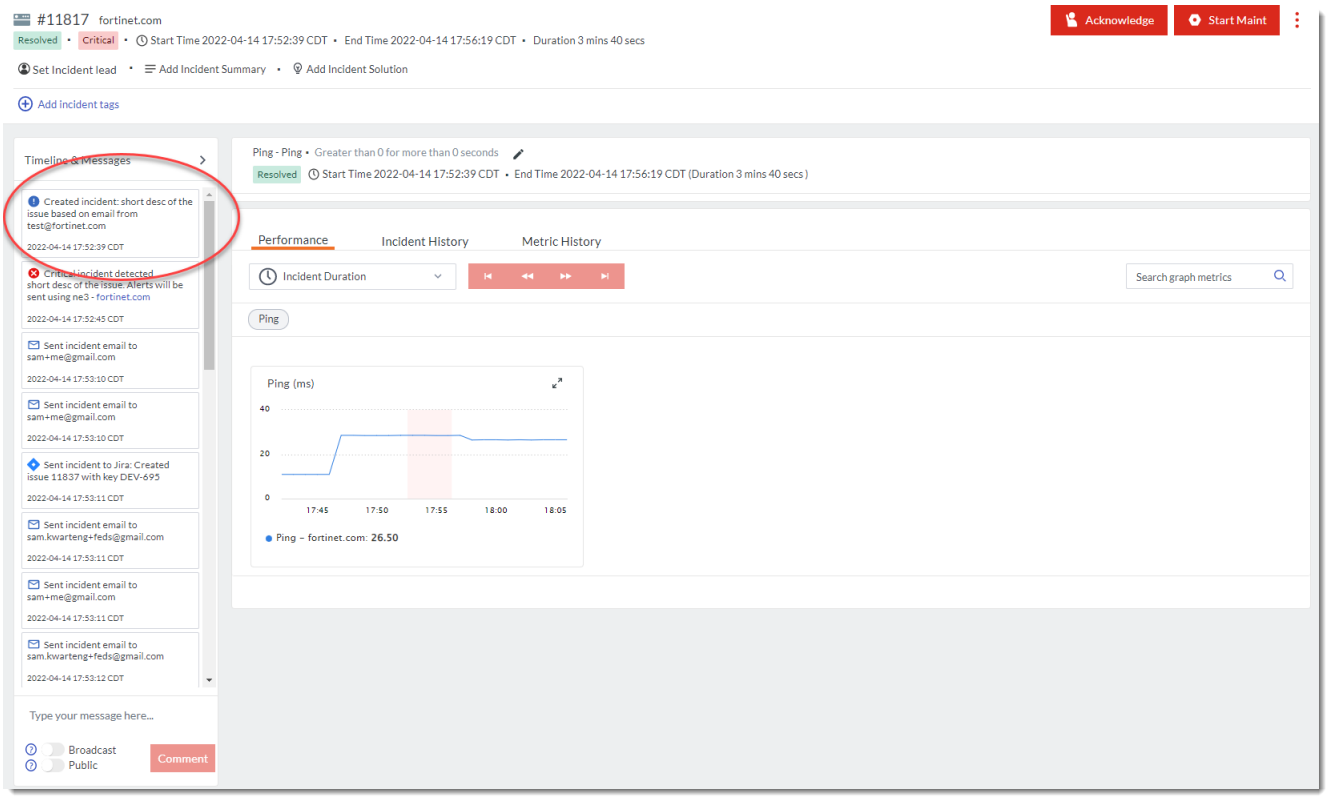
JSON Payload

You can send an email with the body of the email being a JSON dictionary with the following keys:

```
{
   "fqdn": "fortinet.com",
   "plugin_textkey":"com.mysite.email",
   "resource_textkey":"com.resource.email",
   "description":"short desc of the issue",
   "action":"start"
}
```

Incident details

Incidents created by email will be logged in the **Timeline & Messages** feed in the incident's details page. The message includes the incident type (start, end), description, and the sender's email.



View email history

Clicking one of the links under the **Total Messages** column in the **Email Incident Gateway Management** page shows a table view of all messages that have been sent to the gateway, along with their status.



Clicking the **View** link brings up a modal to show the actual email message.

Reports

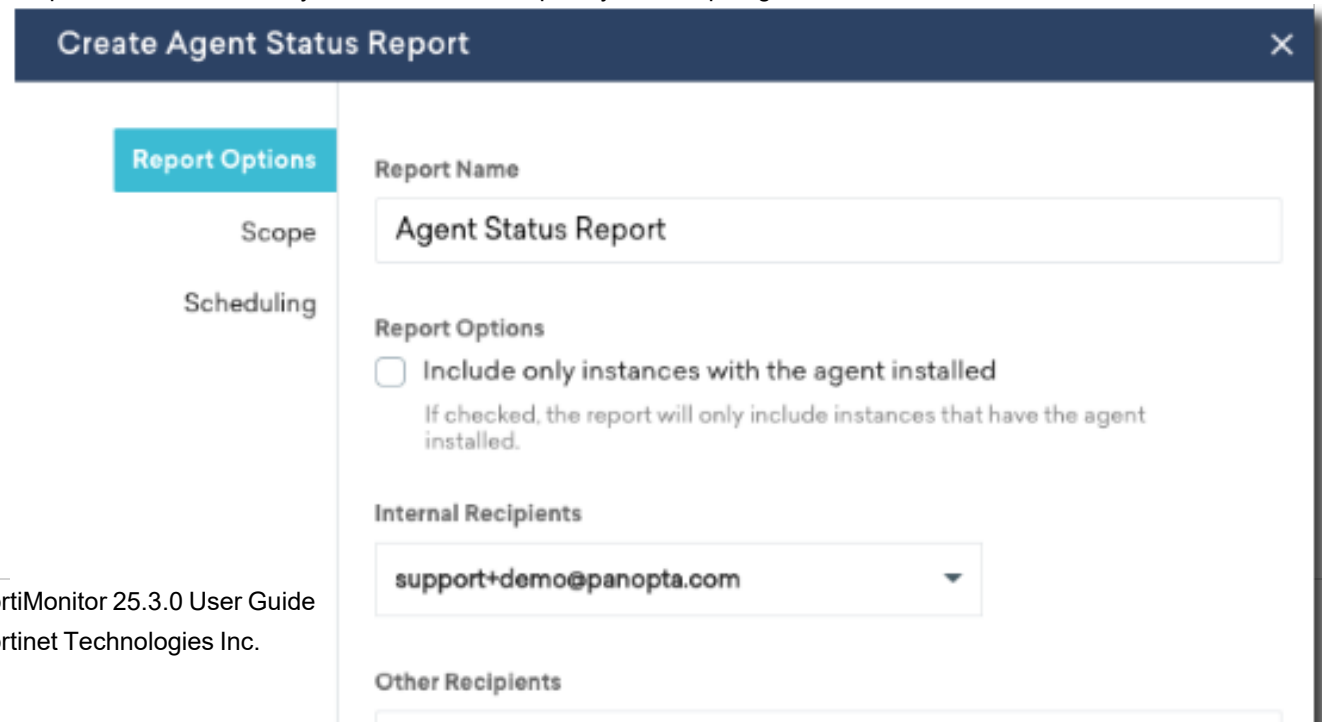
Reports can either be run ad-hoc or on a regular basis to keep you up-to-date on the status of many aspects of your account. Available reports are detailed in the following sections. If there is a particular report you would like, please contact us.

Available reports

- Agent Status Report
- Availability Report
- Capacity Planning Report
- Incident Report
- Incident Metric Export Report
- Interface Utilization Report
- Tag Audit Report
- Tags & Attributes Report
- User Audit Report
- Instance Configuration Report
- Metric Export Report
- Metric Configuration Report
- SD-WAN Report
- Service Performance Report
- MTTR and MTTA Report
- Historical SLA Performance Report
- User Incident Activity Report

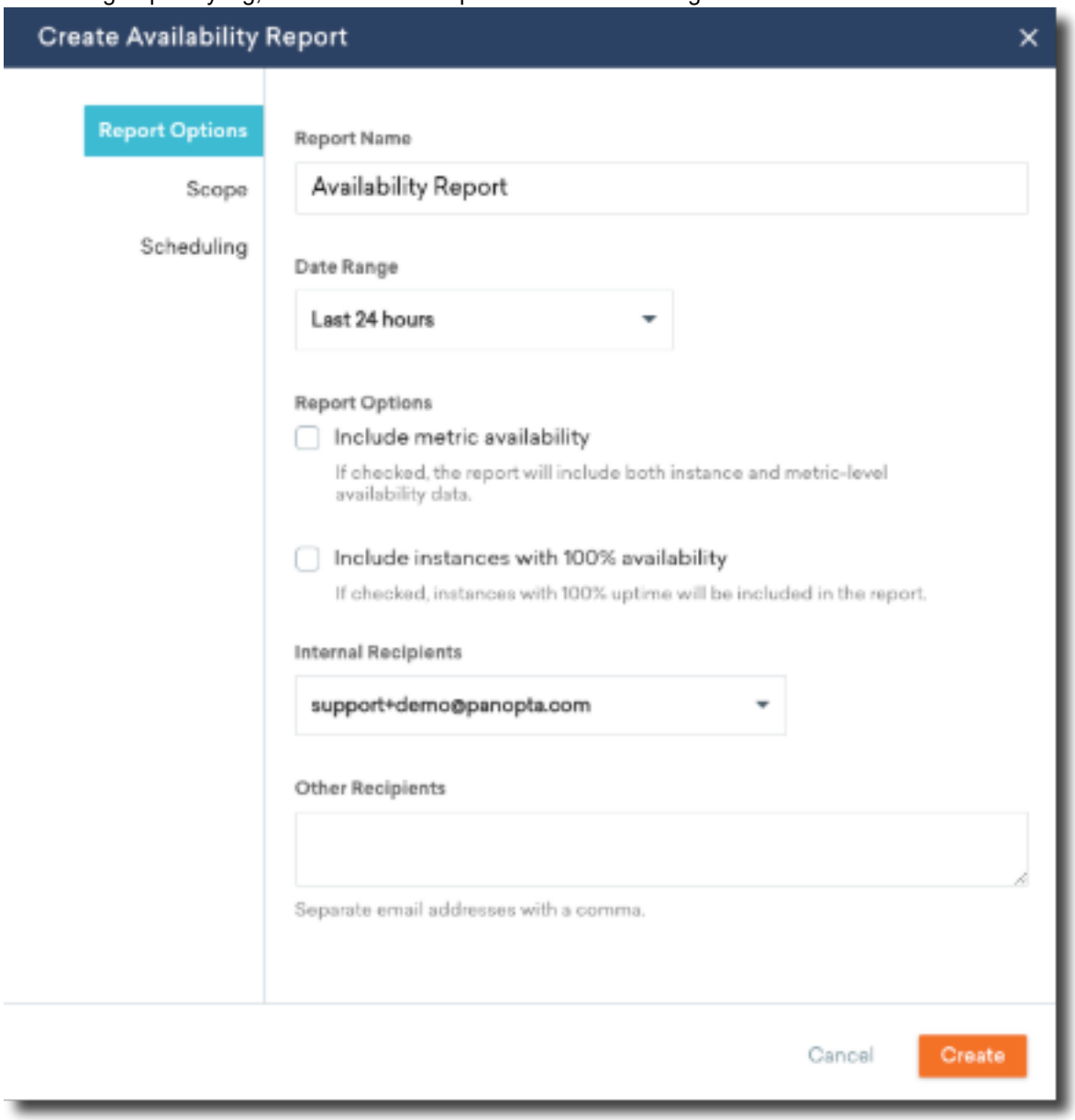
Agent Status Report

The Agent Status report allows you to query your infrastructure for the health and version of your agents. This is helpful for identifying inconsistencies as well as agents that require updates.
Each tab has multiple options, the first allowing you to select the recipients of the report, the second determines the scope, and the last allows you to choose the frequency of the report generation.



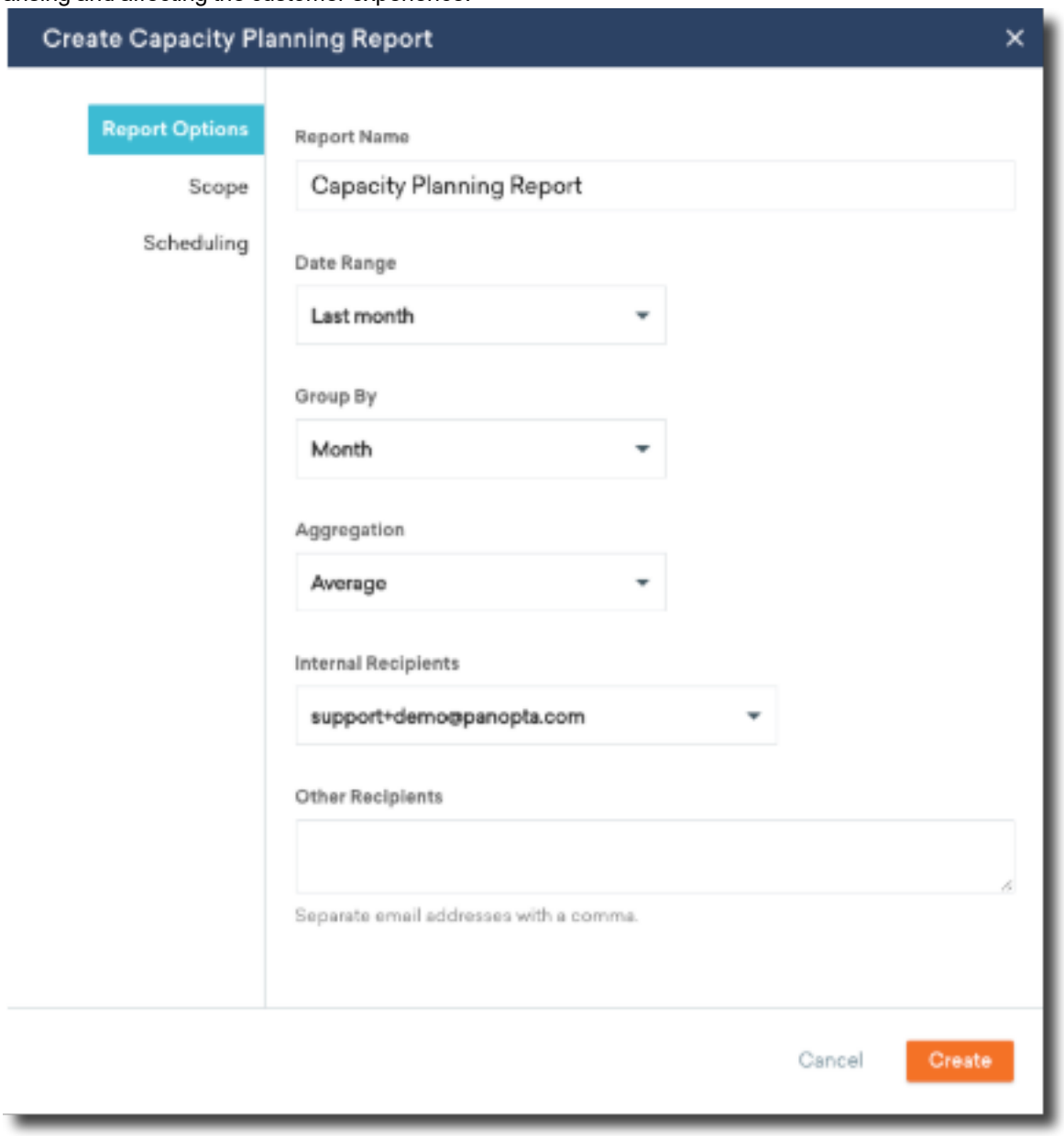
Availability Report

The Availability report gives you a full rundown of the current status of your infrastructure in FortiMonitor, based on a date range. You have additional report options, such as whether or not Server Agent metric availability will be included or if you want to see instances with 100% availability. Similar to the other reports, you have the option to dictate the scope based on group or by tag, and decide if the report should be recurring.



Capacity Planning Report

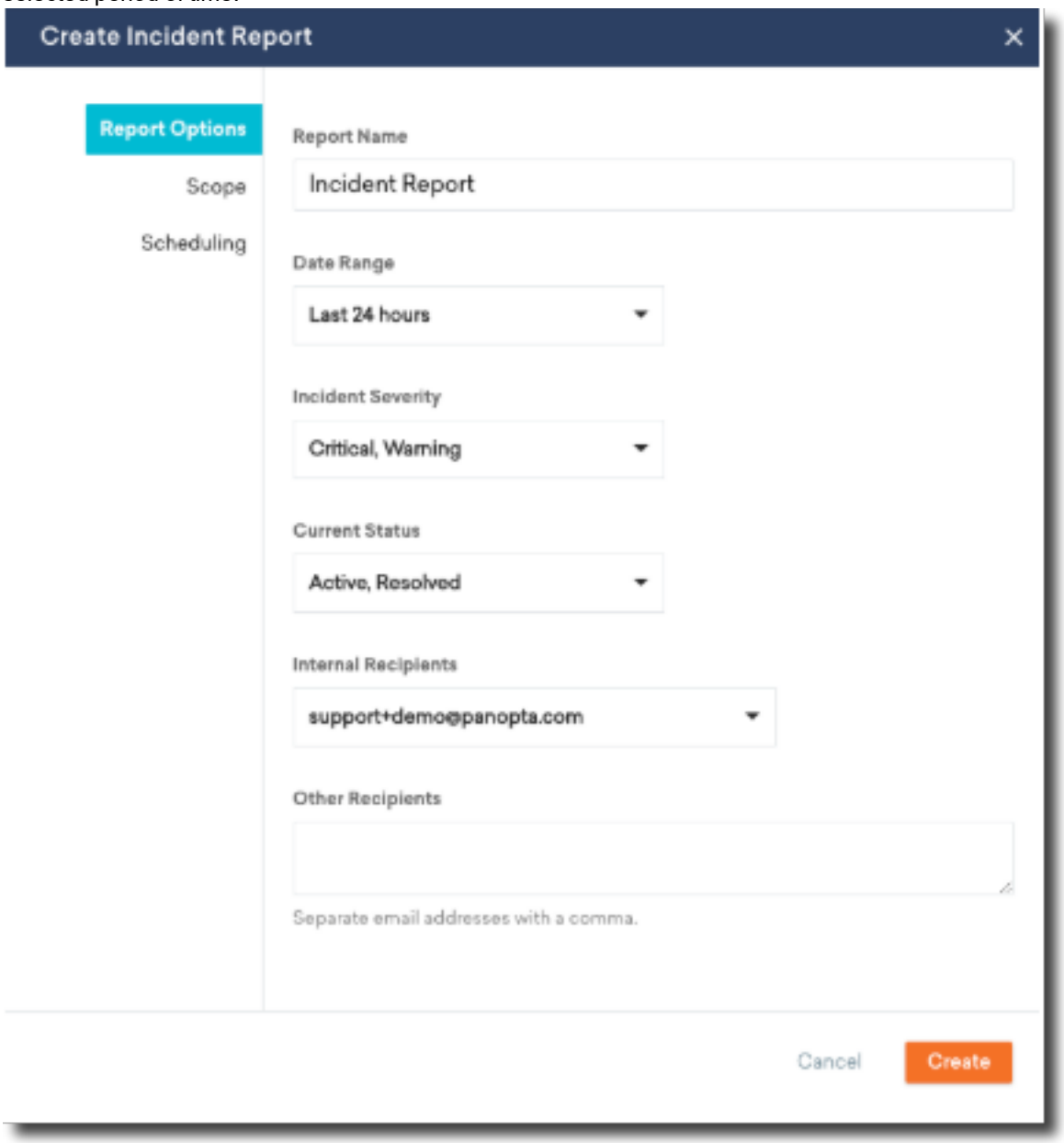
Our Capacity Planning report shows metric trends and usage to help you plan for the future and avoid hardware issues arising and affecting the customer experience.



The available variables are the date range, how to group the report, how you'd like the data aggregated, and who will receive this report. As with our other reports, you can choose the scope and schedule in the next tabs.

Incident Report

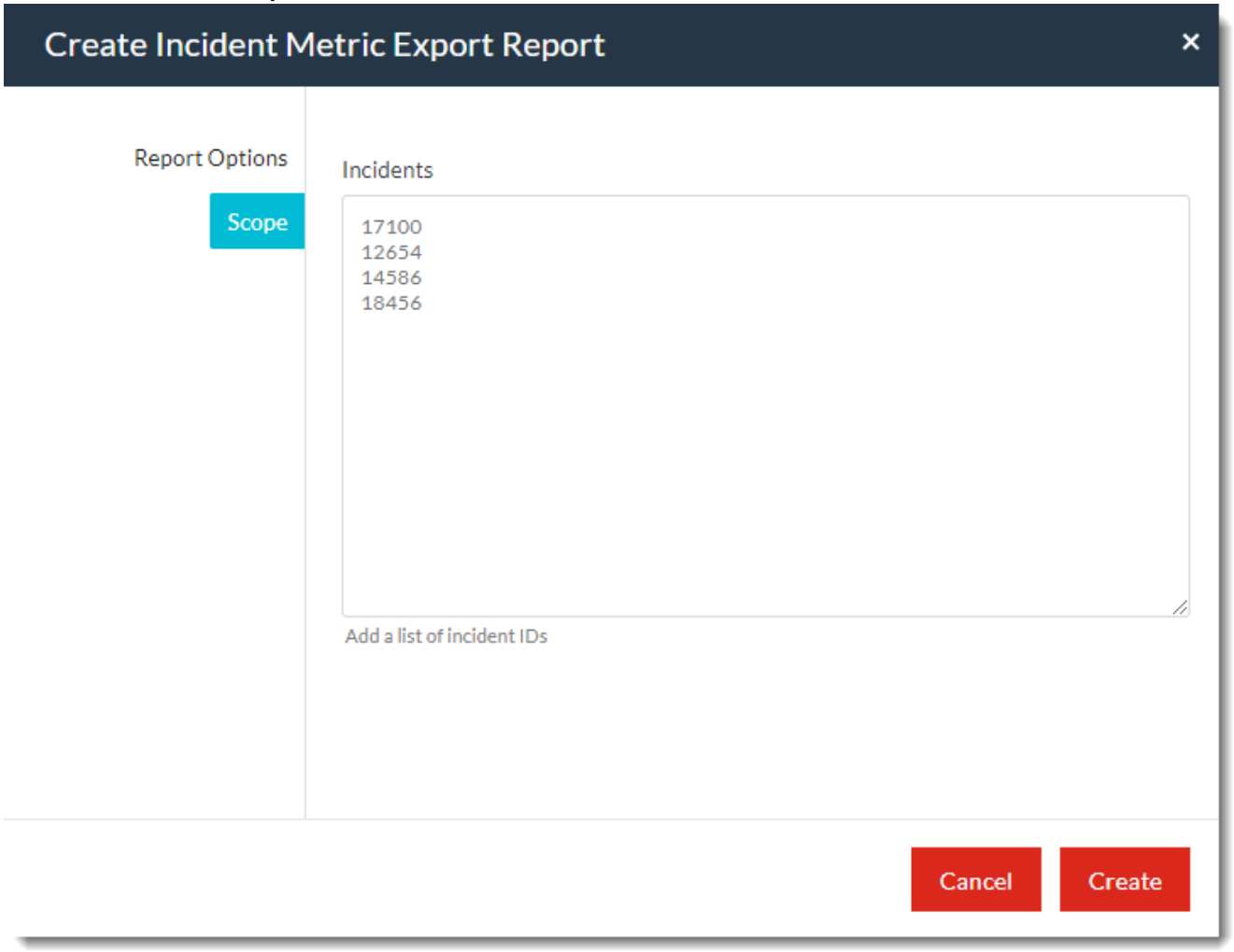
Our Incident report is pretty straightforward, you can see each of the incidents that occurred in your infrastructure over a selected period of time.



You can also decide if you'd like to report based on scope of incidents, severity, and current status.

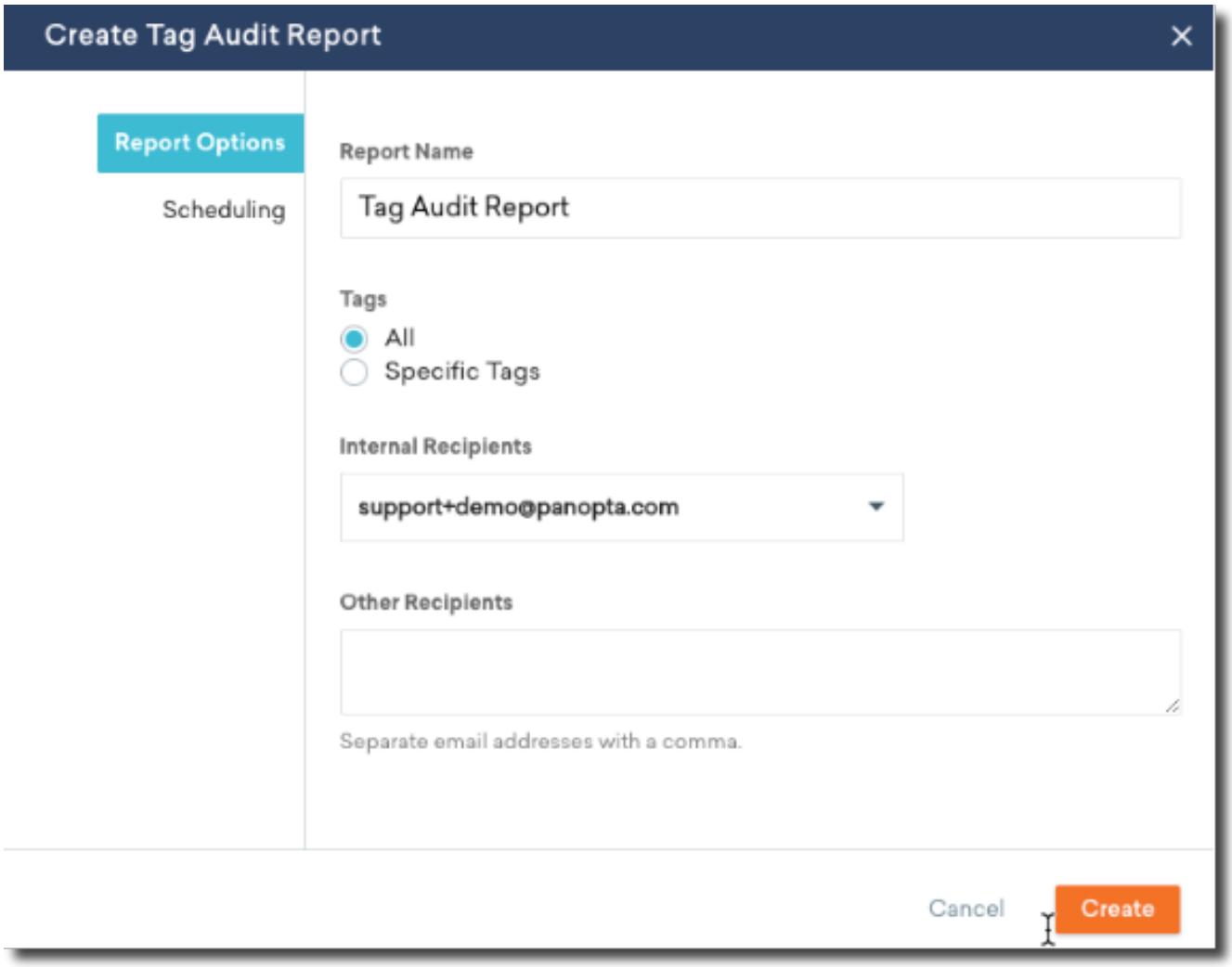
Incident Metric Export Report

The Incident Metric Export report allows you to generate a report for one or more incidents that include detailed metric values for the duration of the incident. Enter the Incident IDs of the incidents that you want to include in the report in the **Incidents** field of the **Scope** tab.



Tag Audit Report

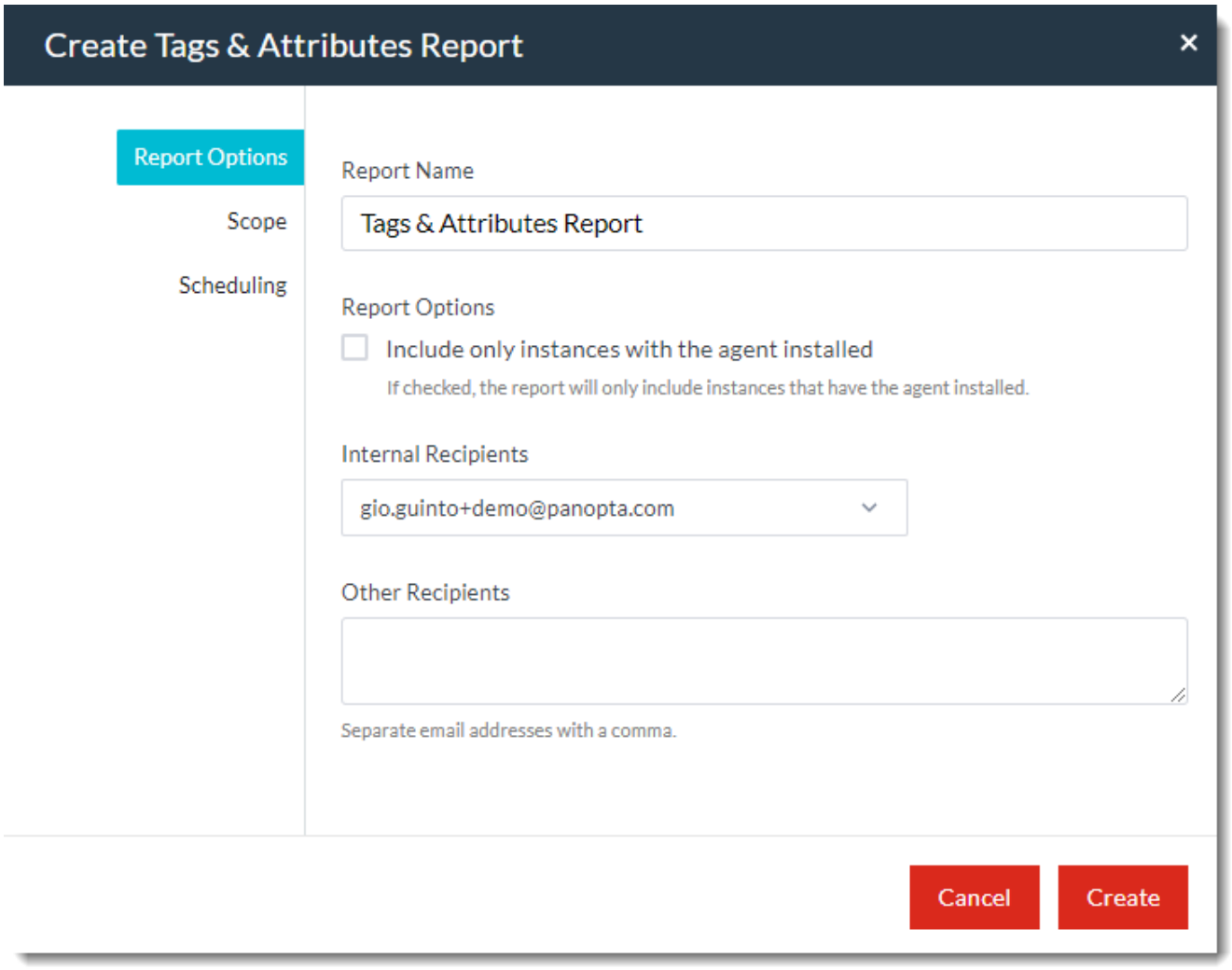
The Tag Audit report allows you to see which tags are in use, and to whom and what they are applied.



You have the option to see all tags, or query by specific tags.

Tags & Attributes Report

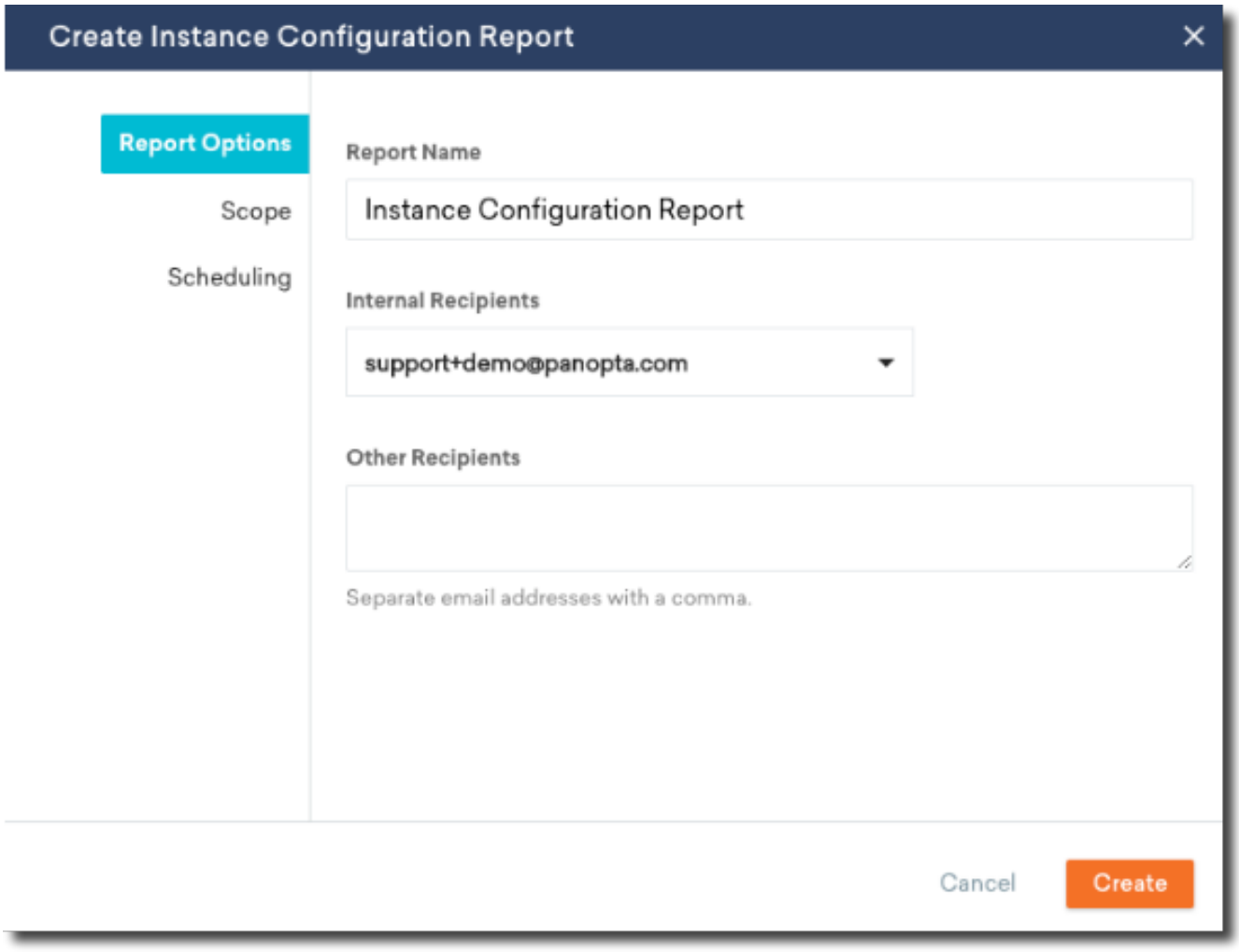
The Tags & Attributes Report shows you a breakdown of all the Tags and Attributes values across a selected scope. You can determine the scope by filtering by Instance, Group, or by Tag, and also schedule the report to run on a one time or recurring basis.



You can determine the scope and filter by Instance, Group, or by Tag, and also schedule the report to run on a one time or recurring basis.

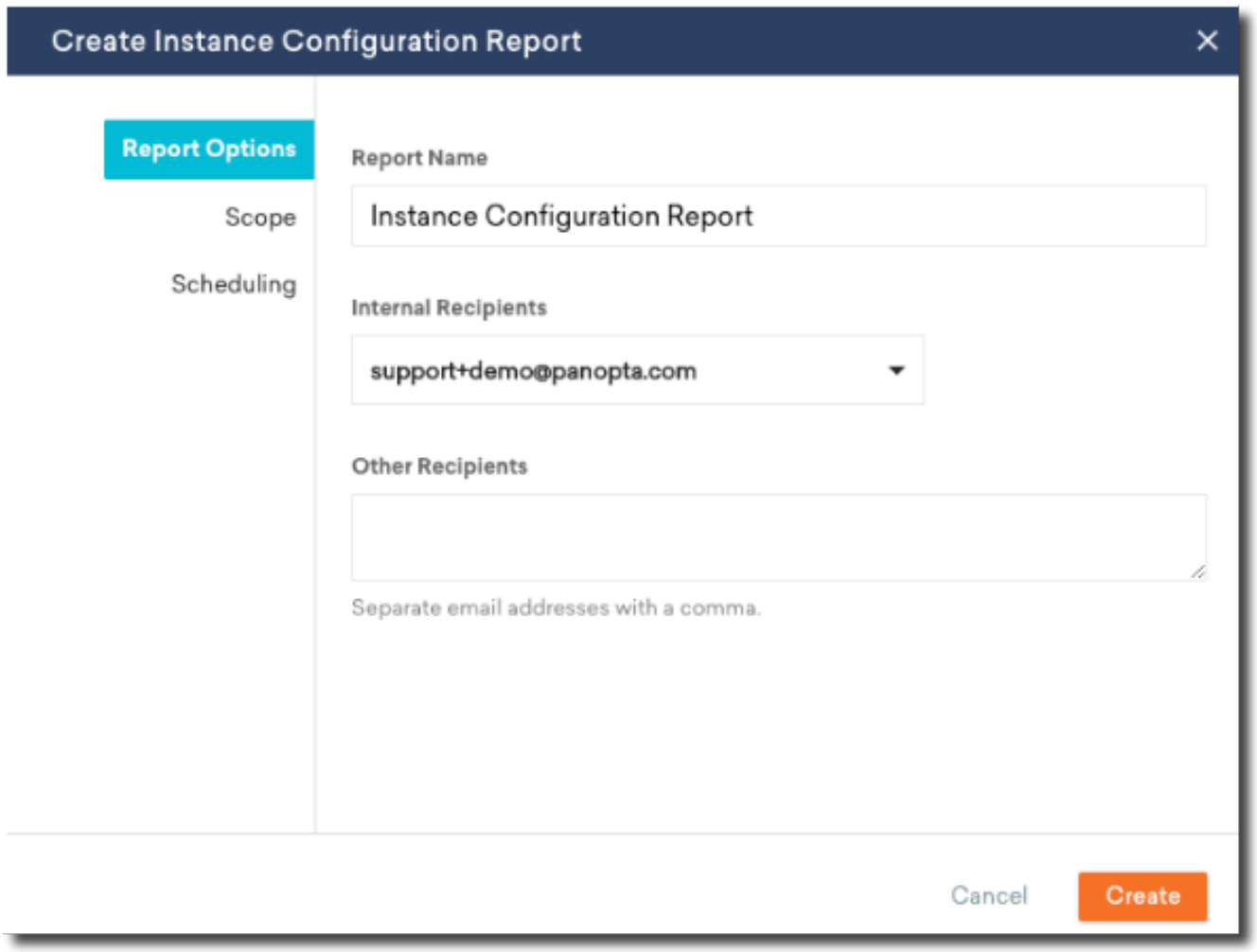
User Audit Report

The User Audit report shows you who is an active user and what permissions they have. You can also see whether or not they have the mobile app installed, and the last time they logged in. The report also includes the **Non-SSO Login** column which shows whether a user can log in via SSO or not.



Instance Configuration Report

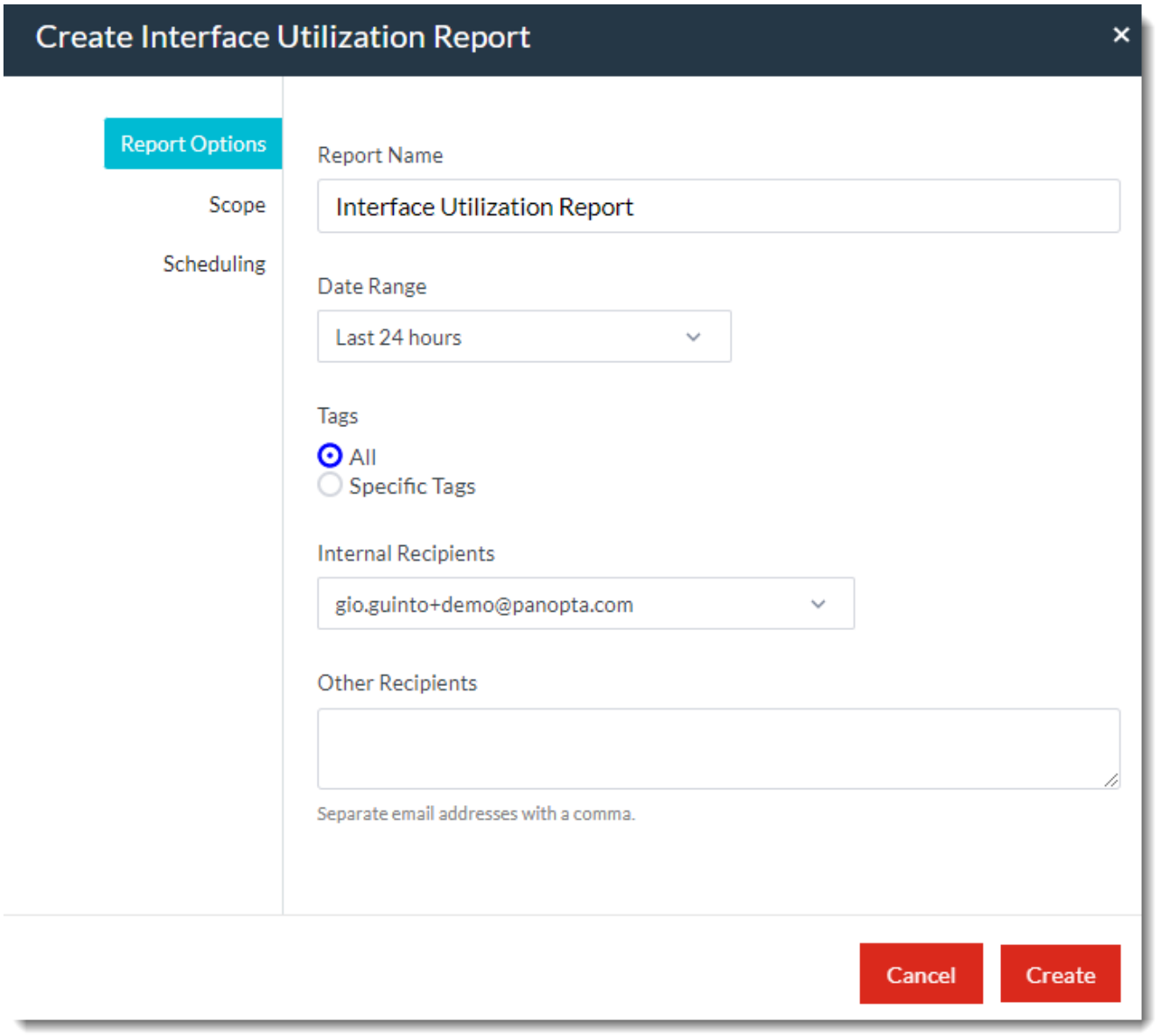
The Instance Configuration Report gives you the ability to export the current monitoring on a determined group of servers.



You can determine the scope and filter by Group or by Tag, and also schedule the report to run on a one time or recurring basis. The report returned gives you an overview of the selected servers, including information like Name, FQDN's, Group, Primary monitoring location, Alert Timeline, Tags, Templates, and any attributes.

Interface Utilization Report

The Interface Utilization Report provides a detailed view of interface metrics such as bandwidth, errors, and packets.



You can determine the scope and filter by Instance, Group, or by Tag, and also schedule the report to run on a one time or recurring basis.

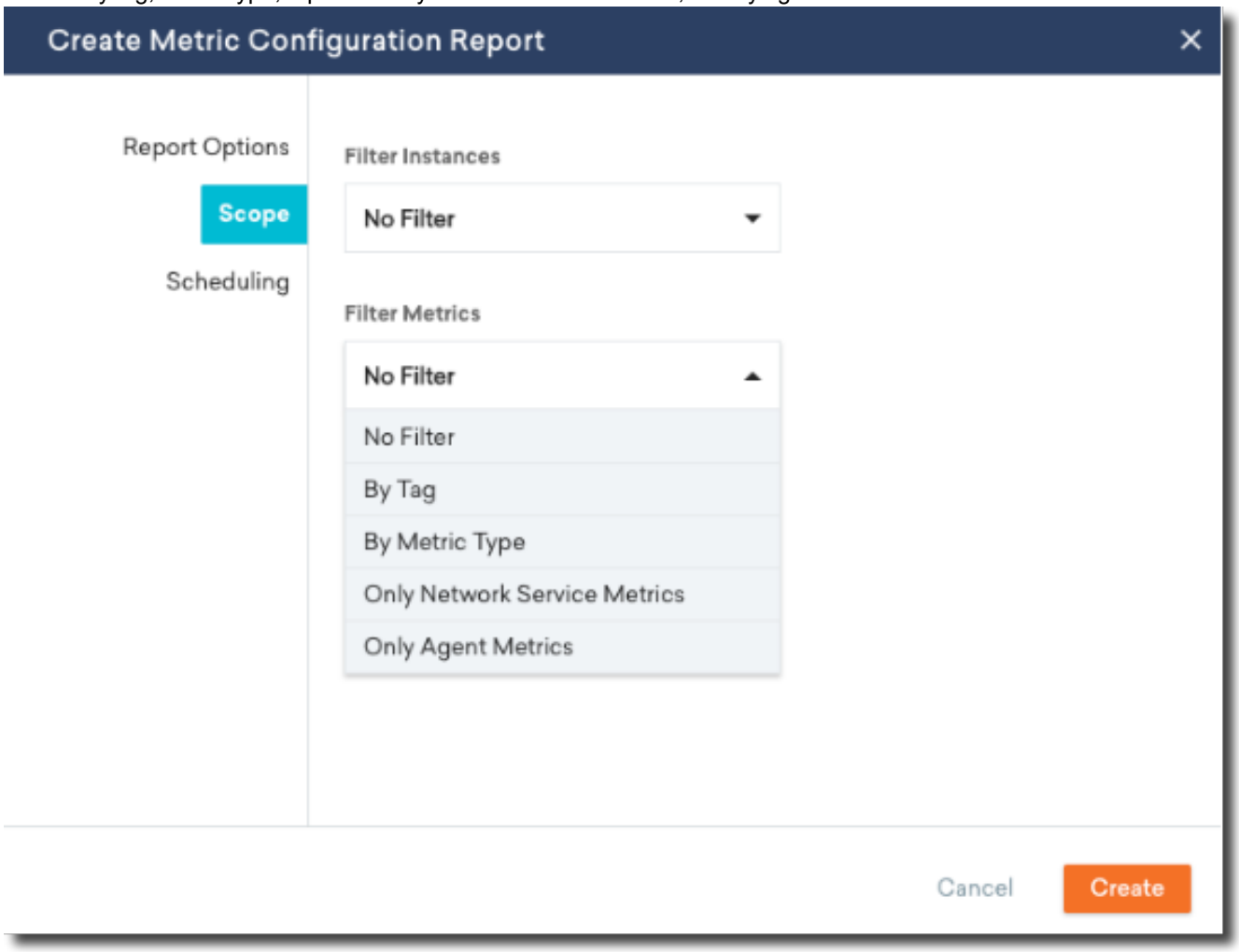
Metric Export Report

This report contains the data details for each metric running in your FortiMonitor account. The report gives you the option to select a date range for the given values and also determine to whom it will be sent. You can limit the scope of the export in the second tab.



Metric Configuration Report

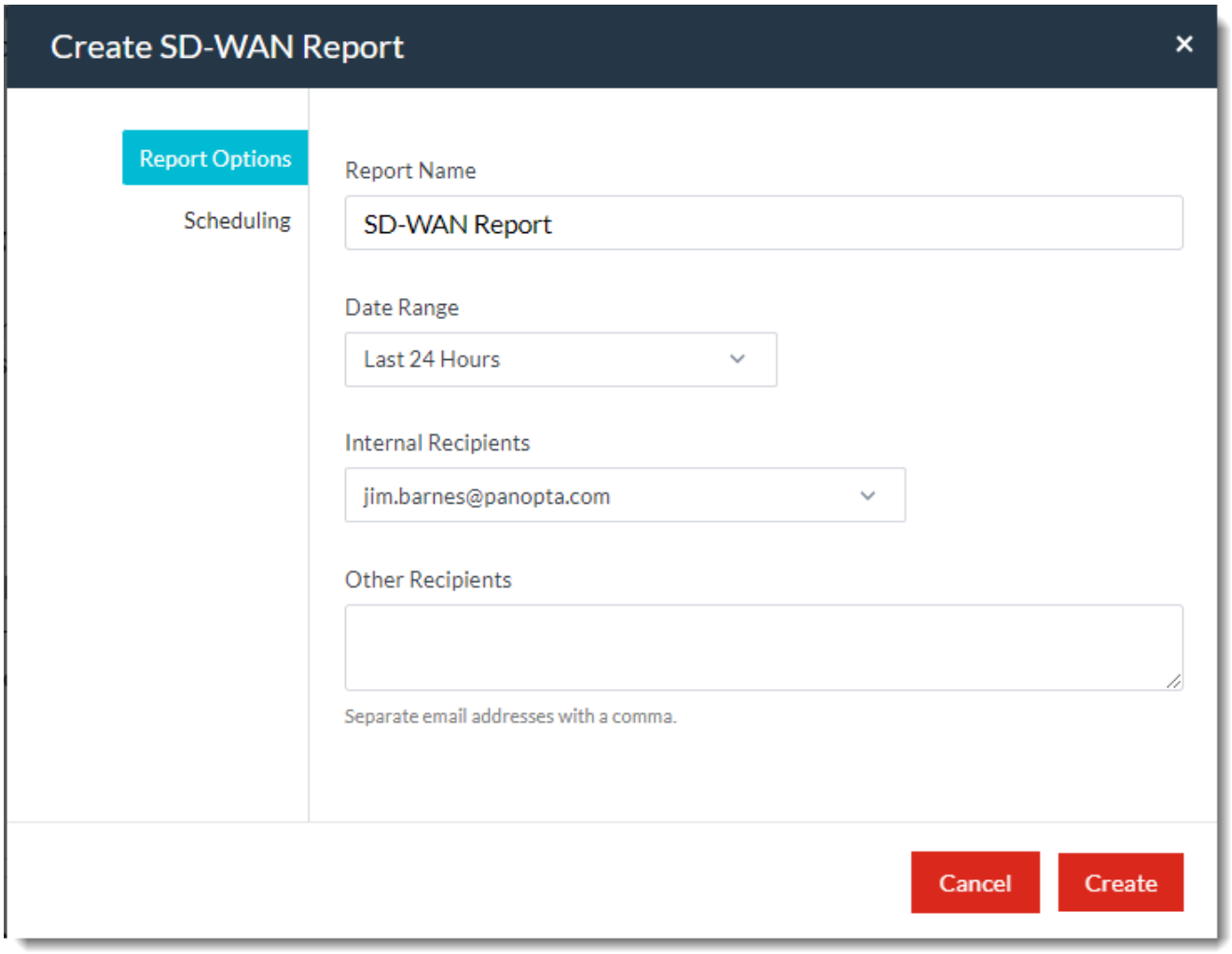
The Metric Configuration report is similar to the Instance configuration report but only reports the metrics. The configuration difference between the two is that under **Scope**, you have an additional option to choose to filter the metrics by tag, metric type, report on only network service metrics, or only agent metrics.



This report is much more detailed than the Instance Configuration report, giving you information like Instance, Metric, assigned Alert timeline, monitoring location of each check, frequency of the check, the current status, applicable tags, and the option to include templates. There are also two separate tabs, one for Network Service Metrics, and the other for Server Metrics.

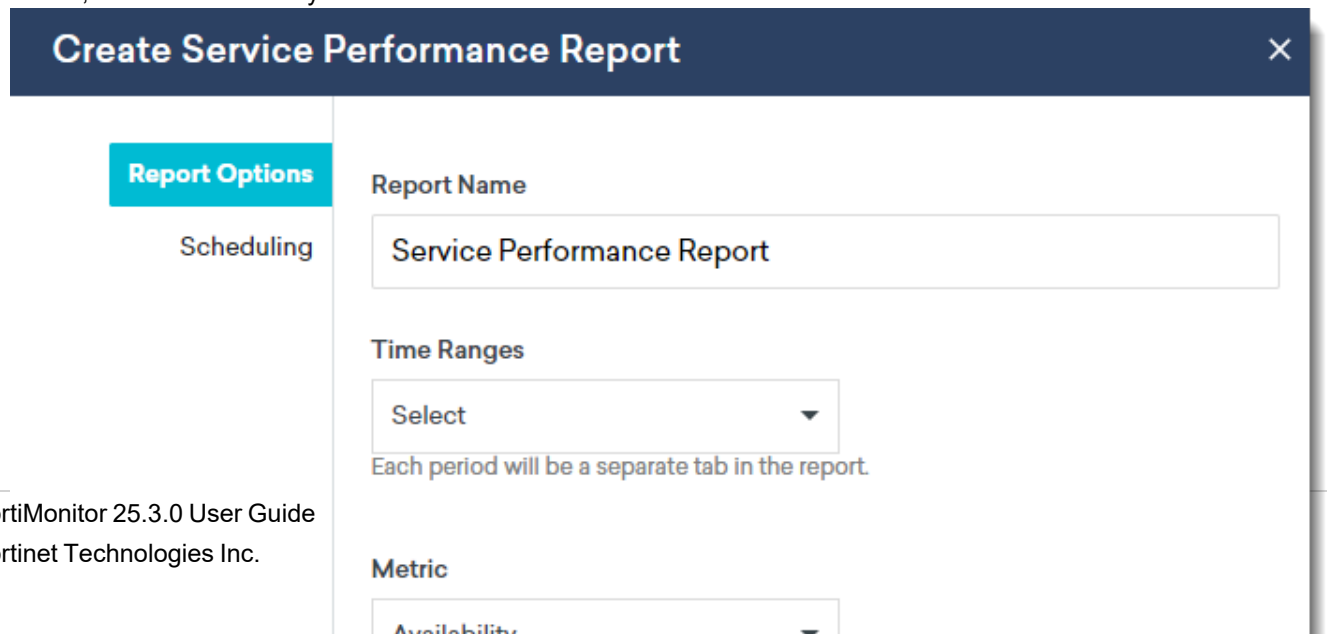
SD-WAN Report

The SD-WAN report provides a detailed view of your SD-WAN metrics.



Service Performance Report

The Service Performance report is similar to the Availability Report, but allows you to view the Availability in a cross-tab format, with metrics on the y-axis and instances on the x-axis.

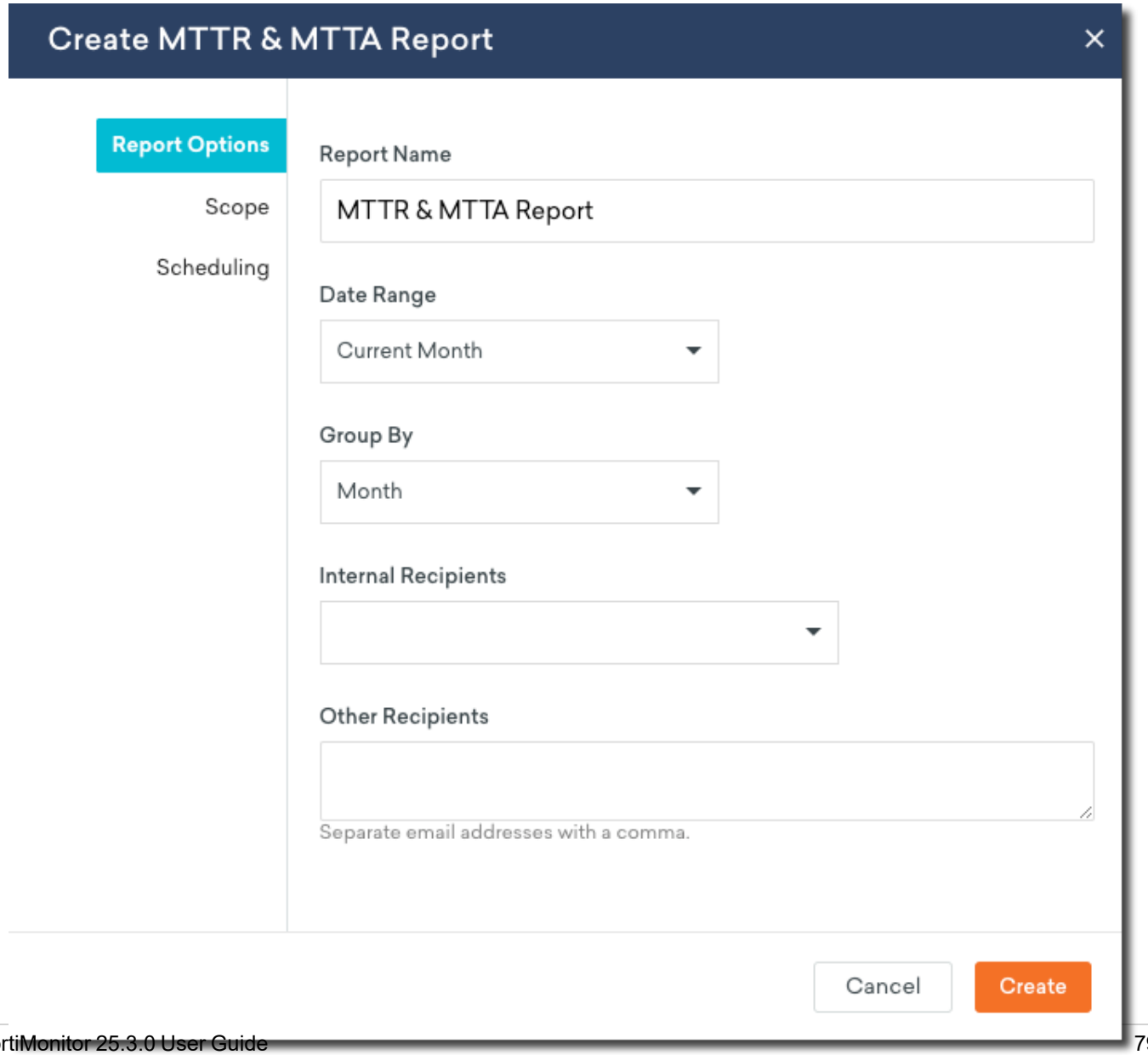


Each row/column is determined by tags, using a basic pattern matching syntax. For example, you tag your instances based on their location: "US-East", "US-West", and "US-North". You can use the pattern "US-*" to generate a column for each. If your metric tag syntax is based on the check location, like "Uptime-Fresno", "Uptime-Chicago", and "Uptime-DC", you can use the "Uptime-*" pattern to generate a row for each. This will result in a 3x3 table, with the average availability of the cross-section in each cell. If more than one instance or metric matches each tag, the average will be used.

You can generate the availability, total downtime, or average downtime.

MTTR and MTTA Report

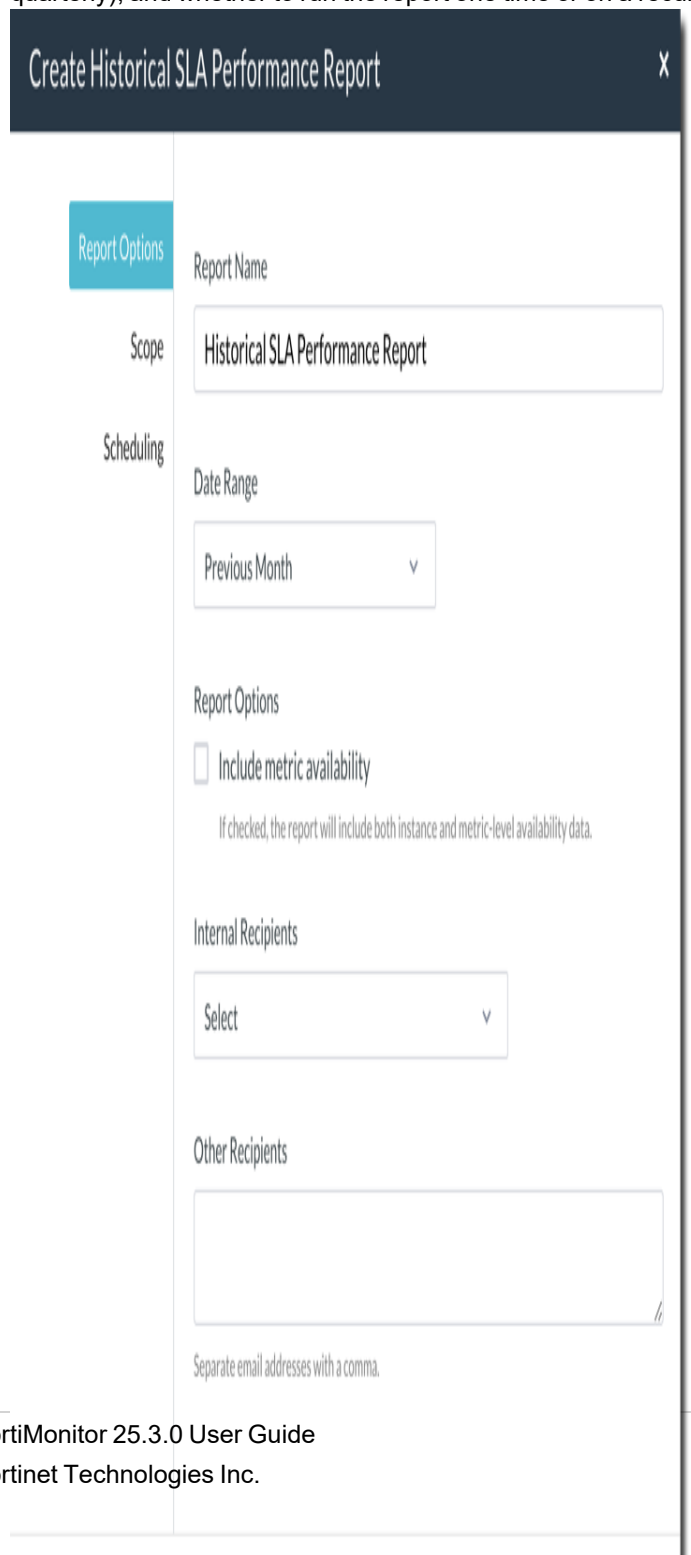
The Mean Time to Resolution (MTTR) and Mean Time to Acknowledge (MTTA) report provides two metrics that measure the average elapsed time from when an incident occurs until the incident is resolved or acknowledged. The MTTA is the average elapsed time incidents are acknowledged after its occurrence. MTTR tracks the average resolution time of incidents.



Similar to other reports, you can also determine the scope and filter by Group or by Tag, and also schedule the report to run on a one time or recurring basis.

Historical SLA Performance Report

The Historical SLA Performance report provides the historical availability (in percent) of instances in your infrastructure. The report gives you the option to select instances, a date range from the given values (previous month, monthly, quarterly), and whether to run the report one time or on a recurring basis.

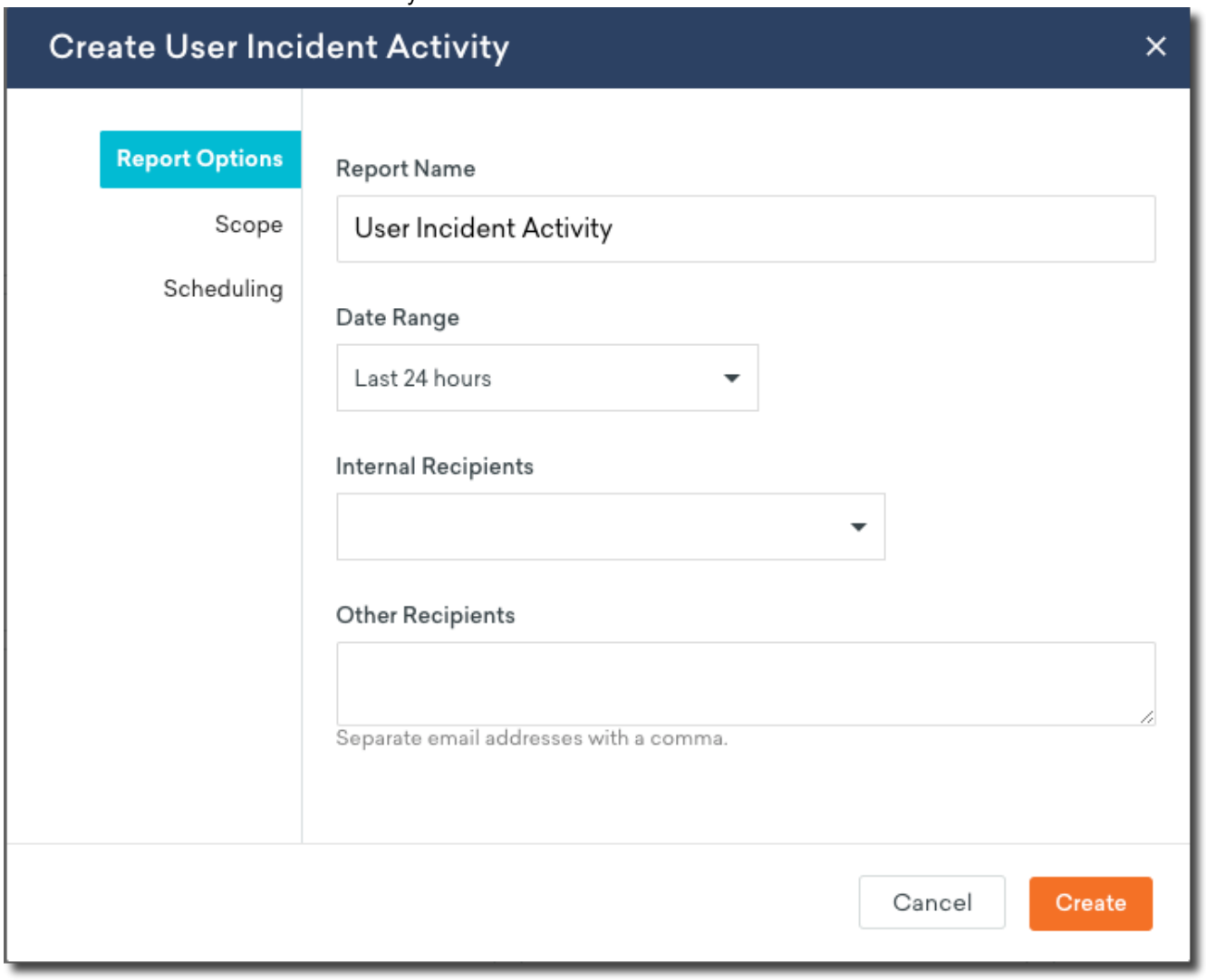


The report returned gives you the monthly or quarterly average availability of each selected instance.

User Incident Activity Report

The User Incident Activity report provides the following information:

- The number of incidents a user is alerted on
- The total alerts received by a user
- The number of incidents acknowledged by a user
- The number of incidents escalated by a user



Similar to other reports, you can also configure the scope, date range, and whether to run the report one time or on a recurring basis.

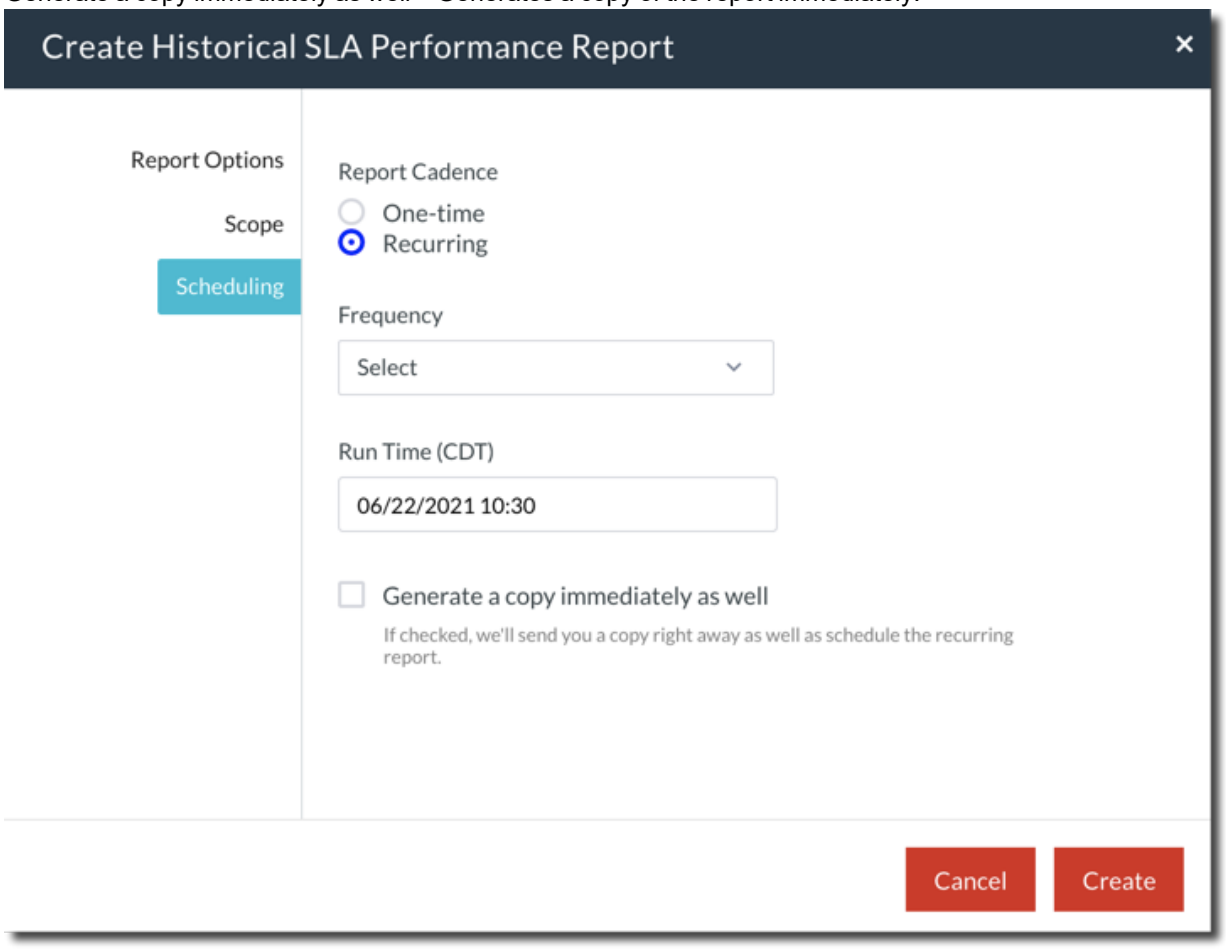
Create a report

To create or schedule a report, perform the following steps:

1. From the navigation menu, select **Reporting > Reports**. Alternatively, you can also use the **Infrastructure and Resource Catalog** by clicking **Add** then selecting **Reports**.
2. Select a report from the **Canned reports** screen, then click **Create**.

   Note: The configuration options vary depending on the type of report you have chosen.
3. After filling in the **Report** and **Scope** options, select **Scheduling**.
   a. To generate a single report, select **One-time** from the **Report Cadence** option.
   b. To schedule a report, select **Recurring** from the **Report Cadence** option and configure the following additional options:
   c. Frequency – How often the report is run (daily, weekly, monthly).
   d. Run Time – Exact date and time when the first report is run.

   Generate a copy immediately as well – Generates a copy of the report immediately.



   e.
4. Select **Generate**. Upon completion, the report will be sent your email address.

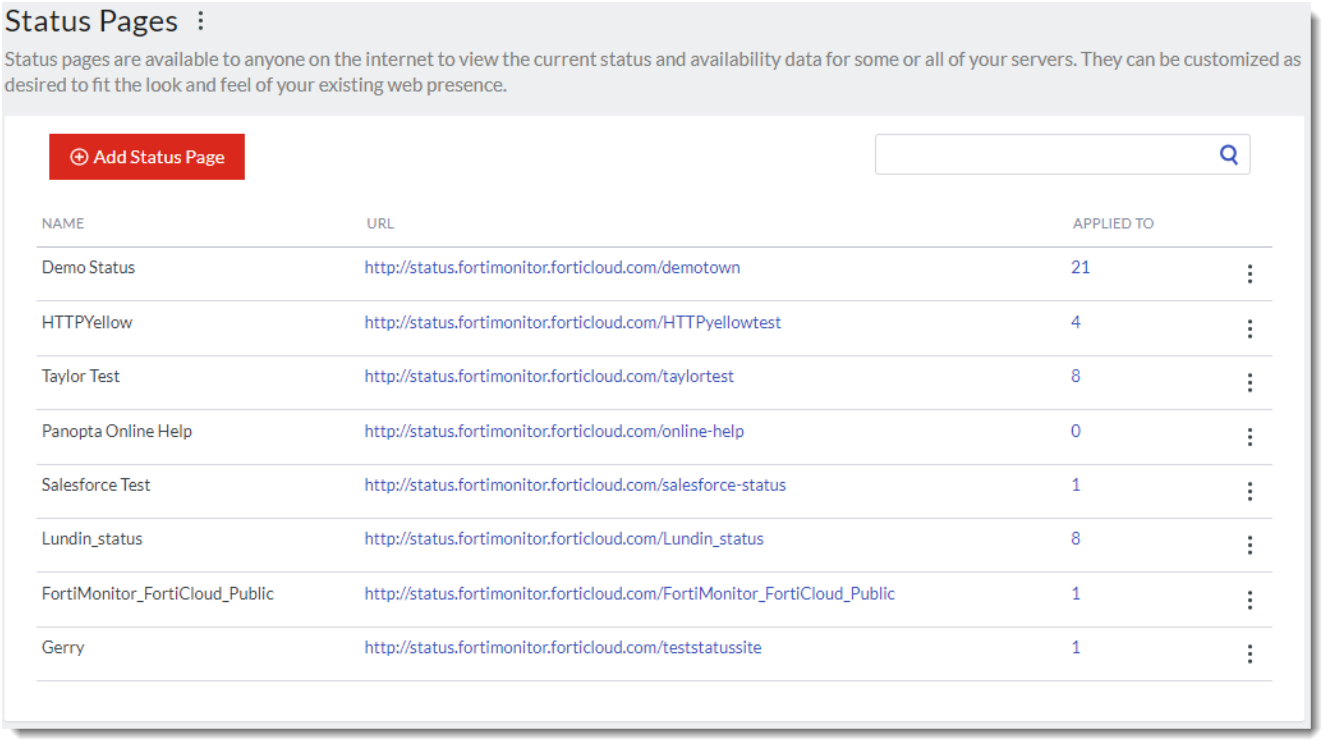
Public status pages

We recently released new customizable status reports. If you would like to try these out, then please let our support team know and we will enable this feature for your account.

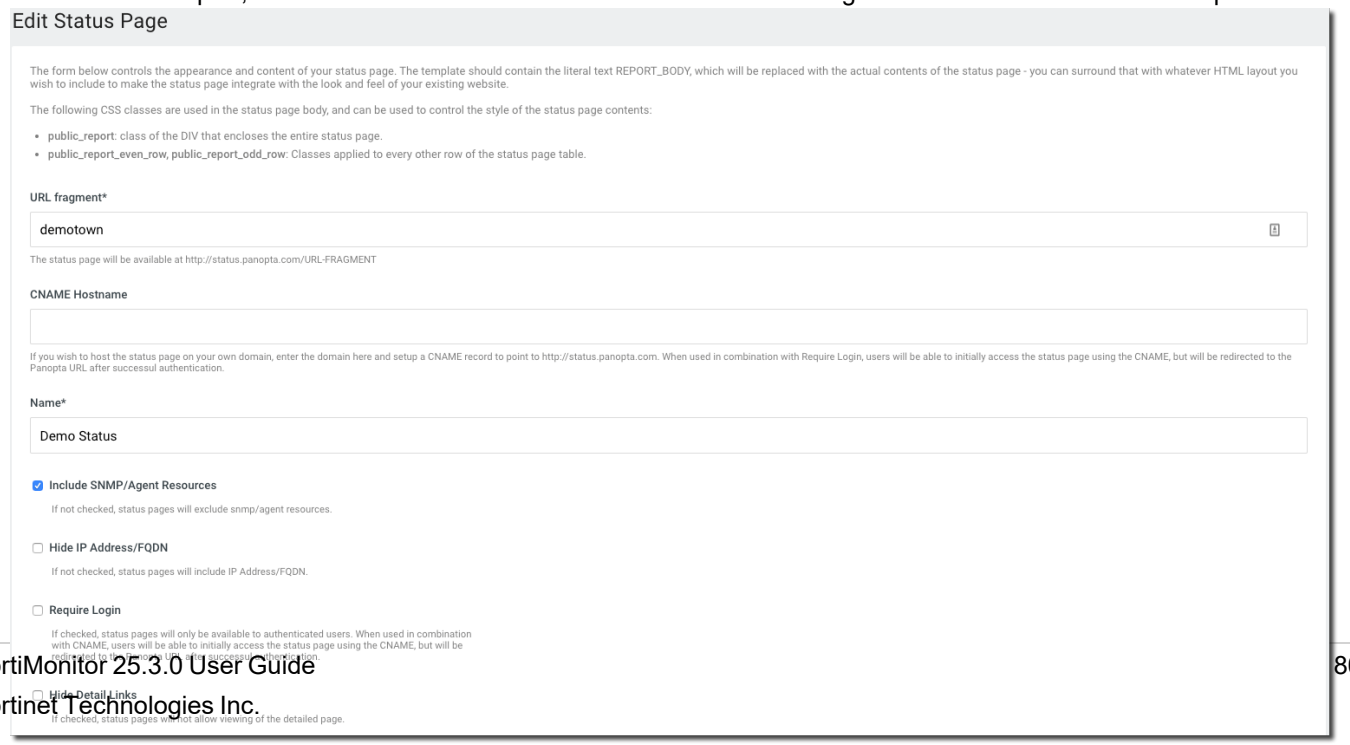
Our status reports were designed for just this purpose - they let you publish a publicly viewable page that lists the current status and availability for the servers FortiMonitor is monitoring. You can setup any number of status reports you desire, segmenting your servers into separate reports if desired.

To create a new status, go to the **Reporting** > **Status Pages**.

On this new page, click the **Add Status Report** button.



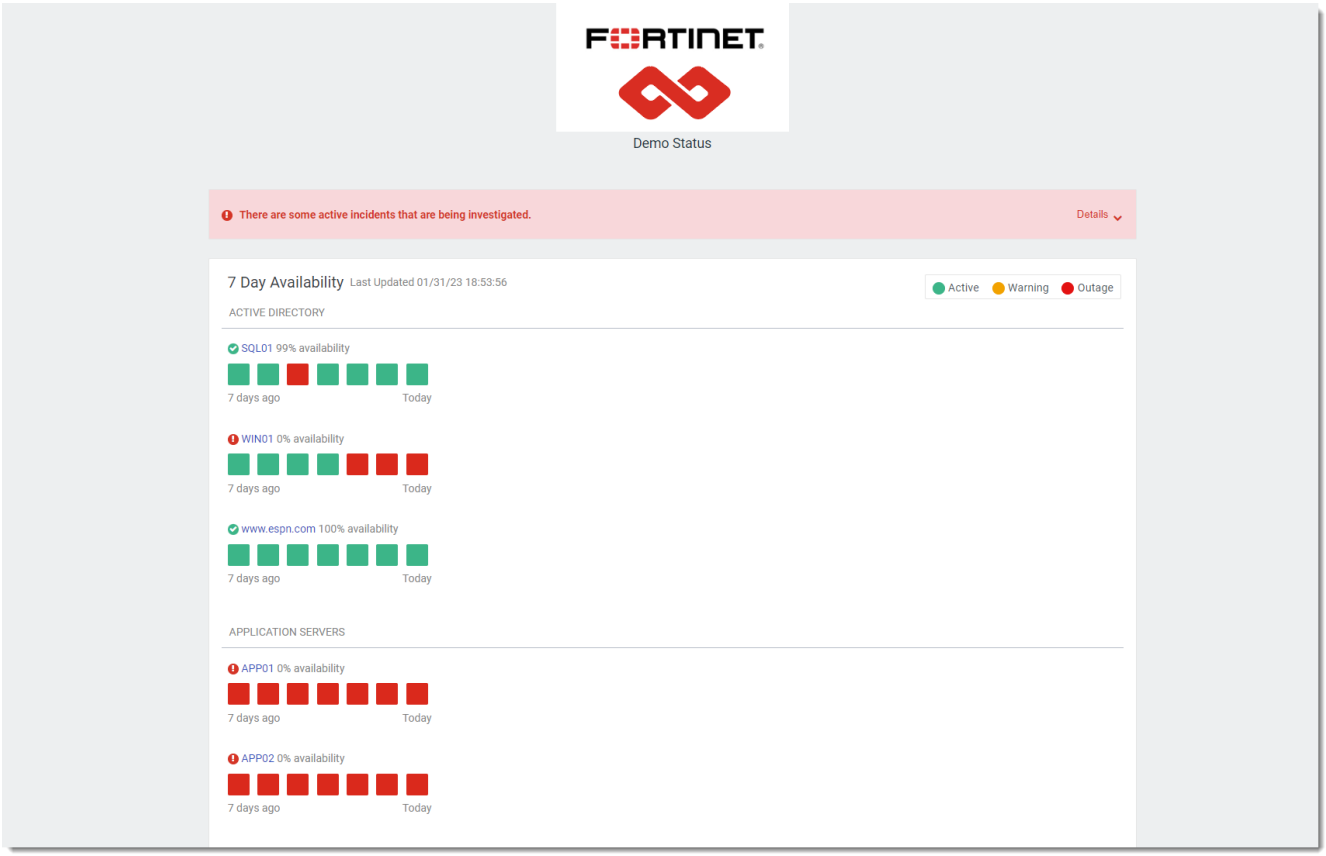
This brings you to a page where you can configure the report, specifying the URL that will be used for the report, the duration of the report, what servers are to be included as well as other settings that control the details of the report.



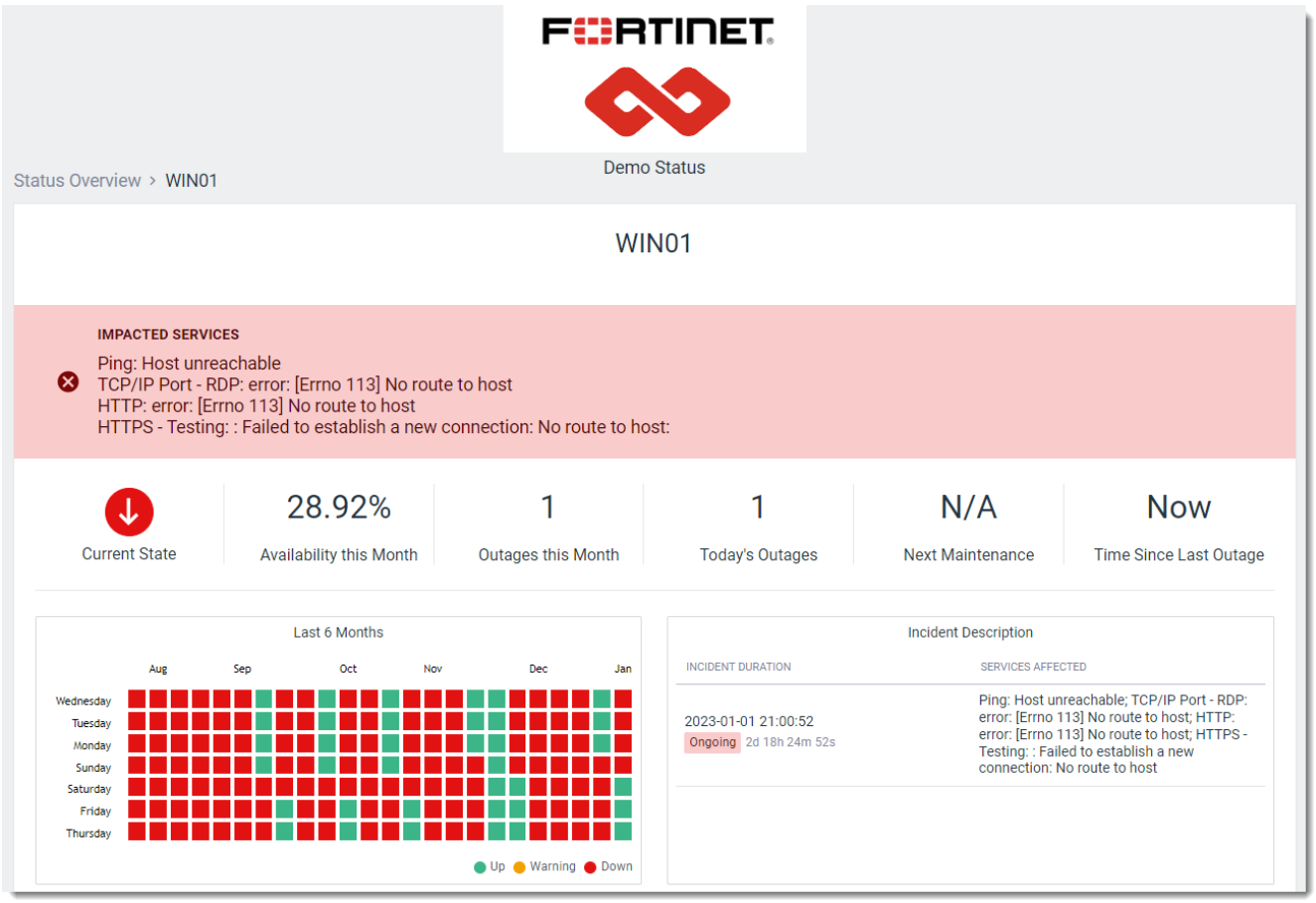
Note: If the **Require Login** option is enabled and the user is logged in to FortiMonitor, The status page will not prompt for authentication.

You can choose how you would like to sort the report itself, as well as choose which servers or server groups to include in the report. You can also specify a template to use for the body of the report.

Select **Create Report** when you are done. You should now see your new status report listed on the status reports page complete with URL. If you open this URL you should see something like the following:



You can hover over any dot for more information on that day. ***The arrows on the left state the server's current status, a green arrow meaning up and red one meaning down. To the right is we show the performance of that server over the past week.*** A green dot indicates that your server was up all of that day, a yellow dot indicates that there was at least one alert and a red dot indicates that there was an outage. You can also click on any one of the servers, which will take you to a more detailed report for that server. You can see an example of this below:



By clicking on any day in the calendar, you can see a detailed outage log open to the right. This page will also include graphs of all of the network service and Agent checks running on this server, as well as availability percentages and any upcoming maintenance periods.

Customizing Status Pages

Status pages were designed to support flexible customization so that they can be integrated into the look and feel of your website.

The overall look of the page is controlled by a HTML template that can be edited within the our control panel. This template needs to contain a text string "REPORT_BODY", which is dynamically replaced with the HTML for the report body when the report is generated. The HTML can be any valid HTML, from a basic wrapper with your logo at the top to a fully-customized page with complex navigation and layout.

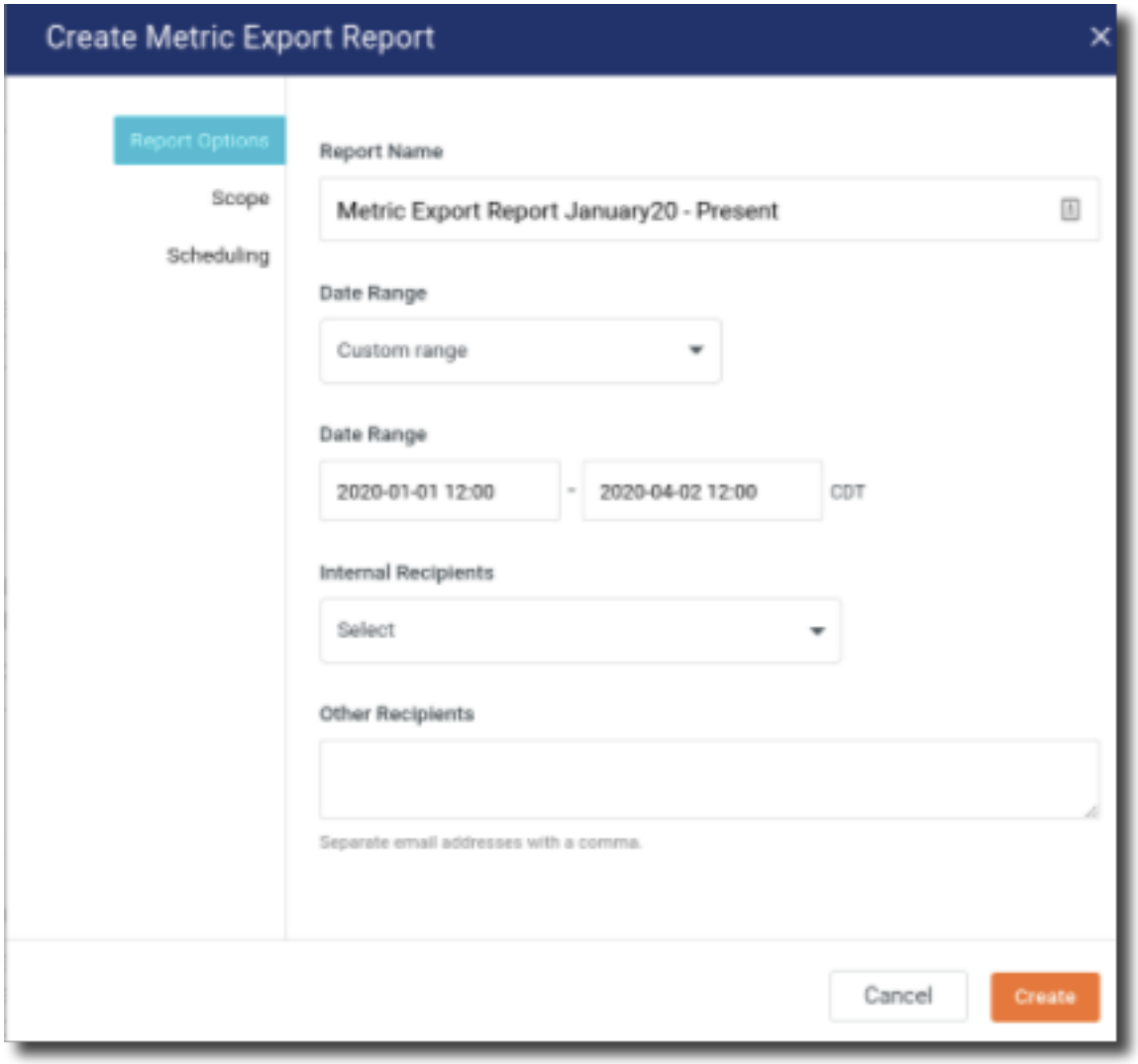
The actual report body HTML is setup to support styling via CSS. A few important classes to be aware of are public report, which is the outermost DIV that contains the entire report. Inside this DIV, each of the headers and table elements are tagged with classes to support custom styling.

Exporting Metric Reports for long duration time periods

In some cases, the time range will be too large to export the data. To get around this, you can use 6-month sequential batches to export these metrics

Here's an example of how to do this:

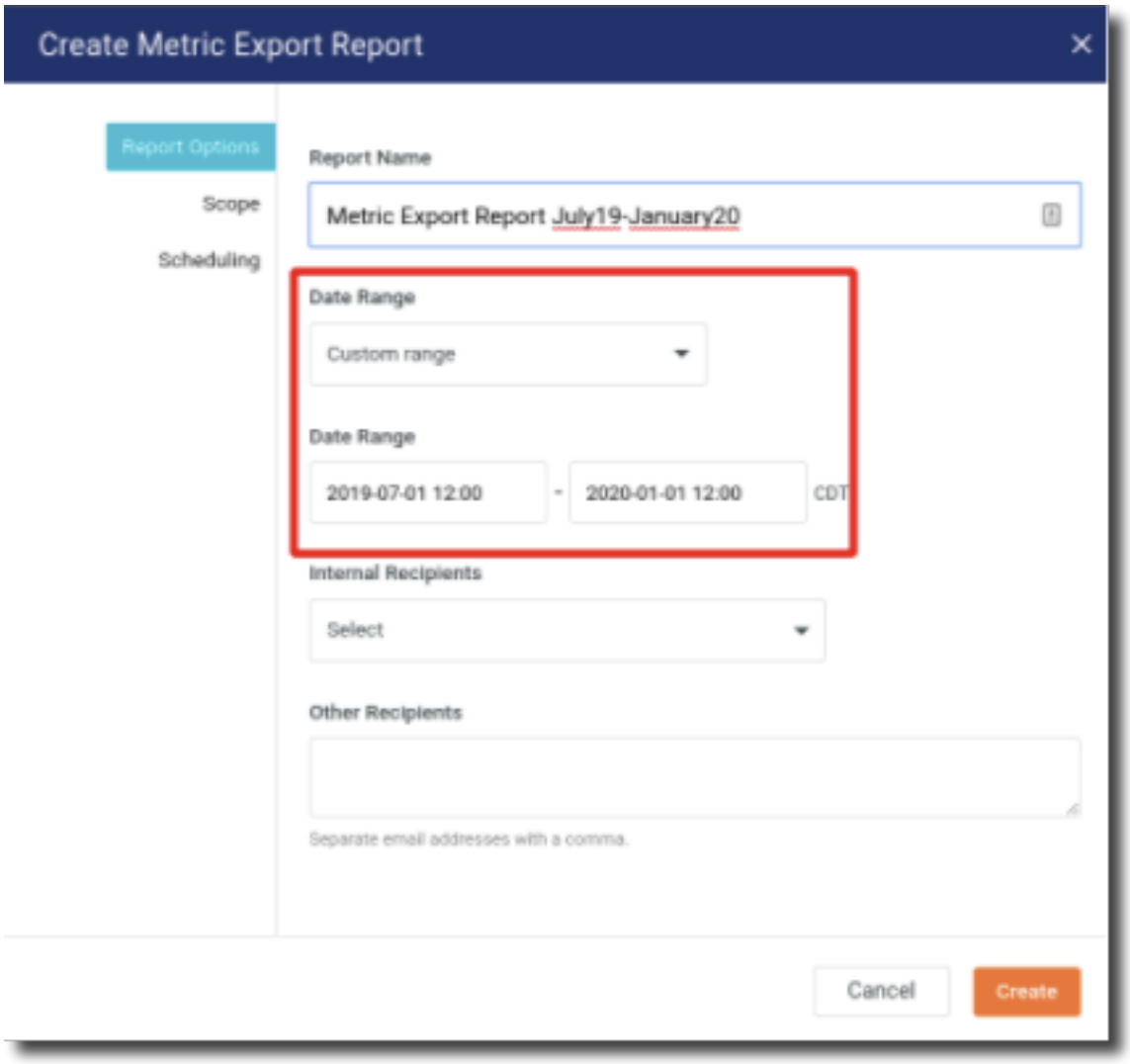
- Select the custom range: January 1st, 2020 - April 2nd, 2020.
- Select the recipients.
- Select the instance in the scope tab, one at a time, for example, Device 1.
- Click **Create**.



You should get the email with the report attached. Also, you can download it from the **Report History** under the name **Metric Export Report** (it can take time until the **Download** link shows up).

For the next range:

- Select the custom range: July 1st, 2019 - January 1st, 2020.
- Select the recipients.
- Select the instance in the scope tab, one at a time, for example, **Device 1**.
- Click **Create**.



You can repeat these steps for the rest of the time windows:

January 1st, 2019 -July 1st, 2019

July 1st, 2018 - January 1st, 2019

Maintenance Schedules

From time to time, you will need to perform routine tasks on your infrastructure that will render it unreachable or cause it to operate outside of the norm - upgrades, maintenance, etc. For that reason, we provide Maintenance Schedules - finite periods where you can instruct FortiMonitor's monitoring of your infrastructure to behave and react in special ways.

Creating custom Maintenance Schedules

Maintenance Schedules page

Each of the options below will open the same **Create Maintenance Period** drawer.

From the Maintenance Schedules page

1. From the navigation menu, select **Maintenance**, then select **Schedule & History**.
2. In the top left corner, select **Add Maintenance**.

From the Instance Details page

1. Navigate to any instance in your account.
2. Select **Maintenance**.
3. Select **Add Maintenance**. The Immediate Maintenance options are discussed below.

From an instance group

1. Navigate to an instance group in your account.
2. Towards the top right, there's a block for **Maintenance**. Click the **Add Maintenance** button.
3. Select **Custom**. The Immediate Maintenance options are discussed below.

Configure a Maintenance period

When configuring Maintenance, you can control the general handling of the maintenance period, the timing, and the scope of your infrastructure which it covers.

General configuration options

Field/Option	Description
Maintenance Period Name	The **Maintenance Period name** is displayed throughout the FortiMonitor control panel and can be anything of your choosing
Start Time	The **Start Time** is the time at which the Maintenance Period should commence. If the Maintenance Period will be recurring (see Timing), it will always start at this specific time of day.
Timing	Length of time, in minutes, that the Maintenance Period will last. • Duration (mins) – If the Maintenance Period is recurring, it will last this long each time. • Ending at – The Maintenance Period will end at this date and time.

Description	A helpful way to communicate what the Maintenance Period covers and why it's being used.
Monitoring	The options under Monitoring dictate how FortiMonitor monitors the specific infrastructure covered by the Maintenance. • Continue monitoring but don't create incidents –All monitoring for the instances covered by the Maintenance Period will continue - however, incidents will not be created, regardless of the monitoring data collected. • Continue monitoring & creating incidents but don't send alerts – All monitoring for the instances covered by the Maintenance Period will continue and incidents will be created accordingly - however, no alerts will be sent for any detected incidents. • Continue monitoring, creating incidents, and sending alerts – Things will operate as normal - however, an Alert Timeline override must be selected to be used for alerting.
Options	• Show maintenance indicator in public reports – If checked, we'll show a Maintenance badge in your FortiMonitor Status pages if you're using them. • End applicable outages when the period begins – If checked, any active incidents that are present when maintenance begins will be force-closed. • Apply to sub-tenants – (Multi-tenant users only) If checked, the maintenance will also be applied to sub-tenants. Enabling the option will also restrict you to applying maintenance by tag.

Timing configuration options

By default, Maintenance is a one-time event. However, if you have recurring maintenance, you can use the recurring event option. Select the recurring option will present a set of scheduling controls similar to what you'd see in Gmail or Outlook.

Option	Description
Every n days	Beginning on the day of your Maintenance start date, Maintenance will occur every n days from that point.
Every n weeks	Beginning on the day of your Maintenance start date, Maintenance will occur on the selected days of the week, every n number of weeks. For example, use 1 for every week, 2 for every other week.
Every n months on the nth day of the month	Every n number of months, on the date of the month that coincides with the date of your Start Date. For example, entering a start date of May 1st every 1 month will schedule Maintenance on the first day of every month.
Every n months on the nth day of the week of the month	Every month, Maintenance will be created on the n day of the nth week. For example, setting a start date of May 1, 2019, which is the first Tuesday of May, will schedule Maintenance for the first Tuesday of every month.

Recurrence end date	The recurring maintenance will end on this date.
Adjust for daylight savings	Select a timezone if you want maintenance schedules to adapt to the timezone's daylight savings schedule. This option is used to adjust maintenance schedule start times based on the beginning and end of daylight savings time in the specified timezone.

Scope configuration options

Option	Description
Impacted Instances	• By Name – Utilizing the infrastructure tree, you can select the specific instances which will be part of the Maintenance Period. • By Group – Utilizing the infrastructure tree, you can select the group(s) which will be part of the Maintenance Period. When the Maintenance Period begins, we'll check each selected group for contained instances; instances present at that time will be placed under maintenance • By Tag – Using the tag selector, enter the instance tags you'd like to select instances by. By default, all the tags you enter must be present on the instance at the time of maintenance creation. However, if you switch it to Match Any, only one of the tags must be present on the instance. If you edit an instance and add a tag that is being used as a maintenance selector for an active Maintenance Period, we'll put it under maintenance accordingly. Additionally, when adding an instance that matches the tags used in an active maintenance, the instance is automatically put under maintenance as well. • By Metrics – You can select the metrics which will be part of the Maintenance Period. Any metrics that you select will be put under maintenance. Monitoring will continue as usual for metrics that were not selected.
Impacted metrics	• All – All metrics will be included in this maintenance schedule. • Selected – Only metrics that you add will be included in this maintenance schedule.

Immediate Maintenance

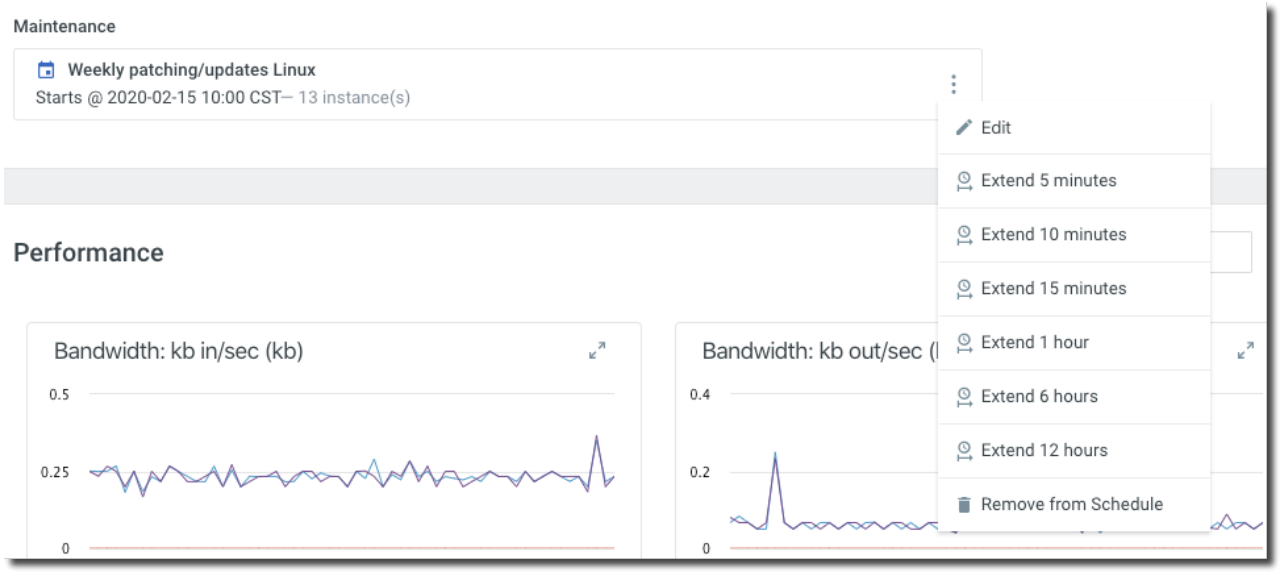
If you do not need to customize your Maintenance Period, you can invoke immediate maintenance from any instance or group.

1. On the right side of the page, locate the **Maintenance** block.
2. Click **Add Maintenance**.
3. Select your desired duration.

Maintenance will begin immediately for your instance or group. Monitoring data will continue to report, but alerts will not be sent if an incident is identified.

Extend a Maintenance Schedule

To extend a maintenance period, perform the following:

1. Navigate to any instance in your account.
2. In the **Details** section of the page, locate the **Maintenance** block.
   Click the **3-dot menu** and choose the duration of the extension.
3. 

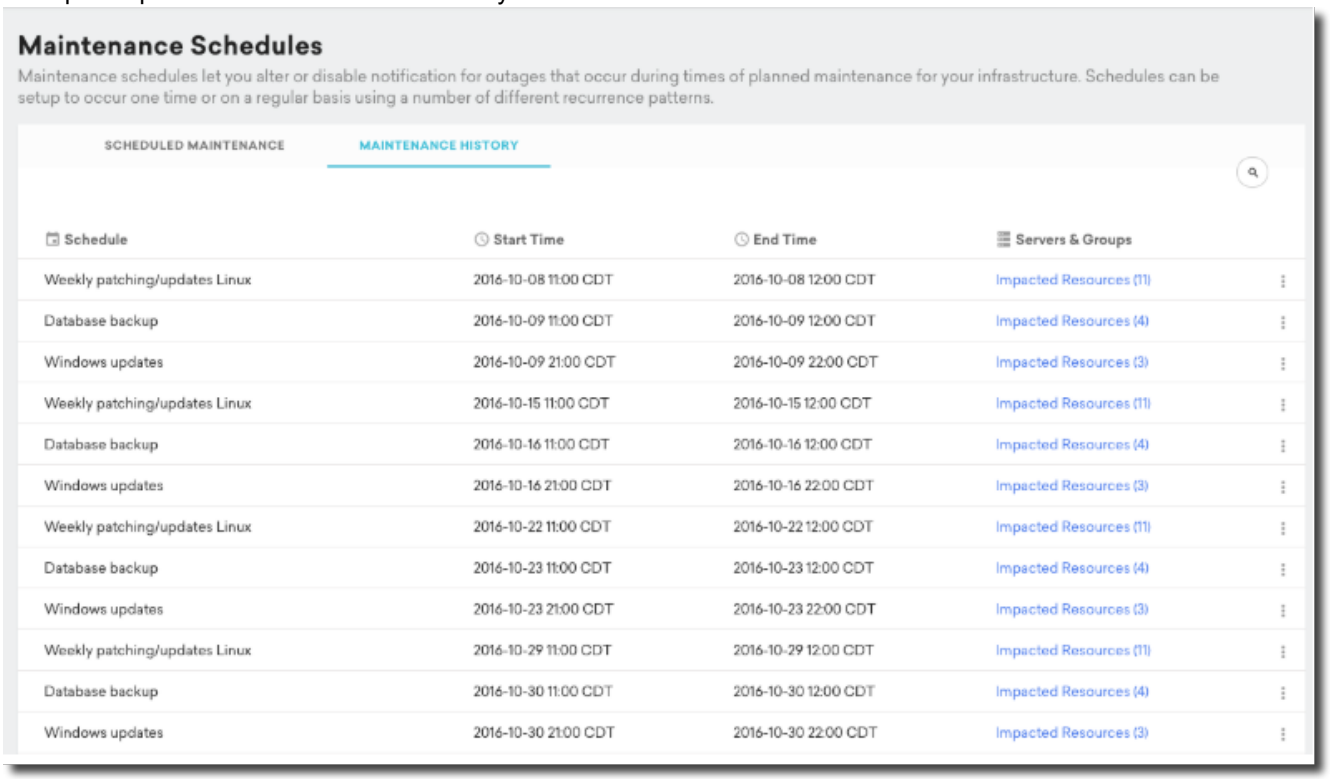
End an existing maintenance

To end an existing maintenance, perform the following:

1. Navigate to the instance.
2. In the **Details** section of the page, locate the **Maintenance** block.
   Click the **3-dot menu** of the active maintenance that you want to end then select **End Maintenance**.
3. 

Maintenance history

Your maintenance history is a very useful tool, after all you will need to be able to tell the difference between a planned and genuine outage. If you do not already know how to configure maintenance schedules, you may want to read this article. To access your maintenance history click on Monitoring and then “Maintenance Schedule,” then on the new page that opens up select the “Maintenance History” tab.



You can search through your maintenance history based on the server name, or limit your history by start date. You can also use your maintenance history to create new maintenance schedules. Just select the maintenance period you would like to replicate and click the “Copy” button to the right of it. A pop up will open up that will allow you to create a new maintenance schedule based on the old one.

Do not forget about your maintenance history, it is your friend! If a customer asks why you were down at a particular time, you will want to be able to say if it was scheduled maintenance or an unexpected outage.

API

- API keys
- View the API library
- Custom incidents using API
- Postman

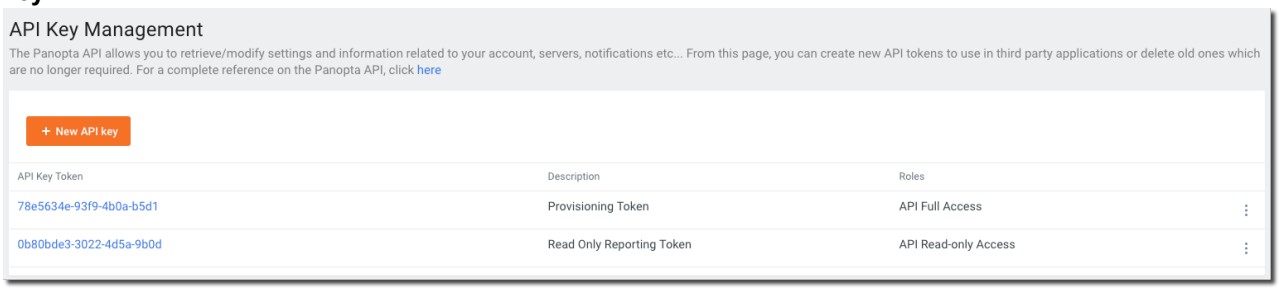
API keys

With the FortiMonitor API, you can achieve any level of automation from adding/deleting new instances, extracting data, starting a maintenance window to much more.

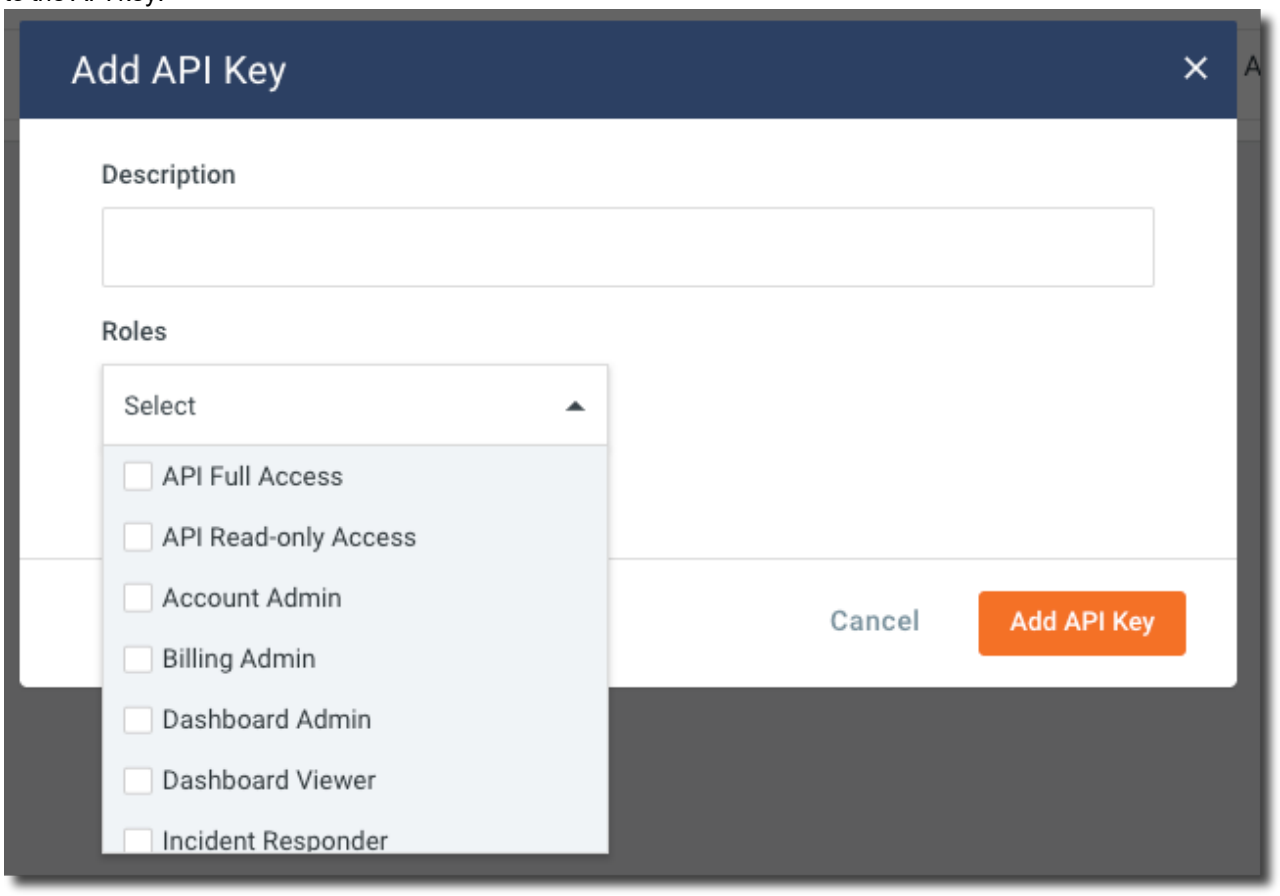
To start using the FortiMonitor API:

1. Generate a new secure API token by selecting **Teams & Activity** > **API Key Management** from the navigation menu.

   The **API Key Management** page shows you all of the existing API keys. To add a new API key, select **+ New API key**.

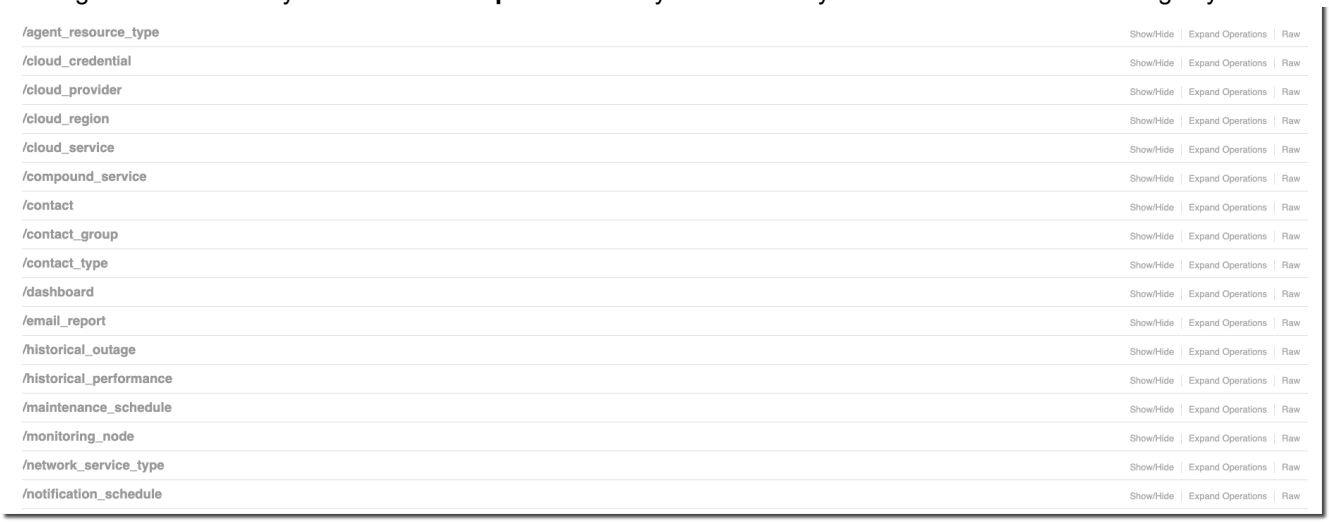
2. 

   From the **Add API Key** modal, add a description and then select the roles (depending on your use cases) to assign to the API key.

3. 

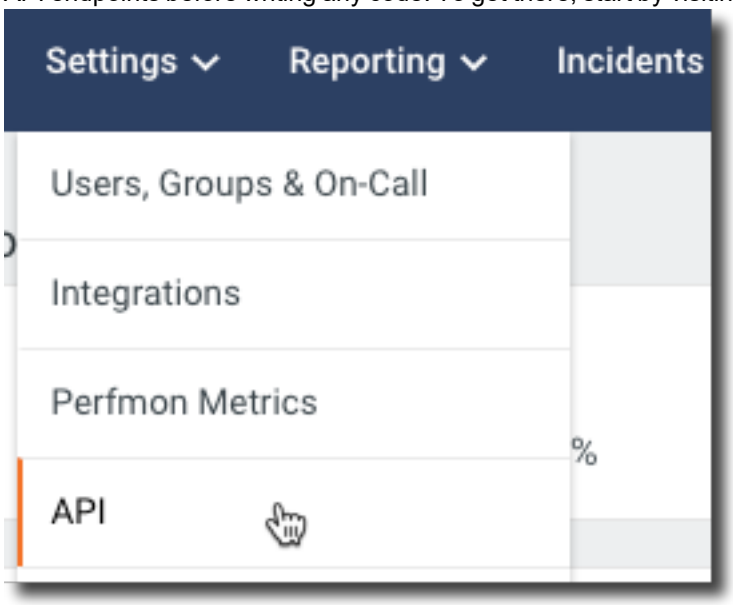
4. Once you select **Add API Key**, a new API key will be created.

Clicking on a token takes you into our **API explorer** so that you can visually interact with API before writing any code.

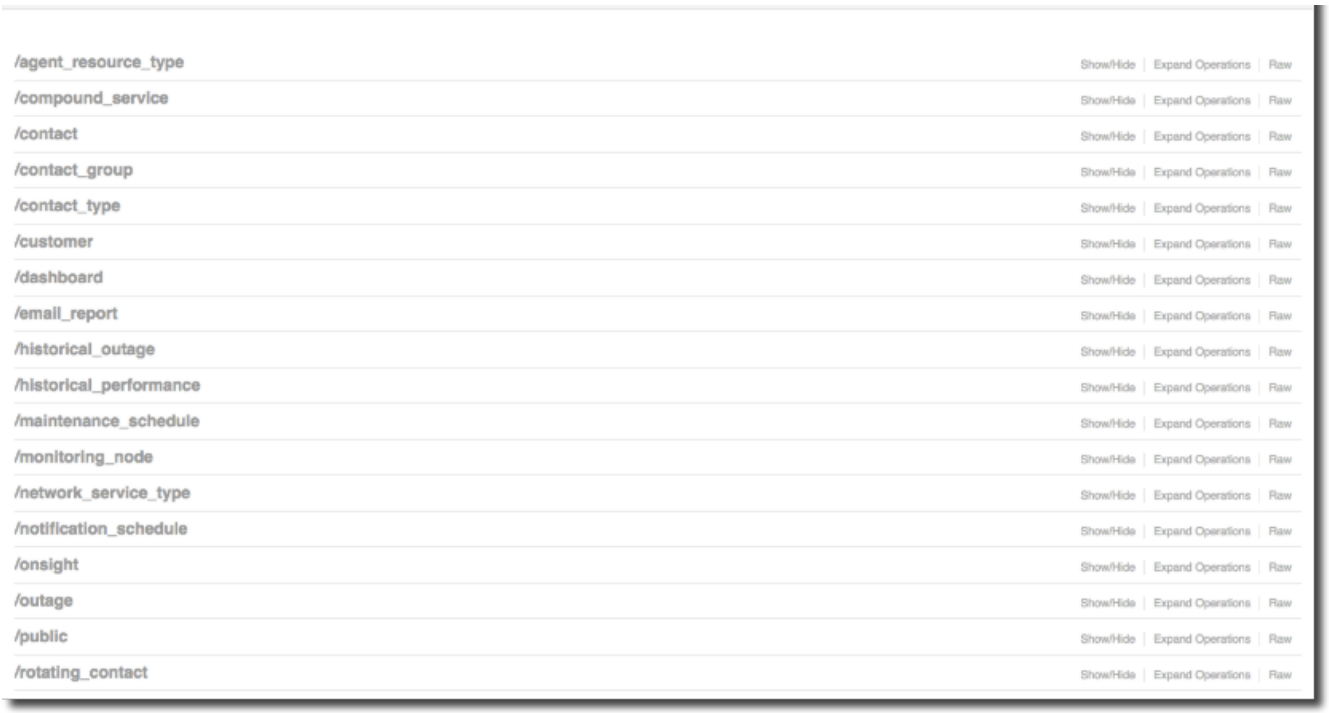


View the API library

The FortiMonitor API enables full feature parity between what you can do manually or through our API. To make it easy to use, we have implemented a Swagger based API explorer to allow you to visually inspect and interact with the various API endpoints before writing any code. To get there, start by visiting the API Key Management section.



Click on one of the existing API tokens or create a new one. This will take you to a page which looks like the below:

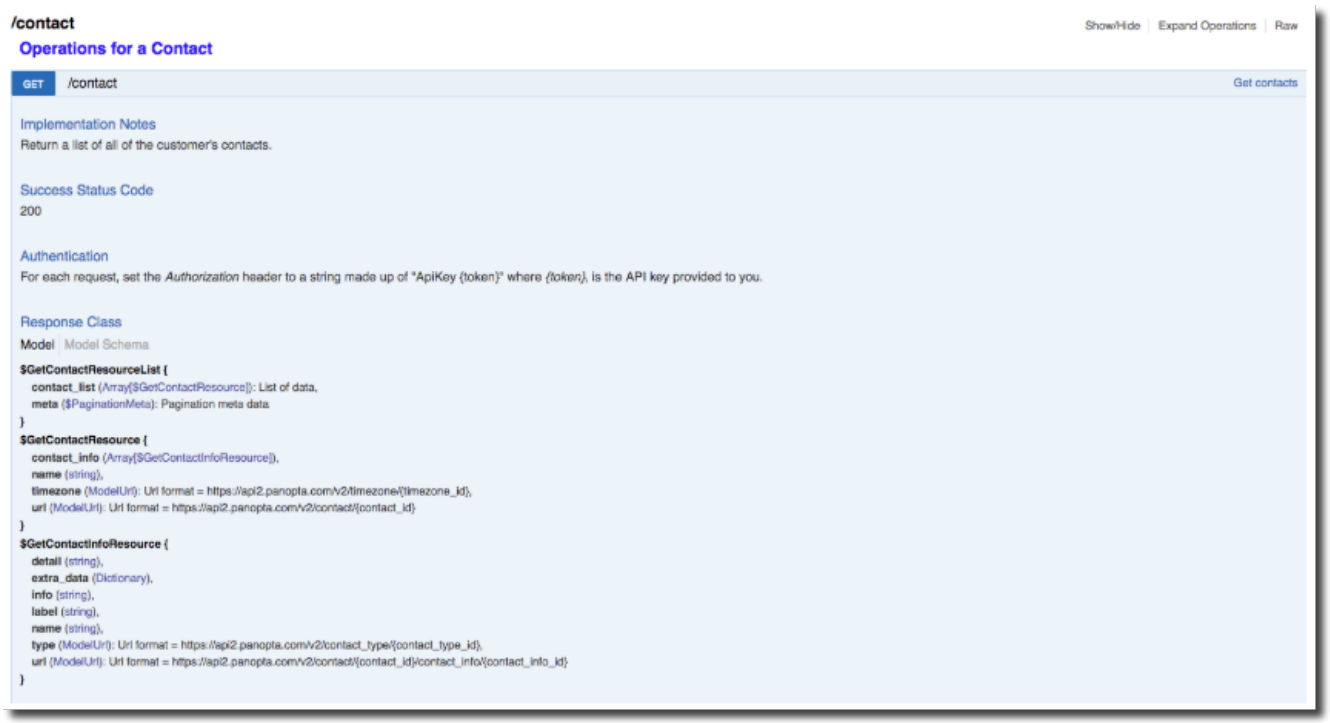


Your API key should be pre-filled in the upper right hand corner. This will automatically send your key to each request you attempt through the UI.

By clicking on Show/Hide for any of the listed categories you should see a list of all the operations that the category supports.



To see more details and any one operation you can click on that operation.



/contact

Show/Hide | Expand Operations | Raw

Operations for a Contact

GET /contact — Get contacts

Implementation Notes

Return a list of all of the customer's contacts.

Success Status Code

200

Authentication

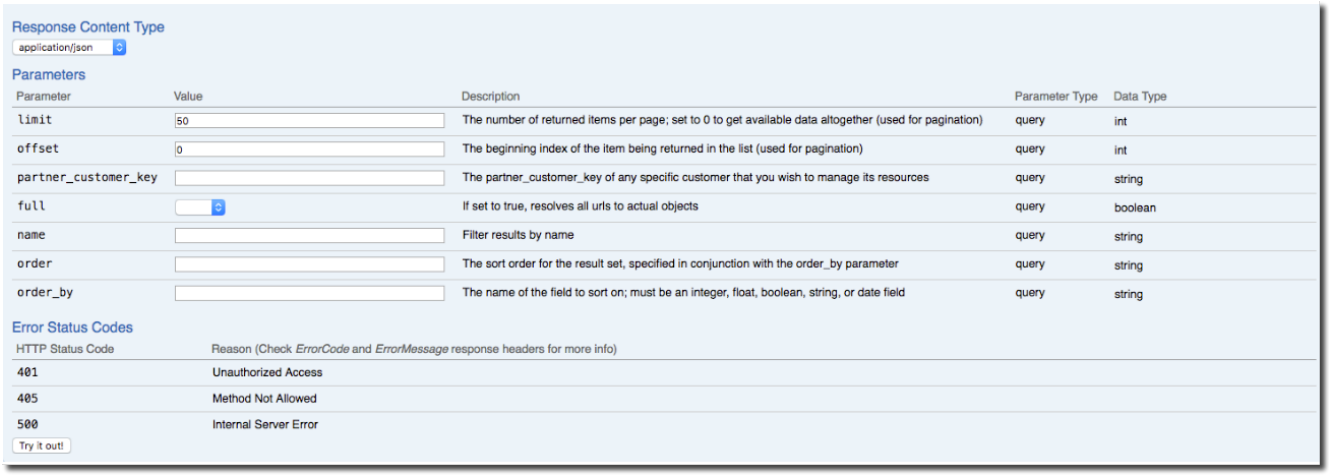
For each request, set the *Authorization* header to a string made up of "ApiKey {token}" where *{token}*, is the API key provided to you.

Response Class

Model | Model Schema

```
$GetContactResourceList {
  contact_list (Array[$GetContactResource]): List of data,
  meta ($PaginationMeta): Pagination meta data
}
$GetContactResource {
  contact_info (Array[$GetContactInfoResource]),
  name (string),
  timezone (ModelUrl): Url format = https://api2.panopta.com/v2/timezone/{timezone_id},
  url (ModelUrl): Url format = https://api2.panopta.com/v2/contact/{contact_id}
}
$GetContactInfoResource {
  detail (string),
  extra_data (Dictionary),
  info (string),
  label (string),
  name (string),
  type (ModelUrl): Url format = https://api2.panopta.com/v2/contact_type/{contact_type_id},
  url (ModelUrl): Url format = https://api2.panopta.com/v2/contact/{contact_id}/contact_info/{contact_info_id}
}
```

This detailed view will give you all the information you need to understand how the objects are constructed and what variables you will be working with. The detailed view also give you the option to set configure the parameter and content type of your response.



Response Content Type

application/json

Parameters

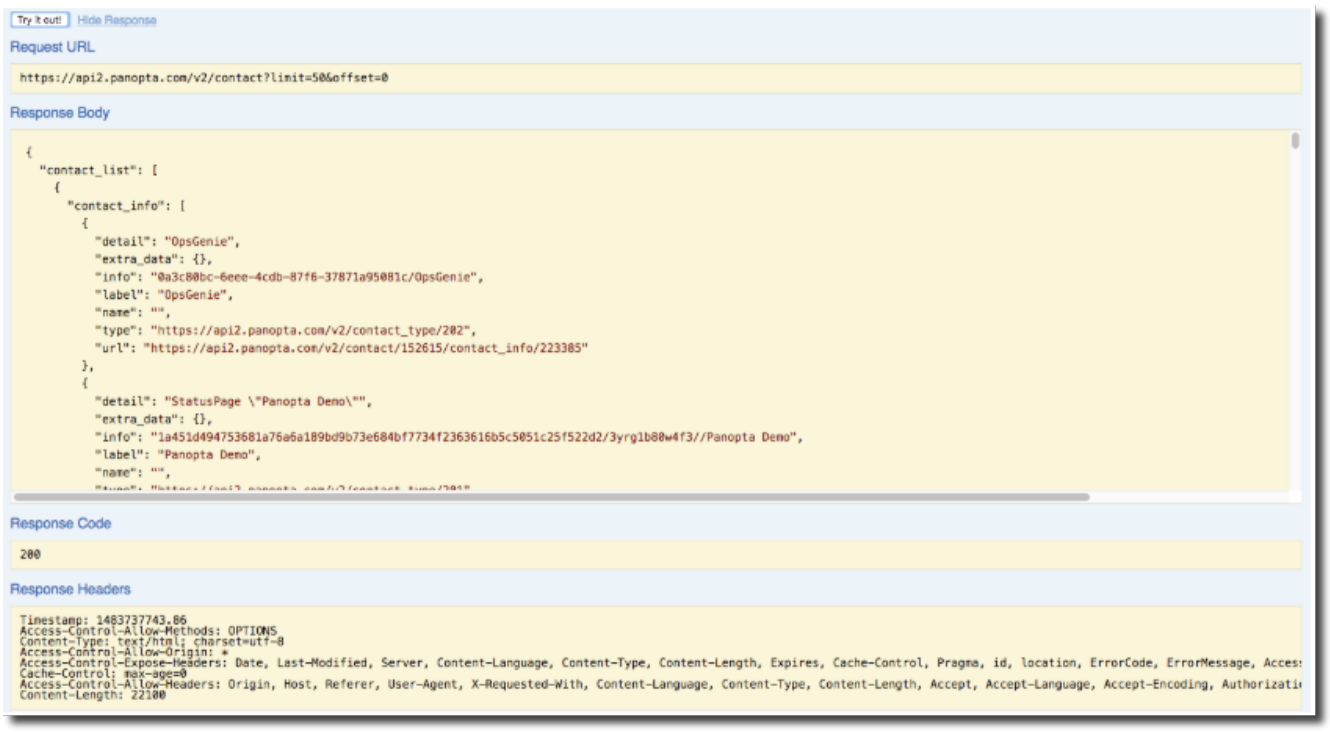
Parameter	Value	Description	Parameter Type	Data Type
limit	50	The number of returned items per page; set to 0 to get available data altogether (used for pagination)	query	int
offset	0	The beginning index of the item being returned in the list (used for pagination)	query	int
partner_customer_key		The partner_customer_key of any specific customer that you wish to manage its resources	query	string
full		If set to true, resolves all urls to actual objects	query	boolean
name		Filter results by name	query	string
order		The sort order for the result set, specified in conjunction with the order_by parameter	query	string
order_by		The name of the field to sort on; must be an integer, float, boolean, string, or date field	query	string

Error Status Codes

HTTP Status Code	Reason (Check *ErrorCode* and *ErrorMessage* response headers for more info)
401	Unauthorized Access
405	Method Not Allowed
500	Internal Server Error

Try it out!

The Try it out! button will run the request with the given configuration and show the response below



Custom incidents using API

FortiMonitor supports the creation of custom incidents using API, which provides a mechanism for your own applications to push event data to our system and leverage FortiMonitor's alerting and dashboard functionality.

To use the API to create and close custom incidents, the **Custom Incidents** feature must be enabled on your account. Please contact our support team to have it enabled.

Access FortiMonitor's API library to see all of the supported API endpoints in detail.

Add a custom incident filter

Before you can start sending custom incidents using the API, you need to enable support on the server whose agent will be sending the incident data. To add a custom incident filter, see this topic.

Create a custom incident

This API request allows you to create a custom incident on any of your servers in FortiMonitor. The location response header contains the full URL of the new resource.

Endpoint

```
POST /server/{server_id}/custom_incident
```

The **{server_id}** is the **Server ID** of the instance where this custom incident will be created.

Authentication

For each request, set the Authorization header to a string made up of **ApiKey {token}** where **{token}** is the API key provided to you.

Users sending API requests to FortiMonitor must have an API key and the proper user permissions.

To view and manage your API keys, see API keys.

Payload

```
{
"description": null,
"metadata": null,
"plugin_textkey": null,
"resource_textkey": null,
"start_time": null
}
```

The attributes are described in the following table:

Attribute	Data type	Required	Description
description	String	✓	Human-readable description of the incident, which is shown in the control panel and alerts.
metadata	Dictionary		(optional) Optional dictionary of key/value fields that will be added to the incident description.
plugin_textkey	String	✓	Textkey which specifies the application category the incident relates to.
resource_textkey	String	✓	Textkey which specifies the metric the incident relates to.
start_time	Date		(optional) Start time in UTC; format: YYYY-MM-DD HH:MM:SS. If not provided, defaults to current time.

Resolve an incident

This API request allows you to force resolution of a custom incident.

Endpoint

```
PUT /outage/{outage_id}/force_resolve
```

The **{outage_id}** is the **Incident ID** of the custom incident that you want to resolve using this endpoint.

Authentication

For each request, set the Authorization header to a string made up of **ApiKey {token}** where **{token}** is the API key provided to you.

Users sending API requests to FortiMonitor must have an API key and the proper user permissions.

To view and manage your API keys, see API keys.

Payload

```
{
"reason": null
}
```

The attributes are described in the following table:

Attribute	Data type	Required	Description
reason	String		(optional) The reason for ending the incident. This will be appended to the incident log.

HTTP status code summary

The HTTP status codes is part of the response message that provides information on the status of your API request. The following table describes the HTTP status error codes in detail:

HTTP status code	Reason
200 - OK	The API request is successful.
400 - Bad Request	The request failed, often due to a missing or misconfigured parameter. Check the response body for details on validation errors.
401 - Unauthorized Access	The API key provided is not valid.
404 - Not Found	The requested resource does not exist.
405 - Method Not Allowed	The server rejected the method for the requested resource.
500 - Internal Server Error	The API request failed due to an internal server error.

Check the **ErrorCode** and **ErrorMessage** response headers for more information.

Using Postman with FortiMonitor's API

Postman is a GUI based application that can interact with query RESTful APIs made by others or test ones you have made yourself. When exploring API endpoints, it's helpful to have a program that can make these requests without needing programming or network experience. We highly recommend using the **desktop version** which supports Windows, OSX, and Linux systems.

Authenticating with Postman

1. Obtain your API key here.
2. Open **Postman**.
3. Create a new request.
4. Under **Headers** add a new key for **Content-Type**.
5. Set the value to **application/json**.
6. Add a new Key for **Authorization**.
7. Set the value to **ApiKey $TOKEN**.
8. Test by getting a list of servers: 'https://api2.panopta.com/v2/server'

Your end result should look similar to this:

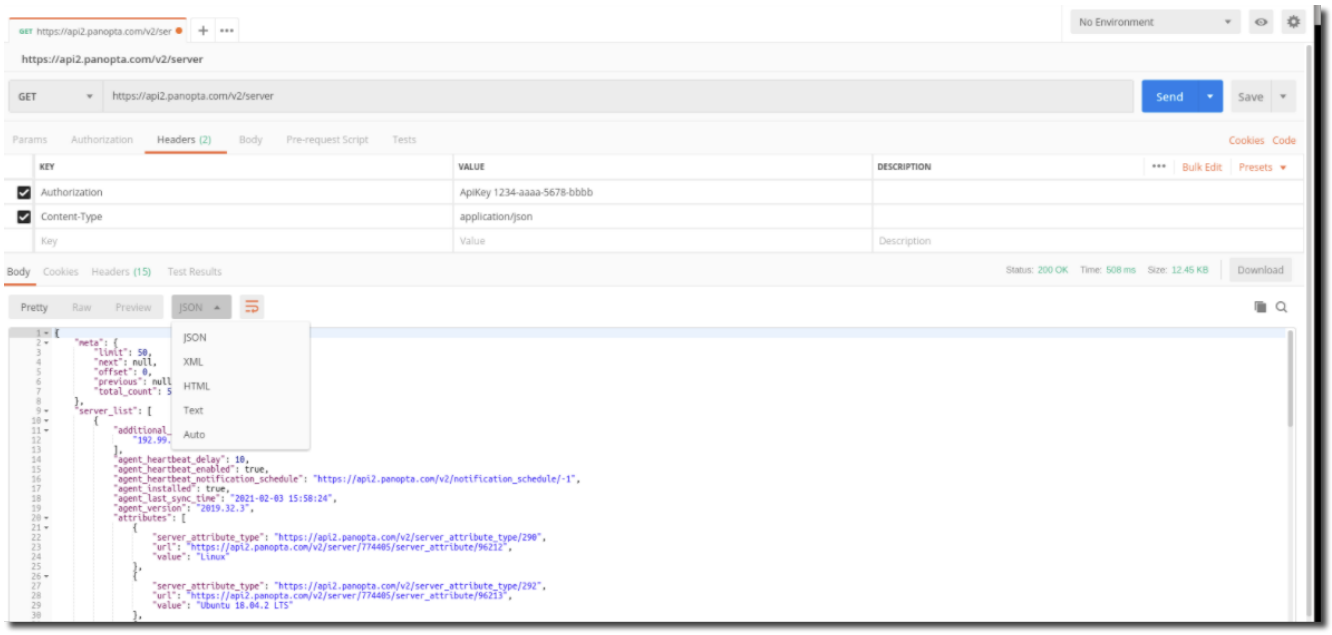


Using a GET request

In this example, we'll be gathering all servers in our account with a basic GET call.

1. Set the request method to **GET**.
2. Grab the **/v2/server** URL from the API explorer.
3. Send the request.
4. Once a response is received, under **Body** set the mode to **Pretty** and the type to **JSON** to view more easily readable responses.

Your output should be readable and contain information about your servers.

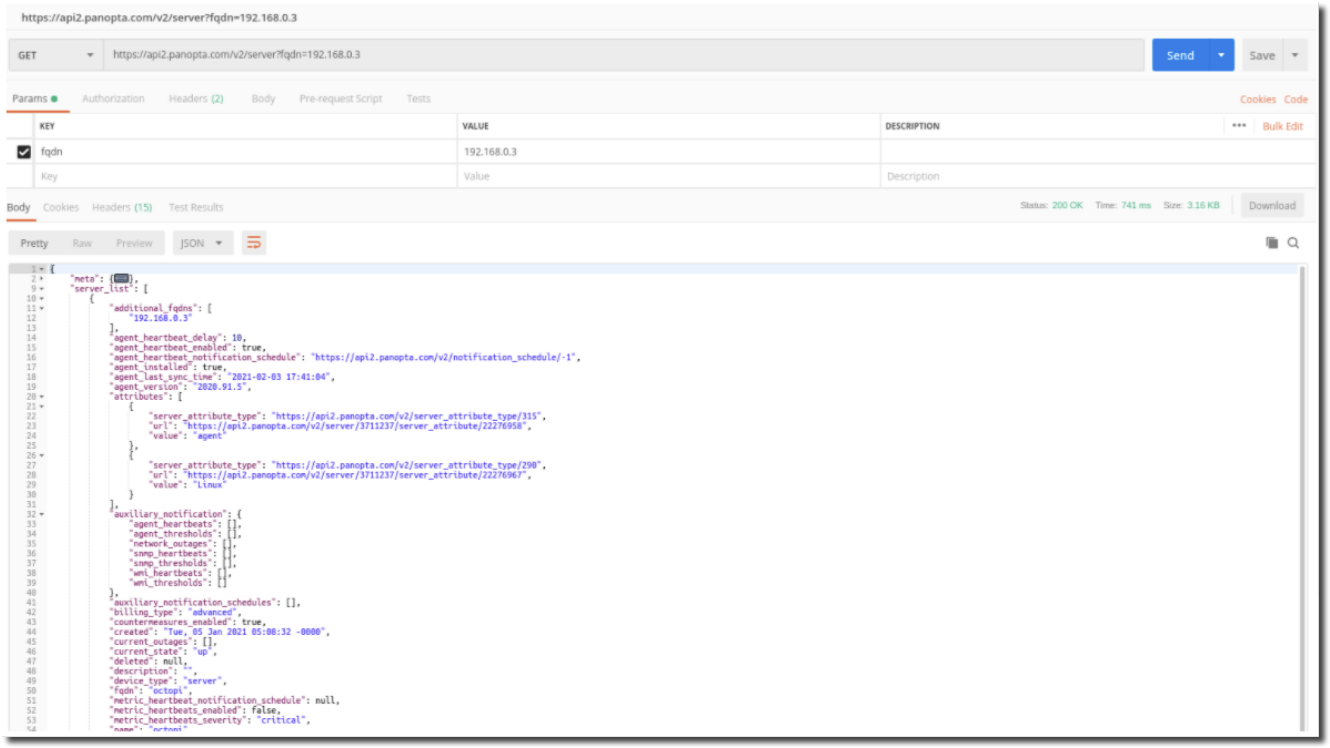


Filtering servers by FQDN

1. In the **Params** tab, create a new key labeled **fqdn**.
2. In the value section enter your IP or FQDN - here we're using `192.168.0.3`.



3. Send - Expected response **200 OK.**
4. Set your **Body** to **Pretty JSON** again and you can view the information for only servers that match that FQDN.



Using a POST request

In this example, we're creating a maintenance schedule for our security patches tonight at midnight UTC. We've disabled alerting, set it to end currently active outages, and exclude any outages that pop up from counting against our instance's availability percentage.

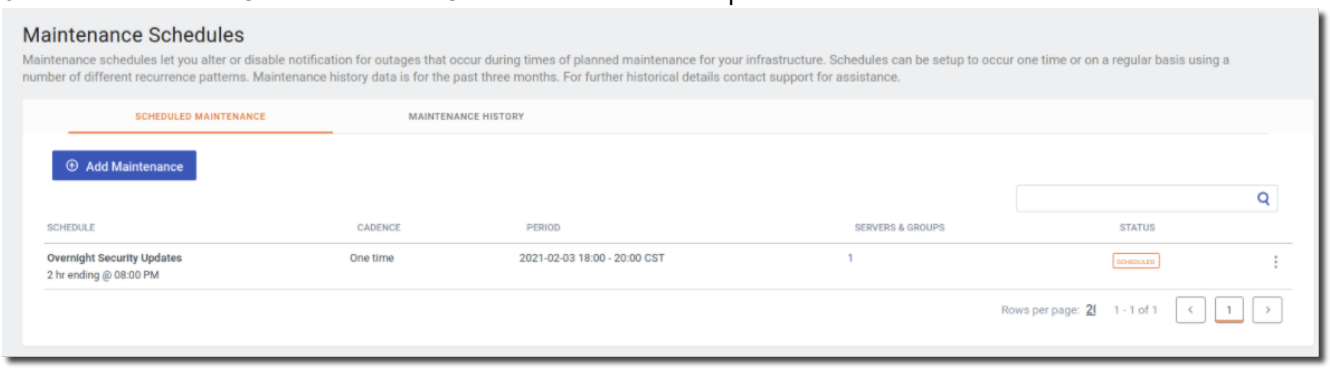
1. Query for your server using **GET** (see Using a GET request).
2. In the output look for **url** to grab from the response **Body**.
3. Set method to **POST**.
4. Build your JSON payload. The API explorer contains more description for each key but our example JSON payload

is below:

```
{
"description": "Night patches",
"disrupt_active_outages": true,
"duration": 120,
"exclude_from_availability": true,
"monitoring": "no_alerts",
"name": "Overnight Security Updates",
"original_start_time": "2021-02-04 00:00:00",
"pause_all_checks": false,
"recurrence_interval": 1,
"show_in_public_report": true,
"targets": ["https://api2.panopta.com/v2/server/4705786"]
}
```

5. Send - Expected response **201 CREATED**

6. Either use another **GET** or look in the Control Panel to see the updates.



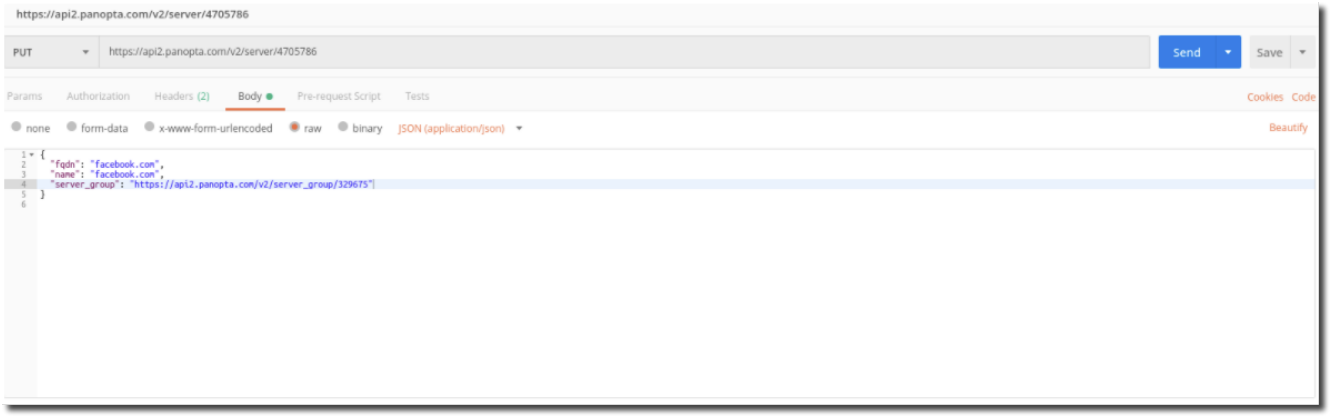
Using a PUT request

1. Get your server's **server_group URL** (see Using a GET request).
2. Create a new JSON payload with your updated name field:

```
{
"fqdn": "facebook.com",
"name": "facebook.com",
"server_group": "https://api2.panopta.com/v2/server_group/329675"
}
```

3. Send - Expected response **204 NO CONTENT.**

4. Either use another **GET** or look in the Control Panel to see the updates.



Using a DELETE request

1. Use **GET** method to search for your instance (see Using a GET request).
2. Modify the URL to be **https://api2.panopta.com/v2/server/$YOUR_SERVER_ID**.
   a. Here we're deleting the server **https://api2.panopta.com/v2/server/4705786**.
3. Set method to **DELETE**.
4. Send - Expected response **204 NO CONTENT.**
5. Either use another **GET** or look in the Control Panel to see the updates.



Multi-tenancy

The FortiMonitor platform is built for multi-tenancy at every level. Partners and managed service providers (MSPs) maintain a Master Tenant account which allows them to use FortiMonitor for yourself as well as maintain a single pane of glass view across all your subtenants. With a Master Tenant account, you can do the following:

- Allow subtenants to log in and manage their own isolated accounts
- Easily jump between different customers (subtenants)
- Programmatically manage customers via the API or manually through the control panel
- Centrally manage SSO configuration from a Master Tenant account
- View and create cross-customer dashboards
- Allow subtenants to use the Master Tenant account's OnSight vCollector
- Allow subtenants to use the Master Tenant account's Alert Timelines
- Share templates from the Master Tenant account
- Set up Monitoring Policies for all subtenants

Multi-tenant account requirements

Multi-tenancy requires an Enterprise license. Please reach out to your account manager for more details.

Key benefits

- Benefit from an actively developed and continually improved infrastructure monitoring solution
- Fully SaaS - no infrastructure to own or operate
- Full access to our support team for all escalated issues
- All-inclusive per node pricing. No additional costs
- Volume pricing

White labeling and reselling

- Run the control panel on your own custom domain name with SSL/TLS
- Customize the look and feel of your control panel with custom CSS and your own logo
- Customize all outbound email sender addresses and email templates with your own branding

Global infrastructure management

With a Master Tenant account, you can manage several aspects of your subtenants' infrastructure. The account allows you to share templates or monitoring policies, among other things, to your subtenants. You can also query data and build dashboards across any of your subtenant accounts. See the following sections:

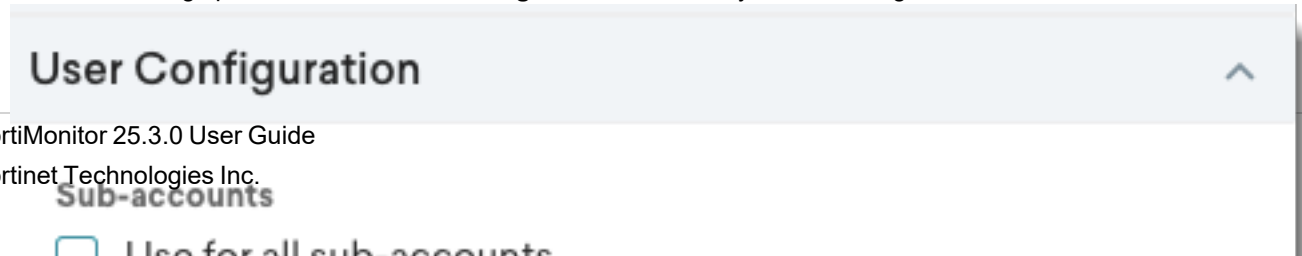
- Manage subtenants
- Manage a multi-tenant environment

Manage subtenants

This article describes the features and capabilities of a **Master Tenant** account. A **Master Tenant** account allows you to manage several aspects of your multi-tenant environment.

Use a single SSO configuration

With a **Master Tenant** account, you have the option to apply your SSO configuration to all of your subtenants. You can find the following option under the **User Configuration** module of your SSO integration.

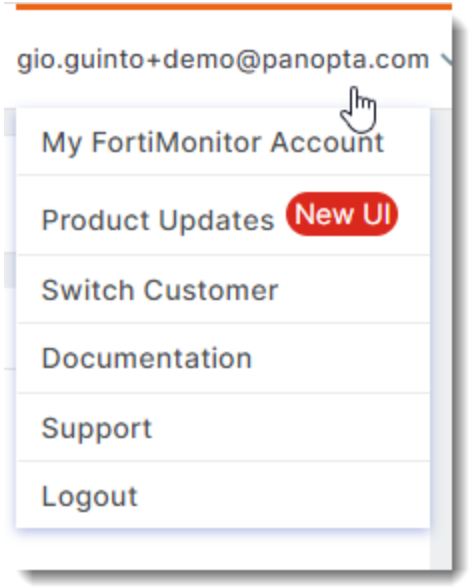


Select the **Use for all sub-accounts** option to use a single SSO configuration for all of your subtenants. Settings configured on the **Master Tenant** account such as **New User notification**, **Default Timezone**, and **Auto Create Users** are also applied to all other tenants. Newly created users are automatically added to the correct subtenant.

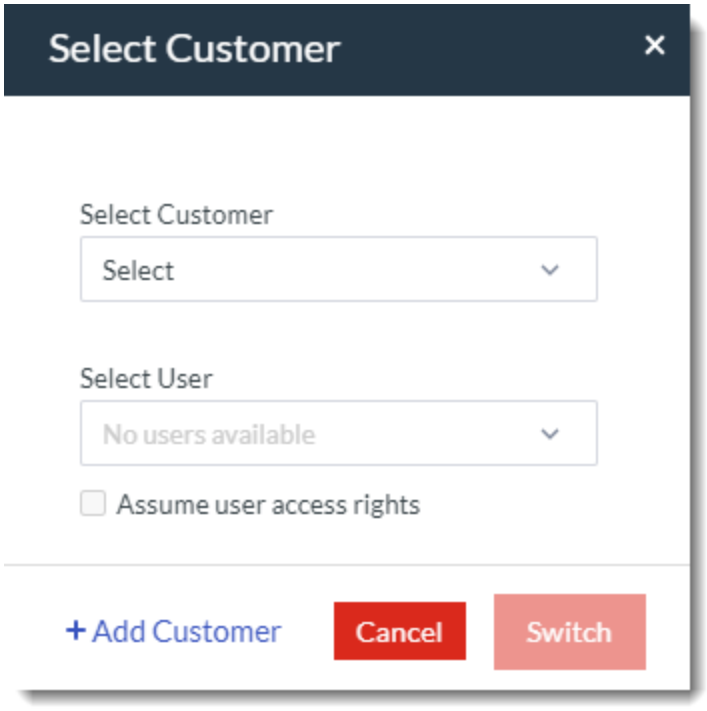
Switch between subtenants

You can use the **Customer Switcher** to switch between your subtenants. To switch to different subtenants, perform the following steps:

1. Click your account name then select **Switch Customer**.

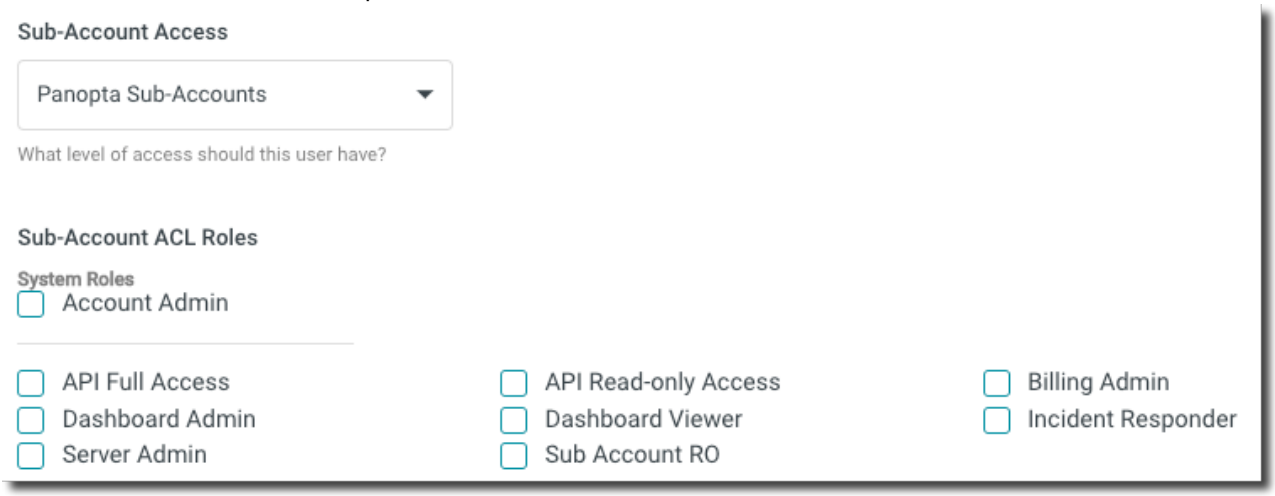


2. On the **Select Customer** modal, select the **Customer** and **User** that you would like to switch to then click **Switch**.



Add a user with access to all subtenants

You can add a user with specific access to all of your account's subtenants.

1. Select **Teams & Activity** > **Users, Groups & On-Call**.
2. Select **+ Add User**.
3. Enter the required information. The fields and options are described in Add users to your account.
   In the **Sub-Account Access** dropdown menu, Select **FortiMonitor Sub-Accounts**.
4. 

5. Assign the user the appropriate ACL roles and scope.
6. Select **Next** or **Contact Information** then fill out the form. The `Contact Information` pane allows you to add various contact methods for the user, which are then later used in Alert Timelines. However, only the email is required. You can also add labels (optional) to each contact information.
7. Select **Finish**. The new user is added to your FortiMonitor account.

Add users to subtenants

You can add FortiMonitor users to any of your account's subtenants by inviting them and configuring their ACL roles and scope. The user's access and permissions to the subtenant account can be configured depending on your requirements.

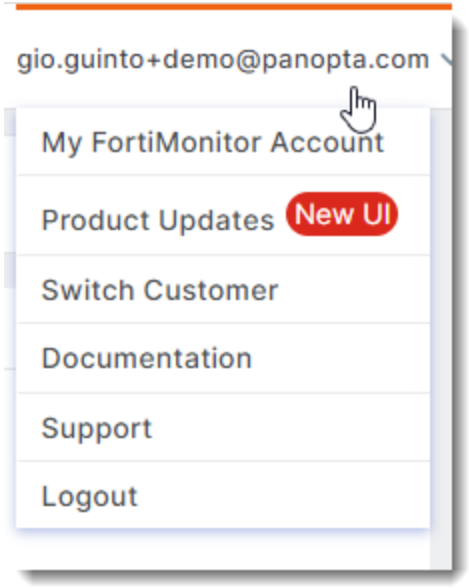
To add a user to a subtenant, perform the following steps:

1. Switch to the subtenant where you want to add the user.
2. On the navigation bar of a subtenant account, select **Teams & Activity > Users, Groups & On-Call**.
3. Click **+ Add** > **Invite Existing User**.
   Note: if you do not see this option, you need to have the feature enabled - please contact our support team.
4. On the subsequent **Invite External User** form, enter the email address the user utilizes to log in to FortiMonitor.
5. Enable the **Skip invitation email** option to activate the invited user without waiting for the invited user to respond to the activation email.
6. Assign the user the appropriate ACL roles and scope.
   You cannot edit the user's contact information - only they can edit it from within their primary account.
7. Click **Finish**. The user will be sent a confirmation email allowing them to confirm they'd like to become a member of your account. Once they accept it, they will be able to perform the actions you granted in step 4.

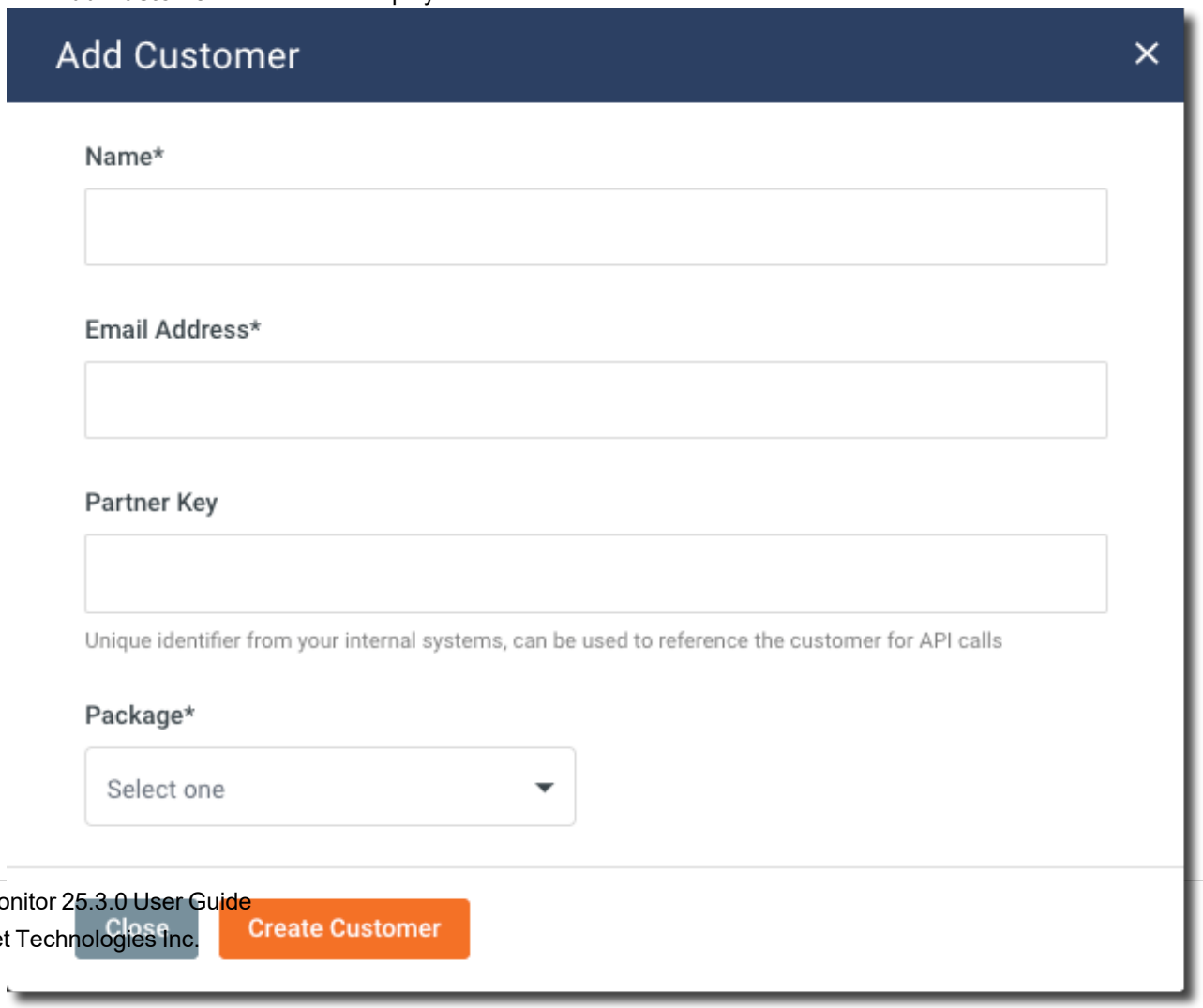
Create a subtenant

To create a new subtenant, perform the following steps:

1. Click your account name then select **Switch Customer**.



2. On the **Select Customer** modal, click **Add Customer**.
The **Add Customer** modal will be displayed.



3. Enter the required information. The fields are described in the following table.

Field	Description
Name	The name of the subtenant.
Email Address	The email address to be associated with the subtenant.
Partner Key	The unique identifier from your internal systems. This can be used to reference the subtenant for API calls.
Package	The monitoring package to assign to the subtenant. The list of available packages is set by FortiMonitor.

4. Click **Create Customer**. The new subtenant will be created.

Manage a multi-tenant environment

As a Master Tenant, you have the ability to manage several aspects of your multi-tenant environment. Read the following sections to learn more about managing your subtenants and sharing certain resources like OnSight vCollectors and templates.

Infrastructure map

The Infrastructure Map provides a high-level view of the health of your multi-tenant environment. You can follow the same Infrastructure Map experience to create customized views and explore the health of your nodes and subtenant's nodes to identify problems in your multi-tenant environment. For more information, see Infrastructure Map.

Dashboards

If you have a multi-tenant account, you can query data and build dashboards across any of your subtenant accounts. This is a powerful method to maintain your single pane of glass view across your different customer segments.

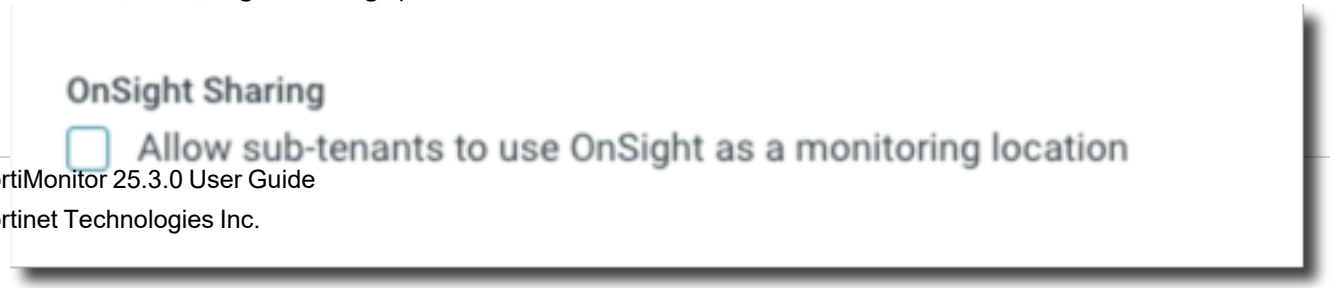
For more information, see Multi-tenant dashboards.

Share an OnSight vCollector

With a Master Tenant account, you can configure your subtenants to use an OnSight vCollector belonging to the Master Tenant account as a monitoring location.

To share an OnSight, perform the following steps:

1. Using your Master Tenant account, go to your OnSight vCollector's details page.
2. Click **Edit** to open the **Edit OnSight vCollector** drawer.
3. Enable the **OnSight Sharing** option.

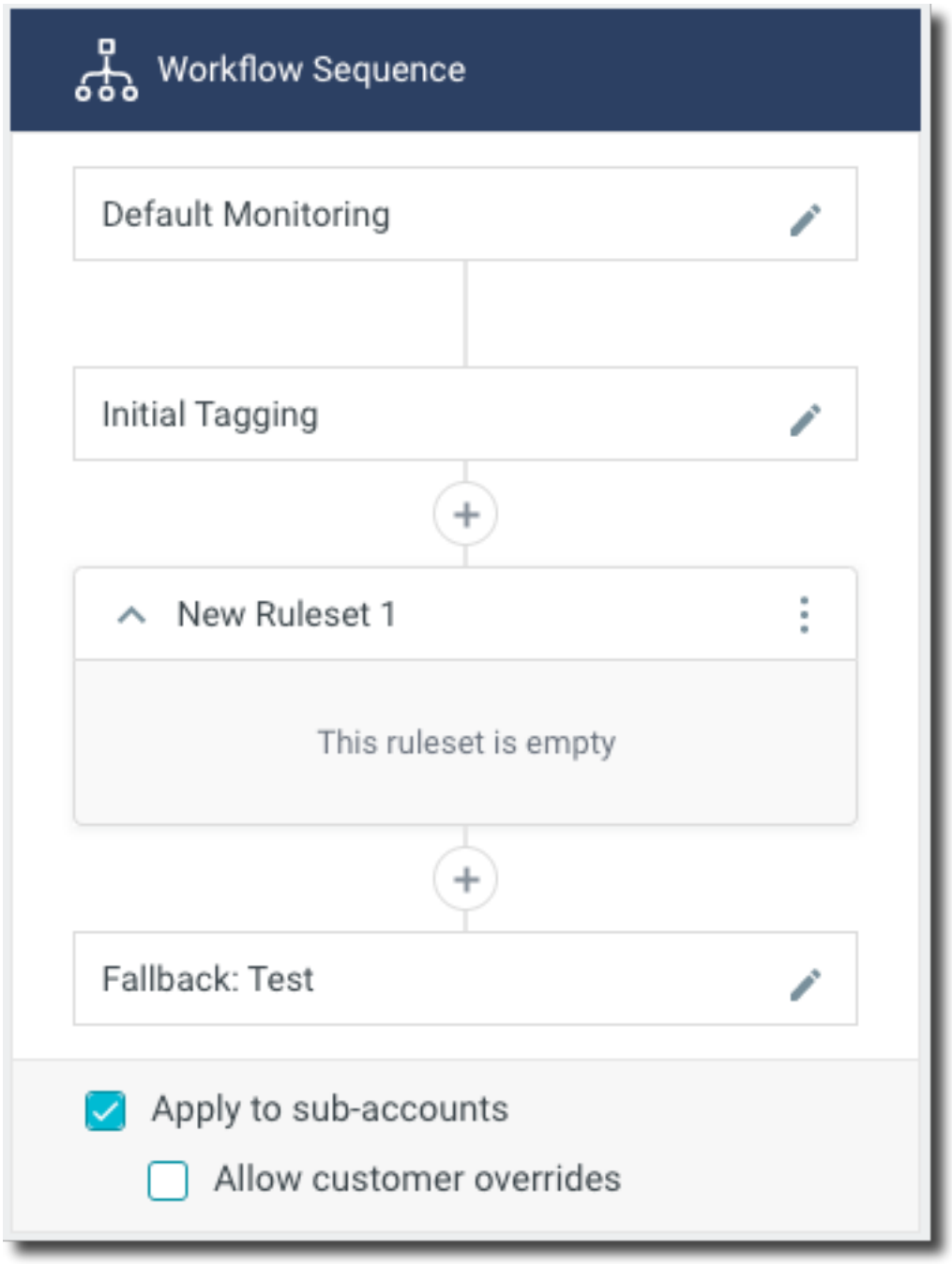


API

You can see a list of shared OnSight vCollectors via API GET calls to **/v2/onsight**.

Monitoring policies

You have the option to enforce your Master Tenant account's monitoring policy to all of your subtenants. This option can be found in your Master Tenant account's Monitoring Policies page. Select **Apply to sub-accounts** to enable the feature.



You can also allow subtenants to override this monitoring policy by enabling the **Allow customer overrides** option.

Templates

Templates created on a Master Tenant account are available to all subtenants. These templates are available to subtenants whether they are applying a template to an instance or instance group, or during initial instance configuration.

See Apply server templates.

Search

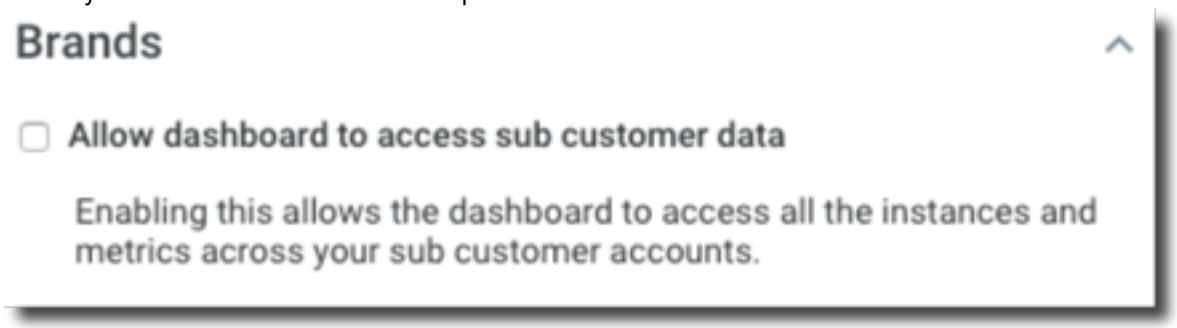
If you have a Master Tenant account, you can use the Global Search feature to search across your whole infrastructure, including subtenants, for customers or instances that match your search keywords. Global Search matches the keywords you entered to an instance's or customer's attributes, including the following:

- Customer
- Partner Customer Key
- Server Name
- FQDNs
- Partner Server ID
- Additional FQDNs or IP addresses

Multi-tenant dashboards

With a **Master Tenant** account, you can query data and build dashboards across any of the subtenant accounts. This is a powerful method to maintain your single pane of glass view across your different customer segments.

In order to start, create a new dashboard and select the **Allow dashboard to access sub customer data**. This option can only be set on new dashboards and upon initial creation.



At this point, you can follow the same dashboard experience to customize how to use the various options. There are four main differences:

Instance filtering
In **Multi-tenancy** mode, instance filtering is limited to **No Filter** and **Filter by Tag**. Filtering by tag will go across all subtenants. This makes it very important to architect and adopt an effective tagging strategy.

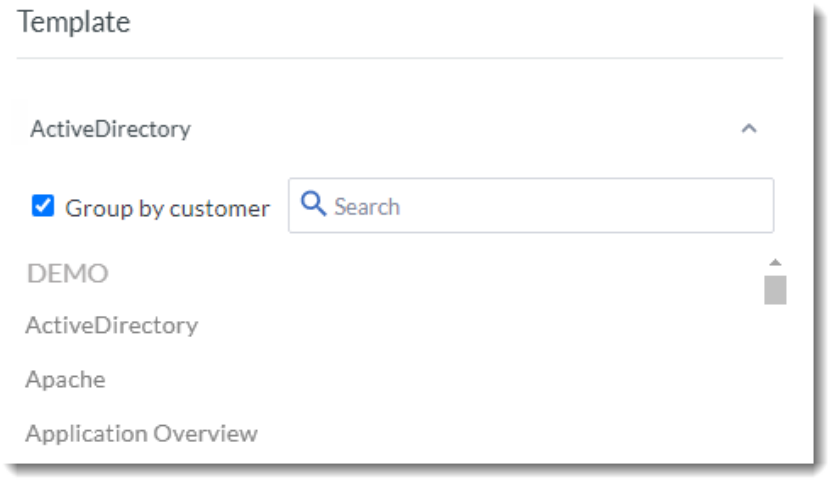
Inclusion of customer information
In certain widgets, the **Customer Name** will be an available field to show in tables and tooltips.

Limited dashboard widgets
Due to the scale/scope of the data being shown across customer accounts, certain dashboard widgets are not available in this mode. The current set of widgets in dashboard mode are:

- Active Incident Counter
- Active Incident Details
- Aggregated Metric
- Metric Heatmap
- Metric Summary
- Resolved Incident Counter
- Resolved Incident Details
- Server Heatmap
- Server Summary

Group by customer template option

Enabling the **Group by customer** option displays all of the available dashboard templates that you have access to and sorts the list by customer name. This allows you to accurately identify the source of the template.



For more details about each widget, please see Dashboard widgets.

White label options

White labeling FortiMonitor provides several customization options that change the look and feel of your control panel using custom CSS and your own logo. This article details all of the available options and delineates the items that you are required to provide for each option. To get started, contact your FortiMonitor Customer Success Manager (CSM). Your CSM will work with you to make sure that all white labeling customizations are applied.

Primary requirements

To utilize white label services, you must configure and provide the following:

- Create a CNAME entry on your existing domain pointing to **wl.monitorengine.com** (e.g. **monitor.companyname.com**)

- Provide an SSL certificate for the chosen domain.

Login and domain options

Please provide all of the SSL certificates required for the domain customization below to your CSM.

You can customize your domain and login information for the purpose of resale. If a custom domain is specified for any of the following options, provide the SSL certificate to FortiMonitor.

- **Base Domain** – this refers to the main sign-in page and defaults to **fortimonitor.forticloud.com**
- **Status Page Domain** – this refers to the account status page and defaults to **status.fortimonitor.forticloud.com**
- **Control Panel URL** – this refers to the specific control panel URL and defaults to **https://fortimonitor.forticloud.com**

SSL options

If you are using a domain that is not “fortimonitor.forticloud.com” you will need to provide the SSL keys and details.

- Once SSL details are added, they will be activated at the next scheduled production release (usually occurs twice a week).
- Once activated, you must ensure domain name (via CNAME DNS registration, see above. The newly branded control panel will then be available.

Note: SSLs in PEM or CRT/KEY format are preferred. Windows PFX will require your decryption password from and is not a preferred format.

- An SSL (x509) key
- An SSL certificate tied to the intended white-label domain
- Any intermediate and root certs, usually provided by the SSL authority when buying the SSL
- The FQDN intended to be used

Email options

Instructions for adding SendGrid to domain SPF are found here. Details regarding SPF are found here.

Emails generated by the FortiMonitor platform use **@fortimonitor.com** for all outbound emails. If you want to use a custom email domain, you need to add SendGrid to your domain SPF requirements. You can alter the domain for the following email addresses:

- **Support Email** (for new user invites) - default is **notice@fortimonitor.forticloud.com**
- **Outage** and **Alert Email** - default is **alerts@fortimonitor.forticloud.com**

Logo and favicon options

If you want to add your logo, you must submit a URL of a hosted image with a transparent background in PNG format, sized roughly **200px by 50px**.

To add a Favicon, you must submit a URL of a hosted image with a transparent background in PNG format, sized roughly **40px by 40px**. This is usually a cropped logo image or similar visual suited for button icons.

If you want to customize your branding in the FortiMonitor control panel, you have the following options:

- **Control Panel Logo URL** – this icon appears in the top left corner of the control panel and defaults to the FortiMonitor logo.
- **Favicon URL** – this icon appears in the upper left corner of the FortiMonitor control panel.

Custom CSS and JS

You must host your custom CSS and JS files. Send the applicable URLs to your FortiMonitor CSM for import.

Further customization of the FortiMonitor control panel may be achieved using custom CSS and Javascript overrides to the standard FortiMonitor configuration. The following options are available:

- **Custom CSS URL** – this is the link your customized CSS style sheet, otherwise, the default FortiMonitor CSS is used
- **Custom JS URL** – this is the link to your customized Javascript resources, otherwise, the default FortiMonitor JS is used
- **Custom Status CSS URL** – this is similar to the Custom CSS URL but is applied specifically to the **Status Report** page
- **Custom Status JS URL** – this is similar to the Custom JS URL but is applied specifically to the **Status Report** page

Mobile app

The FortiMonitor mobile app allows your team to easily access and manage incidents while on mobile. The app is a mobile version of the Incident Hub, ensuring that your team has access to active incidents while on-call. All of the actions that you can perform from the Incident Hub are also available in the mobile app, providing an easy way to deal with incidents and alerts.

- View active incidents in your infrastructure.
- Acknowledge incidents
- Schedule maintenance
- Delay, cancel, or escalate alerts.

Install and configure the mobile app

Install from the Apple App Store and Google Play

The FortiMonitor app is available from the **Apple App Store** and **Google Play Store**.

Configure push notifications

You must enable Push notifications, both on mobile and the FortiMonitor control panel, in order to receive alerts and notifications from the FortiMonitor mobile app.

Mobile configuration

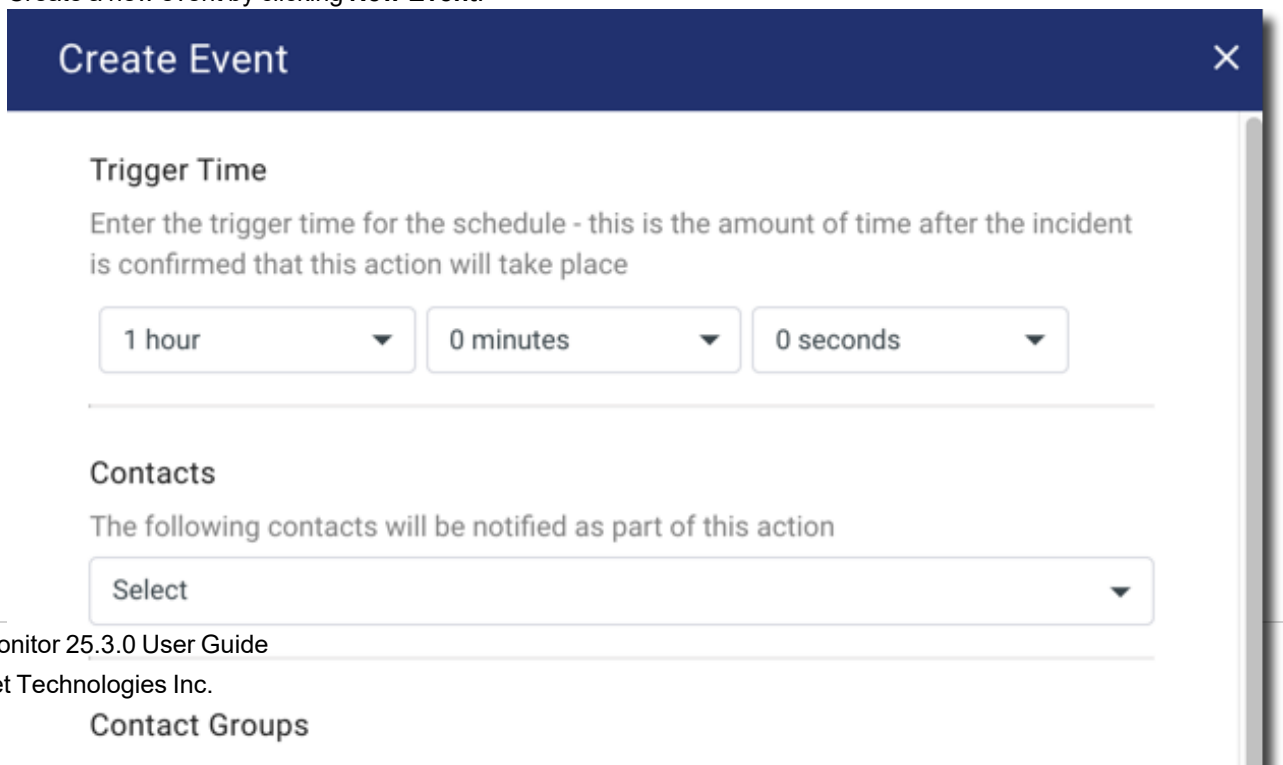
To enable push notifications, perform see the following OS-specific instructions:

- Apple
- Android

Control panel configuration

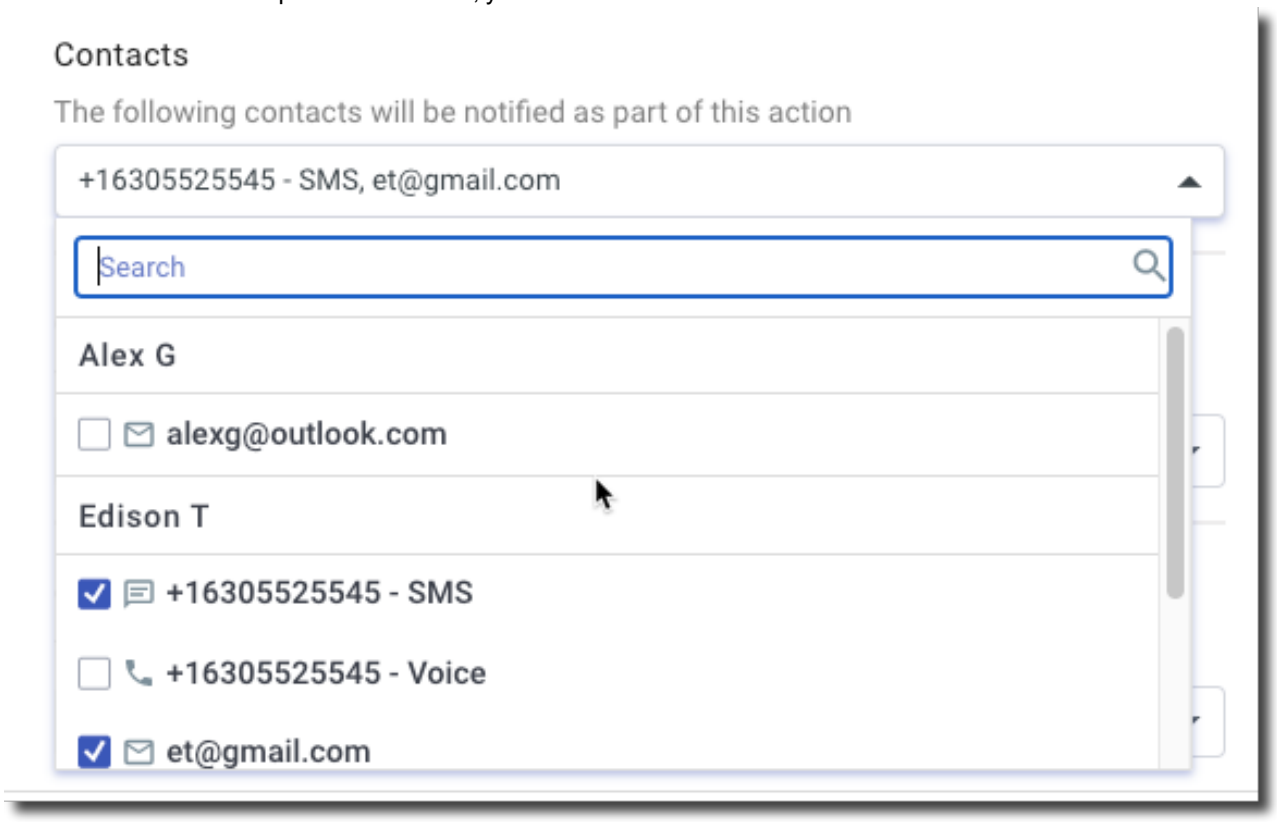
After configuring push notifications, you need to configure an Alert Timeline to use with the mobile app.

1. Open a web browser and log in to FortiMonitor.
2. From the navigation menu, select **Incidents** > **Alert Timelines**.
3. Select the **Alert Timeline** that you want to edit.
   Create a new event by clicking **New Event**.



5. Configure the trigger time. This will determine the amount of time before a notification is sent to the mobile app when an incident is confirmed.

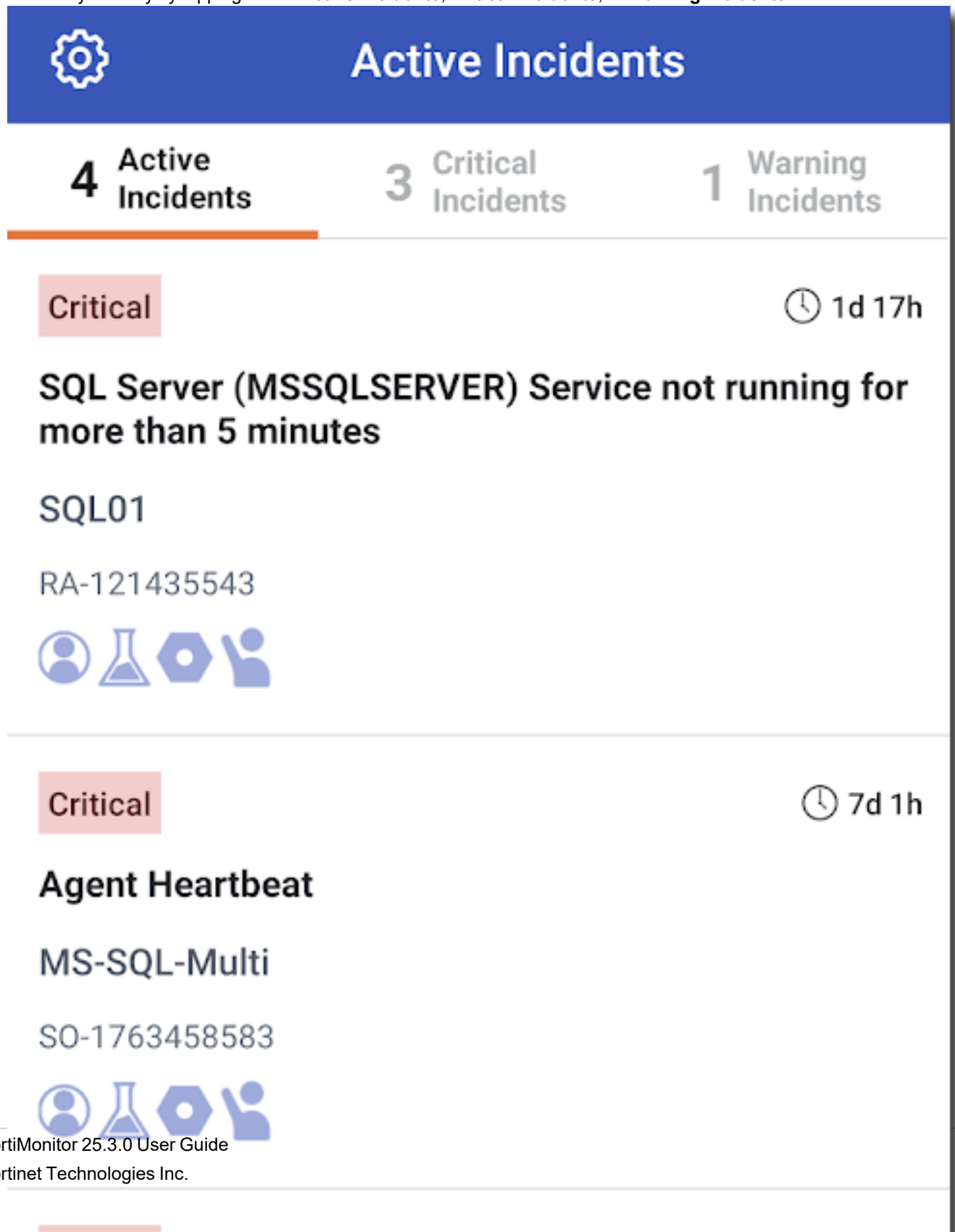
   Under **Contacts**, pick any combination of users, groups, on-call schedules, and integrations to add to the event. Because you can define multiple events for escalations, you can pick the appropriate level of noise for each event. For individual users, you can pick between email, SMS, phone call, or mobile push notification.
   Note: In order to receive push notifications, you need to select a mobile number.



6.
7. Click **Create Action**.

View Active Incidents

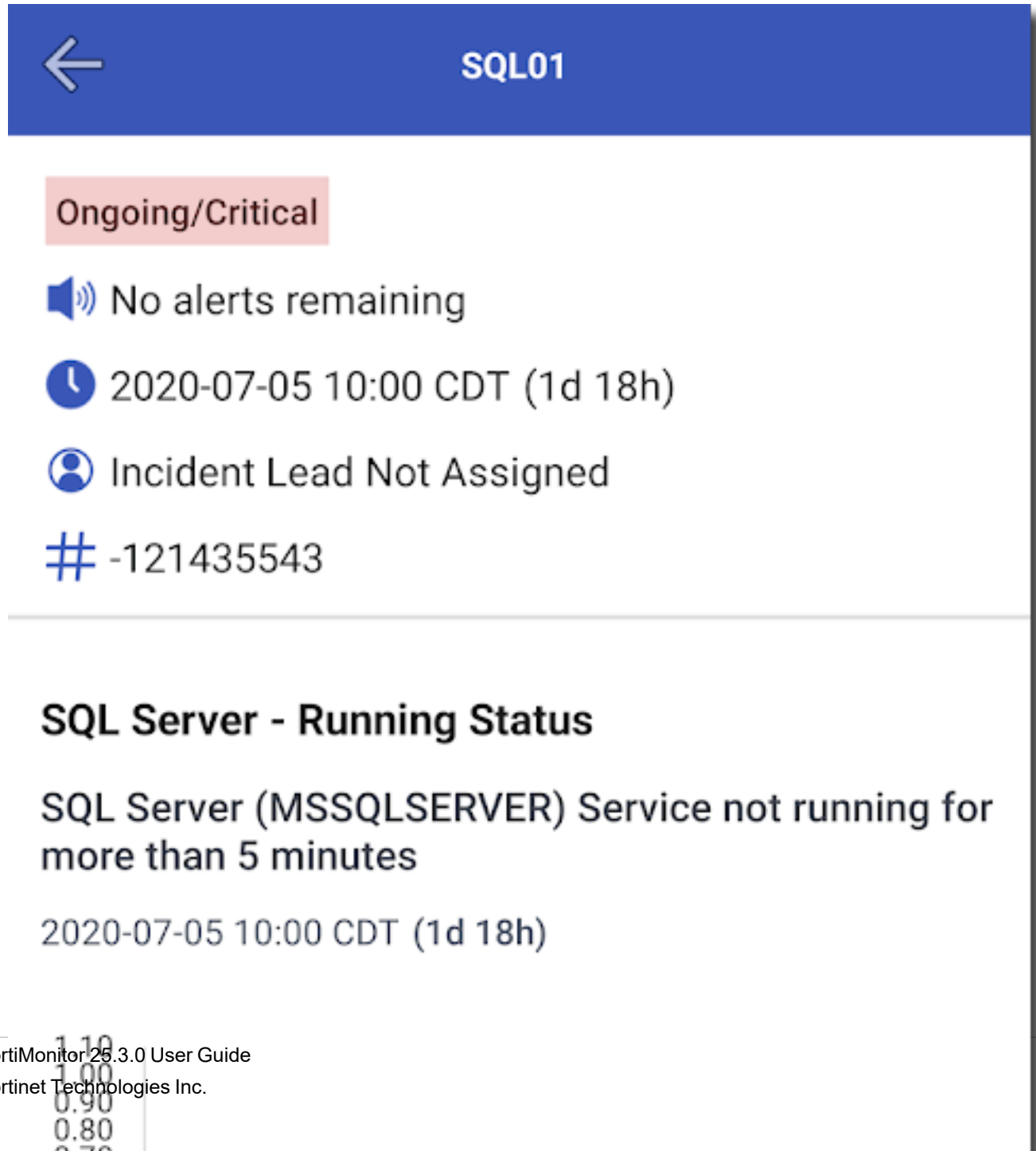
After logging in, the app will display all of the active incidents in your infrastructure. The list of active incidents can be refined by severity by tapping on the **Active Incidents**, **Critical Incidents**, or **Warning Incidents** tabs.



View incident details

You can view an incident's details by clicking on the incident. The incident's details will be displayed. This will include relevant information on the incident such as:

- Succeeding alerts, if any
- Incident start time and duration
- Incident lead assigned to the incident
- Incident ID
- The metric affected by the incident, including relevant metrics pertaining to the incident
- CounterMeasure details



From the page, there are also several actions that you can perform the following:

- Acknowledge the incident
- Schedule maintenance
- Delay, escalate, or cancel alerts

Acknowledging incidents and other actions

The following actions are available from the mobile app:

- Acknowledge an incident
- Schedule Maintenance
- Delay Alerts
- Cancel Alerts
- Escalate Alerts

Acknowledge an incident

Acknowledging an incident marks the incident with your name. You also have the option to add a message to the incident log and broadcast that message to all of the users and integrations configured in the incident's Alert Timeline. To acknowledge an incident, perform the following steps:

1. Tap an active incident to open it.
2. Tap **Ack**. The **Acknowledge Incident** page will be displayed.
3. From this page, you can also see the following:
   - The Incident Lead
   - The list of users who are scheduled to get alerts based on the incident's Alert Timeline
4. (Optional) Add a message to the incident log.
5. (Optional) Enable the **Broadcast Message** option.
6. Tap **Submit**.

Schedule Maintenance

To schedule maintenance from the mobile app, perform the following steps:

1. Tap an active incident to open it.
2. (Optional) Edit the **Maintenance Name**.
3. Enter a **Start Time**.
4. Select a **Maintenance Duration** from the available options.
   - 5 minutes
   - 15 minutes
   - 30 minutes
   - 60 minutes
5. Select a Monitoring option:
   - **Continue monitoring but don't create incidents** –All monitoring for the instances covered by the Maintenance Period will continue - however, incidents will not be created, regardless of the monitoring data collected.
   - **Pause monitoring** – All monitoring for the instances covered by the Maintenance Period will be paused - meaning no incidents will be created and no alerts will be sent.
   - **Continue monitoring & creating incidents but don't send alerts** – All monitoring for the instances covered by the Maintenance Period will continue and incidents will be created accordingly - however, no alerts will be sent for any detected incidents.
   - **Continue monitoring, creating incidents, and sending alerts** – Things will operate as normal - however, an Alert Timeline override must be selected to be used for alerting.
6. You can also choose to enable the following options:
   - **End applicable incidents when the period begins** – If checked, any active incidents that are present when maintenance begins will be force-closed.
   - **Show indicator on status page** – If checked, a Maintenance badge will be displayed on your FortiMonitor Status pages if you are using them.

Delay Alerts

Delaying an alert postpones all of the future alerts on the Alert Timeline for a specified length of time. To delay alerts on one or more incidents, perform the following steps:

1. Tap an active incident to open it.
2. Select the duration you want the alert to be delayed:
   - 15 minutes
   - 30 minutes
   - 45 minutes
   - 60 minutes
3. Tap **Submit**.

From this page, you can also see a timeline of upcoming alerts including team members who are next in line to be alerted and their contact methods.

Cancel Alerts

Cancel Alerts stops all upcoming alerts for the selected incident. To cancel alerts, perform the following steps:

1. Tap an active incident to open it.
2. Tap **Cancel Alerts**.



Escalate Alerts

Escalating an alert immediately sends the next alert configured in the Alert Timeline. To escalate alerts, perform the following steps:

1. Tap an active incident to open it.
2. Tap **Escalate**.



NCM

- NCM installation and initial configuration
- NCM supported devices
- Backup
- Search syntax help
- Mass Config Push
- NCM Slack configuration
- System login
- Discovery
- Scheduling
- User security and access roles
- Sensitive data stripping

NCM installation and initial configuration

This article describes the steps on how to install and configure the FortiMonitor Network Configuration Management (NCM).

Note: NCM supports Fabric devices but that the IP address for the devices may not get passed by the Fabric integration and therefore may need to be manually updated on the device's **Instance Details** page

Prerequisites

Before getting started, you must have the following:

The NCM feature must be enabled on your account. Please contact our support team to enable this feature.

- **The latest version of the OnSight on the server where you want to install FortiMonitor NCM.**
  - The OnSight must be running on a system that has SSH access to the network devices you want to use the NCM on.
  - SSH login credentials for these network devices.
  - Access-control lists and routing must also be in place to allow SSH from the OnSight to the network devices.
  - Shell access to the OnSight to allow installation using the command line. Ideally, use SSH so that you can copy and paste commands easily. You can also SSH into the OnSight or use the console of your hypervisor (for example, VMware).
  - The Onsight must be in an OnSight Group. To create an OnSight group or add an OnSight to an existing group, see Set up FortiMonitor OnSight in HA or for failover. Note that it is possible to install NCM with a single OnSight. However, the OnSight must still be added into a group even if it's the only one there.
- **Network devices that are already in FortiMonitor.**
  - The monitoring location of these devices must be the OnSight where you will install the NCM. Go to a device's Instance Details page to see or change its monitoring location.
  - Ping or SNMP monitoring must already be setup for these devices, with the OnSight used as the monitoring location
- Outbound network access to **https://licensing.unimus.net**.

Install the NCM

To install the NCM, do the following:

1. SSH into your OnSight vCollector.

2. Run the following command to confirm that you have the latest version of the OnSight.

```
onsight check-upgrade
```

a. If the OnSight's version is **2020.68 or later**, update it by running the following command:

```
onsight upgrade
```

b. If the OnSight's version is **2020.50 to 2020.67**, update it by running the following command:

```
curl -fsSL https://packages.panopta.com/install/onsight/onsight.sh --output /usr/bin/onsight
```

Then run:

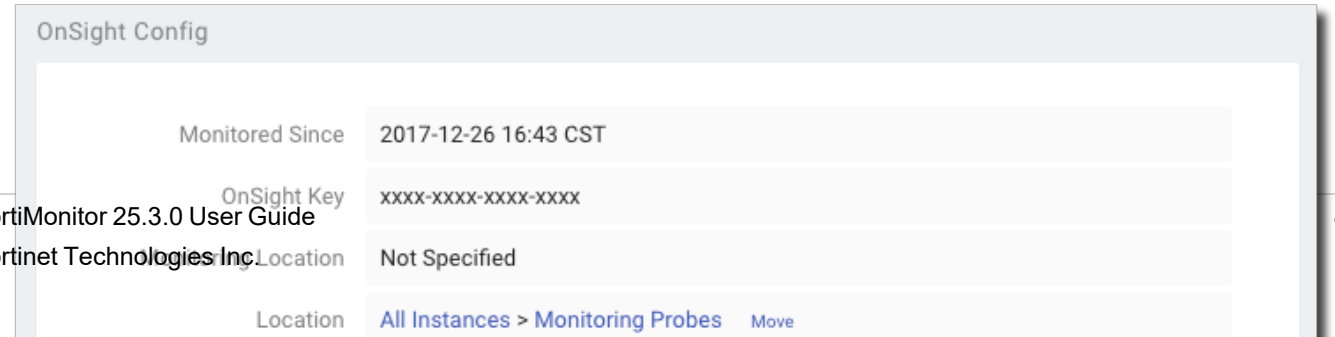
```
onsight upgrade
```

c. If the OnSight's version is **2020.49 or lower**, use the instructions detailed in this article to update it.

3. Install the NCM by running the following command. It will take up to 5 minutes to complete the installation and for the NCM to initialize.

```
onsight ncm-install
```

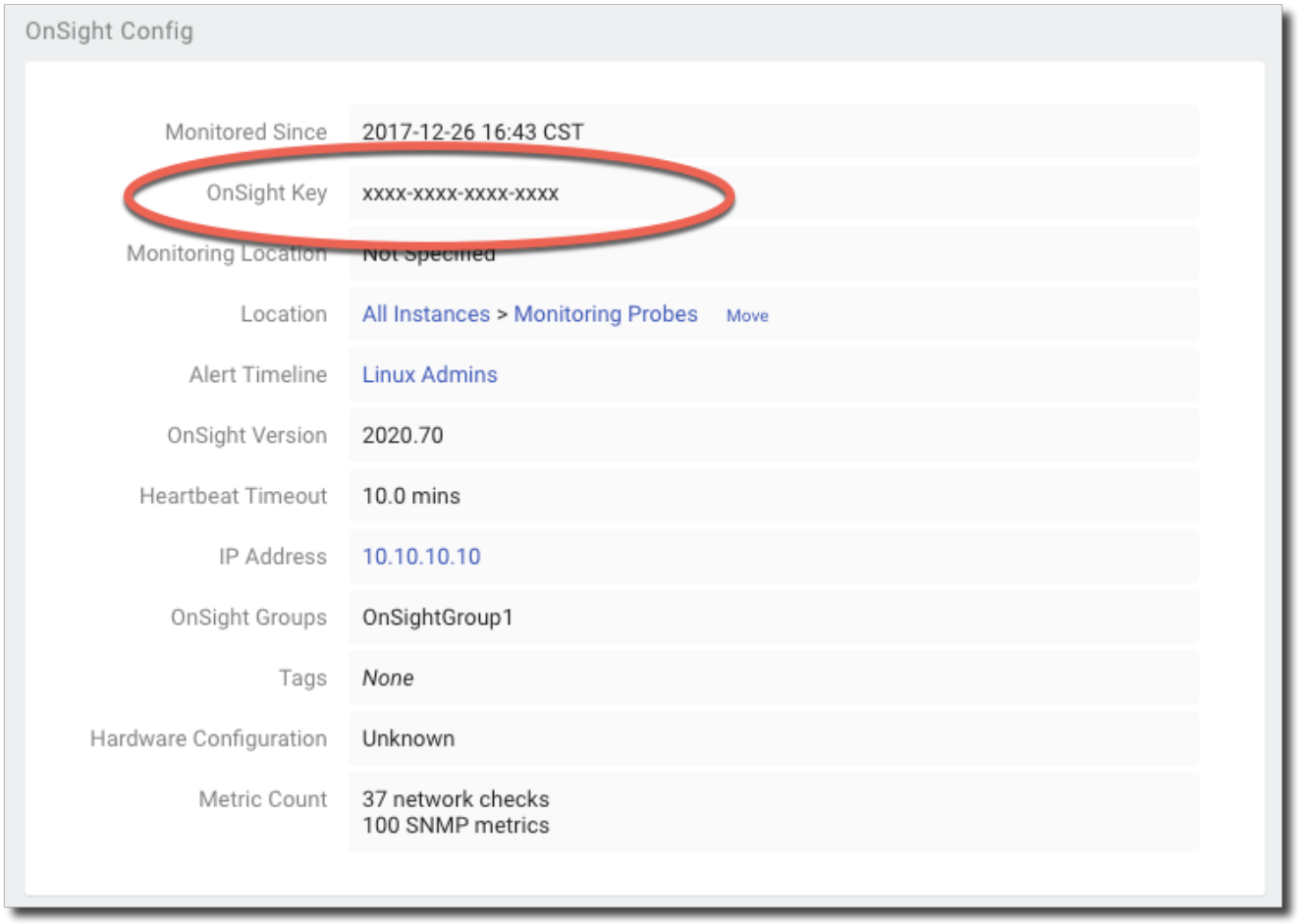
4. Open a browser and go to **http://<ONSIGHT-IP>**. Where **<ONSIGHT-IP>** is the IP address of your OnSight. Go to the OnSight's details page to obtain the IP address.



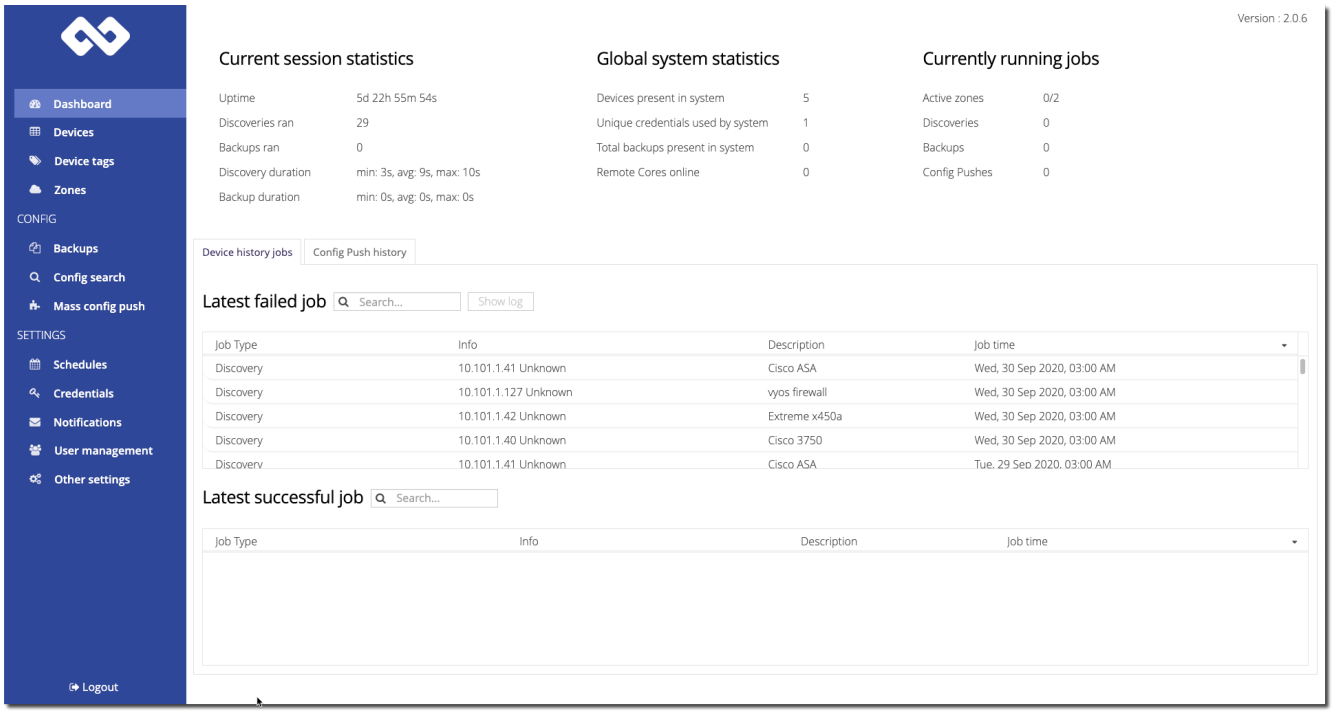
Your browser may prompt a certificate warning when accessing the NCM. To continue, you must accept the self-signed certificate. You can replace the certificate with your own later on.

5. Log in using the following credentials:

- **Username**: `fortimonitor`
- **Password**: The OnSight's appliance key. To obtain the key, go to the OnSight's details page or run the `onsight status` command from the command line.



After logging in, you will see the NCM UI.



Create a new credential

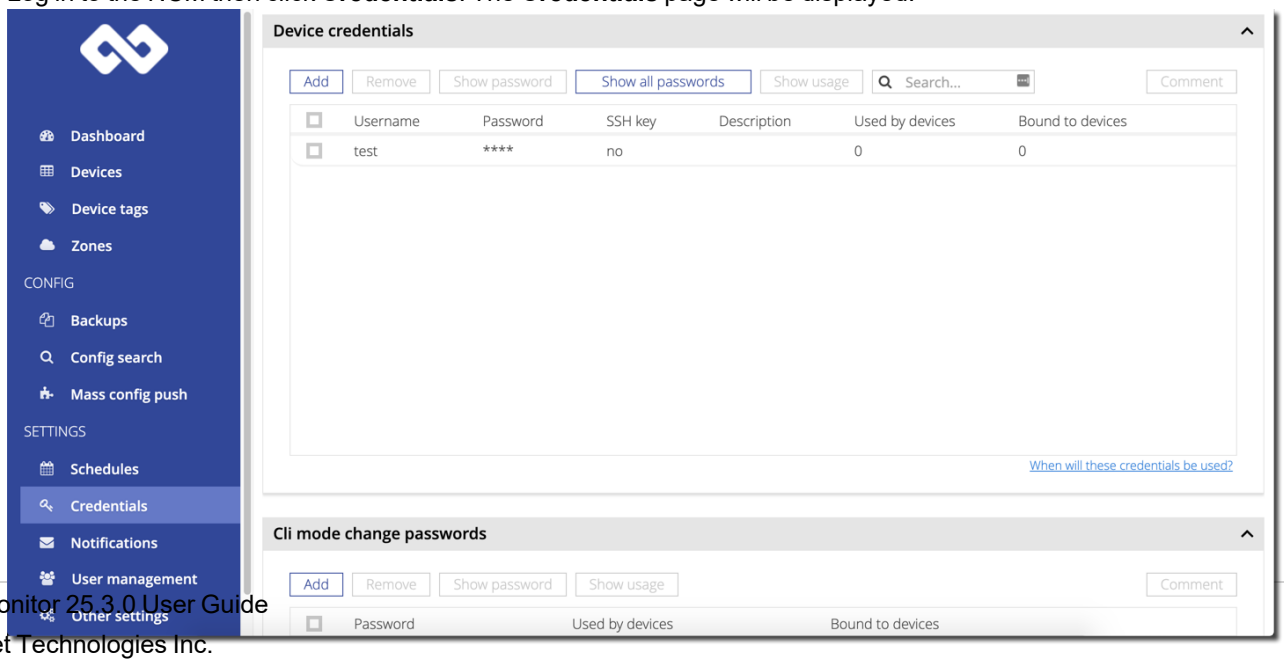
Once logged in, you need to set up the proper SSH credentials to provide the NCM modules access to the network devices. The credentials can be:

- The username/password used to login to the network device.
- An SSH private key which you can later upload to the NCM. The network device must already have the public key from the SSH key pair installed.

Credentials are stored encrypted on the NCM and are not synced up to the FortiMonitor control panel.

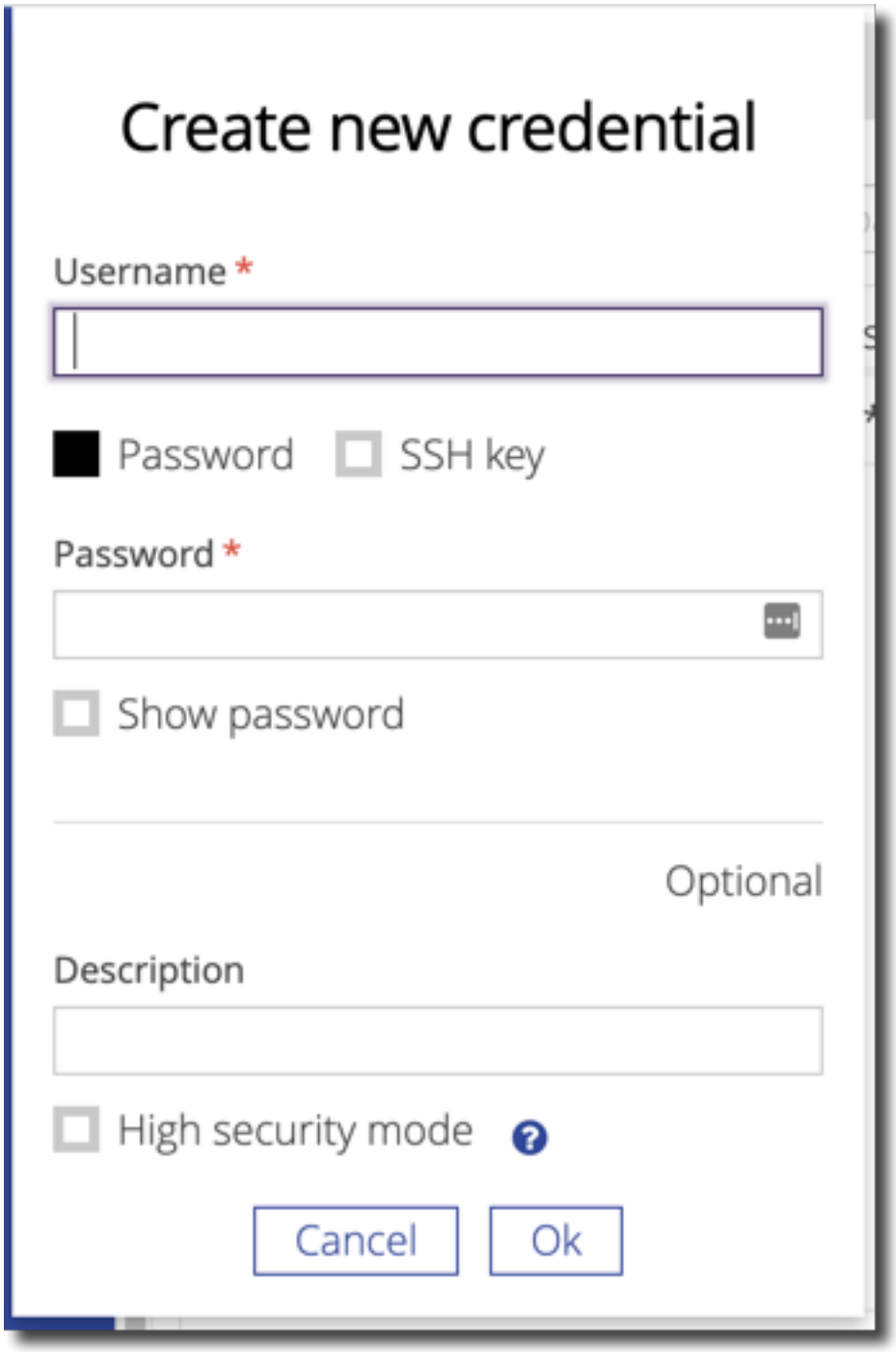
To create a credential, do the following:

Log in to the NCM then click **Credentials**. The **Credentials** page will be displayed.

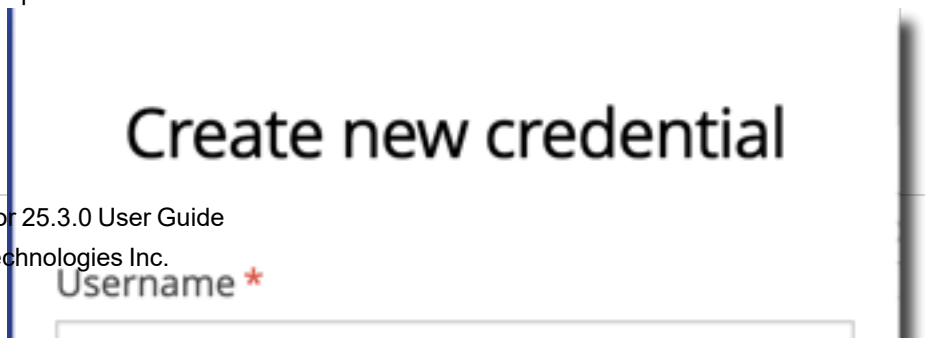


2. Click **Add**. You can use either a username/password combination or an SSH key.

   To use a username/password combination, select the **Password** option then enter a **Username** and a **Password**.

   - 

   To upload an SSH key, select the **SSH key** option then click **Choose File**. Locate the SSH key that you want to upload.

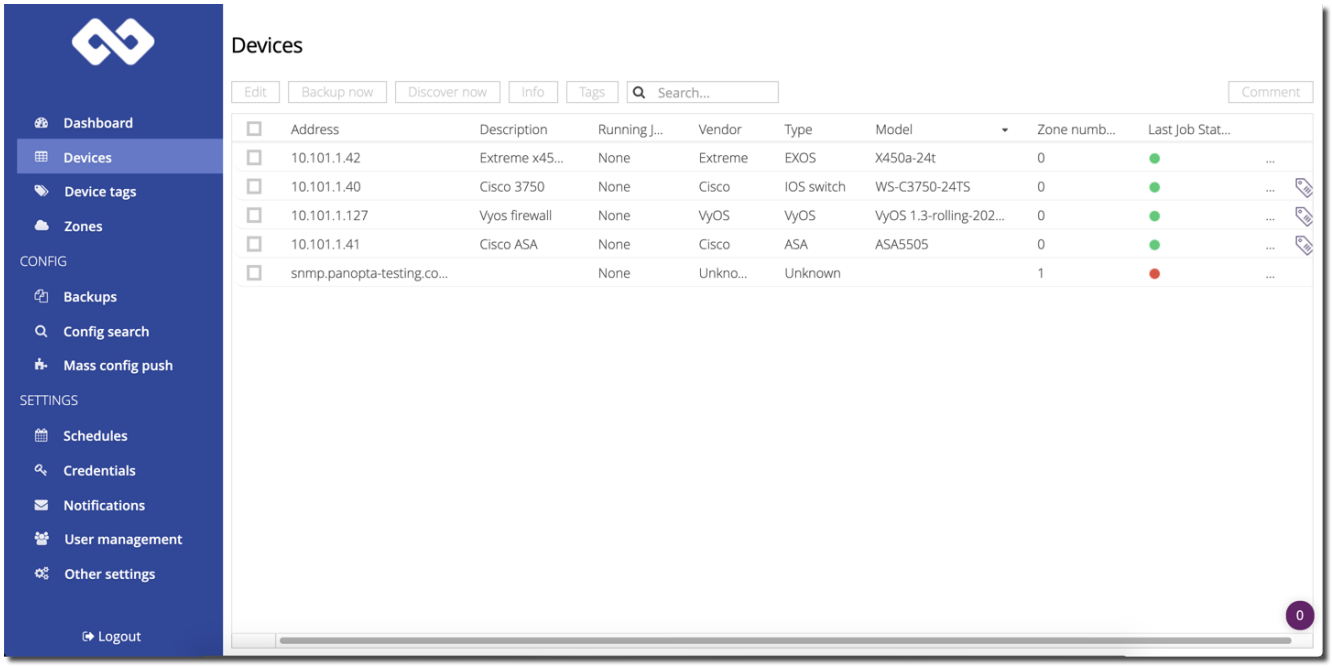
3. Enable the **High security mode** option. Enabling this option will prevent passwords and SSH keys from being displayed as cleartext anywhere in the NCM.
4. Click **Ok**.

Discovery

After adding a credential, discovery will automatically start. All network devices that are being monitored by the OnSight appliance in the group will be pulled in upon startup, and then updated every 30 minutes.

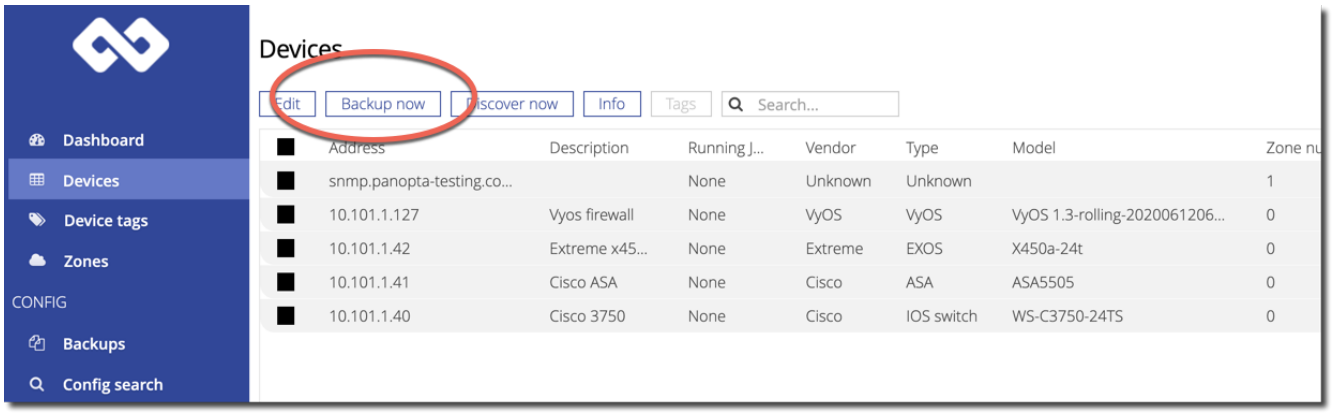
Go to the **Devices** configuration and ensure that:

- All devices you expect to be on the list are there.
- The devices listed on the page should match the list of devices monitored from that OnSight.
- The **Last job status** of all the devices in the table should be green. A green status indicates that discovery for each network device was successful.

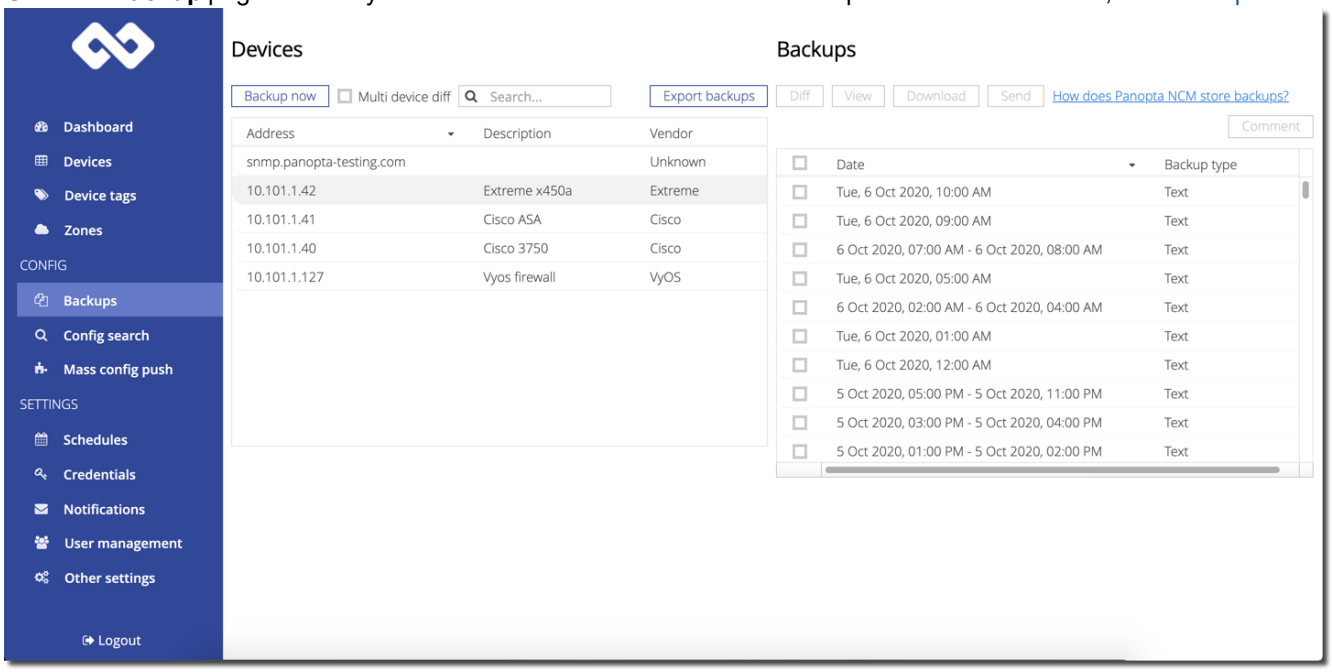


Backup

Make sure that you back up all of the devices that have been discovered by the OnSight. To do this, select all of the devices then click **Backup now**.



Go to the **Backup** page and verify that all of the devices have been backed up. For more information, see Backup.



Uninstall the NCM

To uninstall NCM, run the following command:

```
onsight ncm-uninstall
```

The command tears down all NCM-related containers and removes NCM completely.

NCM Supported Devices

3Com 29XX switch series

3Com non-switch devices (chassis, routers, etc.)

- 3Com switches (Comware)
- 6WIND Turbo Router
- 6Wind Virtual Service Router (VSR)
- A10 Networks AX series
- A10 Networks Thunder series
- A10 Networks vThunder series
- Accedian AMO series
- Accedian Metronids NID
- Adit 600 series
- Adtran NetVanta
- Adtran Total Access
- Adtran TA (Total Access) 2nd gen.
- ADVA LX series console servers
- ADVA MasterOS switches
- ADVA OptiDriver Network Manager
- Alcatel Lucent TiMOS (SR, ISAM and SAS)
- Alcatel OmniSwitch switches
- Allied Telesis AlliedWare
- Alloy switches
- Araknis switches
- Arris C4 series chassis
- Arista switches
- Arista vEOS
- Array Networks Load Balancers (ArrayOS)
- Arris E6000 CER
- Aruba ArubaOS switches
- Aruba ArubaOS-CX devices (Aruba / HPE 8320)
- Aruba IAP (Aruba Instant)
- Aruba Mobility Access Switch
- Aruba Mobility Conductor
- Aruba Mobility Controller
- Aruba Wireless Controllers
- AudioCodes Mediant
- AudioCodes MP-11x

- AudioCodes MP-12x
- Aviat Radios (WTM4200)
- Avaya ERS
- BDCOM OLTs
- Blonder Tongue CMTS
- Brocade FastIron
- Brocade FastIron Chassis
- Brocade FOS
- Brocade G620
- Brocade ICX switches
- Brocade MLXe
- Brocade NetIron
- Brocade NetIron Chassis
- Brocade NOS
- Brocade SLX-OS switches
- Brocade TurboIron
- Calix B6
- Calix BLC
- Calix E7
- Calix OccamOS
- Cambium ePMP series
- Cambium cnMatrix switches
- Casa CMTS
- Ciena Service-aware OS (SAOS)
- Cisco ASA
- Cisco ASA FirePower
- Cisco ASA multi-context
- Cisco ASA TD
- Cisco CatOS
- Cisco CRS series
- Cisco Fire Linux OS (FirePower TDM)
- Cisco FirePOWER (FX-OS)
- Cisco FirePOWER for AWS / KVM / Azure / GCP
- Cisco FirePOWER for VMware

- Cisco IE (industrial ethernet) switches
- Cisco IOS switches
- Cisco IOS routers
- Cisco IOS XE
- Cisco IOS XR
- Cisco MDS switches
- Cisco NCS series
- Cisco Nexus
- Cisco Nexus FMB
- Cisco NXOS (generic NXOS support)
- Cisco RF-Gateway
- Cisco SD-WAN vEdge / vSmart
- Cisco SMB SF2xx / SG2xx / SX2xx switch series
- Cisco SMB switches (SF300, SG300, SF500, SG500, etc.)
- Cisco WLC (wireless controllers)
- Citrix NetScaler
- CheckPoint Gaia devices
- CheckPoint Security Gateway
- CheckPoint Security Management Server
- CheckPoint SMB Gateway
- CheckPoint VSX
- CTS switches (FOS-3128, etc.)
- Cumulus Linux
- D-Link / Dlink 1510 switch series
- D-Link / Dlink 3200 switch series
- D-Link / Dlink 3528 switch series
- D-Link / Dlink 3552 switch series
- D-Link / Dlink 3620 switch series
- D-Link / Dlink DXS 5000 series
- D-Link / Dlink xStack switches
- D-Link / Dlink SmartPro switches
- Dasan OLTs
- Datacom DmOS devices
- Datacom DmSwitch devices

- Datacom routers
- Datacom switches
- DC YunKe Networks switches
- DCN switches
- DD-WRT
- Dell EMC OS10
- Dell Force10 (RTOS)
- Dell FTOS switches / chassis
- Dell PowerConnect switches (not all series supported yet)
- Dell PowerConnect 8024 (and similar PowerConnect switches)
- Dell R1 switches
- Dell Networking N-series switches
- Dell Networking switches (RTOS)
- Dell SmartFabric
- Dell switches (OS10)
- Dell VRTX devices
- Dell X series switches
- DragonWave Horizon Compact
- DragonWave Horizon Compact+
- Draytek Vigor (Discovery and Config Push only, Backup not supported)
- DrayTek VigorSwitch
- EdgeCore switches
- Eltex ME routers
- Eltex MES switches
- Eltex ESR gateways
- Engage IPTube
- Entrasys switches (A4 / B2 series)
- Exablaze FMB
- Exablaze Fusion switches
- Exalt ExtendAir
- Exinda devices
- Extreme 200 series
- Extreme ERS
- Extreme EXOS

- Extreme Extremeware
- Extreme MLXe
- Extreme SLX-OS switches
- Extreme VOSS / VSP OS
- Extreme VX wireless controllers
- Extreme WiNG AP
- Extreme WiNG Virtual Controller
- Extreme WLC
- F5 BIG-IP
- F5 BIG-IQ
- Fiberhome devices
- Fiberstore (FS.com) switches
- Fiberstore (FS.com) Campus switches
- FibroLan µFalcon
- FireBrick devices
- Fortinet FortiAnalyzer
- Fortinet FortiAuthenticator
- Fortinet FortiExtender
- Fortinet FortiGate
- Fortinet FortiOS (FortiADC, FortiWiFi, FortiMail, etc.)
- Fortinet FortiWeb
- Fortinet FortiWLC
- Foundry FastIron
- Foundry FastIron Chassis
- Foundry TurboIron
- Foundry TurboIron Chassis
- FreeNAS
- GCOM switches
- H3C Comware switches
- Halon SecurityRouter
- Harmonics CMTS
- Hatteras / Overture Networks
- HP Blade Switch (BladeCenter)
- HP BladeSystem OA

- HP Comware routers
- HP Comware switches
- HP MSA Storage
- HP MSM Controller
- HP ProCurve switches
- HP ProCurve stack
- HP v1910 switch series
- HP 1920 switch series
- HP 1950 switch series
- HPE 1920S switch series
- HPE StoreFabric devices
- HPE VirtualConnect
- Huawei Eudemon
- Huawei Quidway switches
- Huawei routers (H3C / Comware)
- Huawei SmartAX (Integrated Access Software OLTs)
- Huawei switches (H3C / Comware)
- Huawei USG
- Huawei VRP (including HRP, multi-context, etc.)
- IBM Flex System Fabric
- IBM RackSwitch switches
- IgniteNet FusionSwitch
- IgniteNet MetroLinq
- InfiNet WANFleX devices
- InfoBlox
- Juniper JUNOS
- Juniper JunOS EVO
- Juniper SSG
- LANCOM routers
- LANCOM switches (Discovery and Config Push only, Backup not supported)
- Lenovo FlexSwitch
- Lenovo Flex System Fabric
- Lenovo RackSwitch switches
- Lenovo ThinkSystems RackSwitch switches

- Meinberg LANTIME
- Mellanox InfiniBand switches
- Mellanox switches
- Metamako devices
- Metaswitch Perimeta SBCs
- MicroSens Switches
- MikroTik RouterOS (this includes: RB, CCR, CRS, x86, CHR, etc.)
- MikroTik RouterOS v7
- Moxa switches
- MRV LX series console servers
- MRV MasterOS switches
- MRV OptiDriver Network Manager
- NetApp switches
- netElastic vBNG
- Netgate Appliances with pfSense Plus
- Netgear GSM switches
- Netgear ProSafe M-series switches (M4300, etc.)
- NetScreen (ScreenOS devices)
- Netonix switches
- Nokia OLTs (FX-8)
- Nokia SR OS (SR, ISAM and SAS)
- Nomadix AG
- Nomadix EG
- Nomadix NSE
- OcNOS (switches running OcNOS)
- Omnitron RuggedNet switches
- Opengear Console Servers
- Opengear Infrastructure Manager devices
- Opengear Resilience Gateway (ACM)
- OpenWRT
- OPNsense
- Palo Alto firewalls
- Palo Alto Panorama
- Patton / Inalp SmartNode

- Perle IOLAN
- pfSense
- pfSense Plus
- Planet DSLAMs
- Planet switches
- Planet XGS switches
- Positron GAM devices
- Pulse Secure Virtual Traffic Manager
- Quanta switches
- RAD Carrier Ethernet Switches
- Ribbon (ECI) Apollo
- Riverbed RiOS
- Ruckus FastIron
- Ruckus FastIron Chassis
- Ruckus ICX switches
- Ruckus NetIron Chassis
- Ruckus standalone APs
- Ruckus TurboIron
- Ruckus Unleashed
- Ruckus Wireless Bridges
- Ruckus ZoneDirector
- SAF Tehnika Lumina
- Securepoint UTM
- Siklu EtherHaul
- Siklu EtherHaul Linux
- Siklu Terragraph
- SNR (NAG) Switches
- SonicWall devices (SonicOS)
- Sophos switches
- Sophos UTMs (Astaro)
- Telco Systems T-Marc devices
- TP-Link switches (JetStream series and Smart Switch series switches)
- Transition Networks NID
- TrueNAS

Turris OS devices (Turris Omnia, etc.)

Ubiquiti airFiber

Ubiquiti airFiber 60

Ubiquiti airMax

Ubiquiti Dream Machine (UDM)

Ubiquiti EdgePoint

Ubiquiti EdgePower

Ubiquiti EdgeRouter

Ubiquiti EdgeSwitch

Ubiquiti EdgeSwitch 10X

Ubiquiti GigaBeam

Ubiquiti LTU / LTU-Pro

Ubiquiti TOUGHSwitch

Ubiquiti UFiber OLT

Ubiquiti UniFi APs

Ubiquiti UniFi switches

Viptela vEdge / vSmart

Vyatta

VyOS

WatchGuard (Fireware OS)

YunKe switches

Zhone MXK

ZTE ZXA / ZXAN

ZTE ZXR10

Zyxel switches (GS19xx series)

Zyxel switches (GS22xx series and similar)

Zyxel ATP

ZyXel IES (MSC)

Zyxel MGS switches

Zyxel USG

Zyxel VPN

Zyxel XGS

ZyXel ZyWALL

Backup

Basics

The FortiMonitor Network Configuration Management (NCM) performs backups as per the schedule configured for the device.

The device needs to first be discovered (see the Discovery article) before it can be backed-up.
FortiMonitor NCM will use the discovered connector (protocol/port combination) and the discovered credentials to connect to the device.

After a successful connection, config of the device is retried, and stored.

Automatic exec-mode switching

The NCM can handle logging in to the device in user-exec or privileged-exec mode.
(if the particular device has concepts of **user-exec**/ **privileged-exec**)

If the device needs to be switched into **privileged-exec**, the NCM will do so.
(the **enable** password was already discovered during Discovery)

Startup vs. running config

The NCM will retrieve the configuration from the device using the appropriate commands for each vendor automatically.

If the device has concepts of **startup-config**/ **running-config**, the NCM always works with the **running-config** of the device.
If the device does not have concepts of **startup-config** vs. **running-config**, the NCM will simply retrieve a backup.

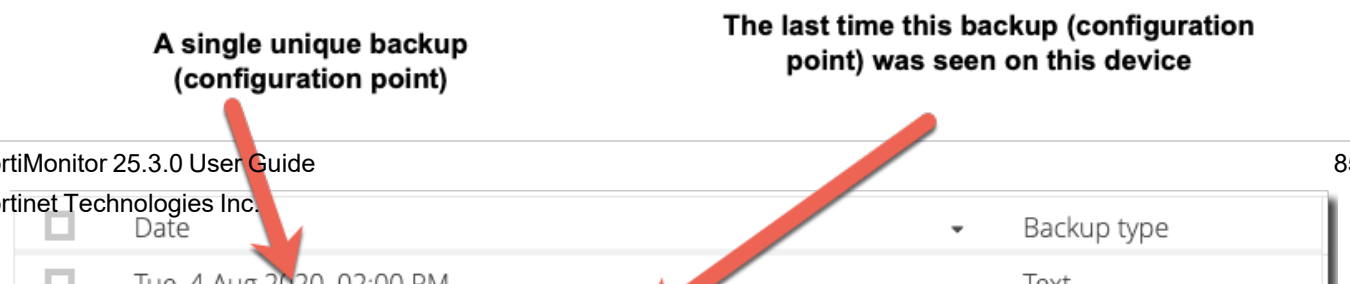
Only unique backups (when config changes) are stored

The NCM automatically stores unique backups of the config.
If a backup retrieved from the device is the same as the last backup already in the system, it will not be stored.

The backups table will show a time range for which a backup was valid for, rather than just backup retrieval timestamp.

Dynamic data (such as timestamps, etc.) are automatically filtered from backups before NCM compares backups to analyze if they have changed.
(this is supported for all our supported vendor drivers)

This feature makes it much easier to see when a device config changes.



Backup filters

As documented in the Backup article, if there are no changes to the config of your device, a new "backup" point is not required. In effect, this means that rather than showing you individual backups, NCM will show you configuration points - ranges of valid configurations on your device to build a configuration timeline of the device. To be able to do this, NCM needs to be able to figure out if there was a config change on the device. For example, if the configuration of the device contains a timestamp (Cisco IOS will show the current timestamp of when "show running-config" was executed), this needs to be ignored so this "change" in the configuration contents doesn't create a new configuration point. NCM contains many built-in filters for such dynamic data (as mentioned in the Backup article), but you can also create your own custom filters for backup contents.

Filtering order:

- "Deleted data" filters
- Built-in "dynamic content filters"
- "Ignored data" filters

Ignored data filters

The "Ignored filters" can be used to ignore changes to parts of the backup - during comparison of the new backup to the currently stored backup, NCM will IGNORE the matched data for the comparison purposes. The data will still be stored as a part of the backup, but if changes in the data happen, those changes will be ignored.

Deleted data filters

The "Deleted data" filters can be used to completely remove parts of the backup - before comparison of the new backup to the currently stored backup, NCM will REMOVE the matched data from the "new" backup. This means that this data will not be stored as a part of the backup at all - as if this data was not received from the device at all.

Filtering types

Line filters

For the line filters ("Line starts with" and "Line ends with"), if there is a match, the whole line is filtered / deleted.

These filters can be used for simple and easy matching rules if you know the exact beginning or ending of a line you want to filter / delete.

Regex filters

If there are no capture groups in the regex, the whole regex match is filtered / deleted. However if you use capture groups, only the content captured by the capture groups will be filtered / deleted.

This allows for very precise and flexible matching using regex and capture groups, so you can filter / delete only exactly what you need.

Search syntax help

Regex search examples

Find any vlan with IDs 1020 or 1030:

```
vlan (1020|1030)
```

Find any vlan in the 10xx range:

```
vlan 10\d{2}
```

Find any router with OSPF router-id in 10.0.0.0/24

```
router-id=10.0.0.\d+
```

Make search case insensitive (will find 'VLAN 1002' or 'vlan 1002')

```
(?i)vlan 1002
```

Regex cheat-sheet

Token	Description
.	any single character
\d	any number
\h	any horizontal white-space (space, tab, etc.)

Token	Description
?	one or zero times
*	any number of times (even zero - zero or more times)
+	at least once (one or more times)
{2}	Repeats 2 times

Token	Description
[abc]	any of the characters inside ('a' or 'b' or 'c')
[a-z]	any of the characters inside the range (a through z)
|	or
()	group - for example '(foo|bar)' - 'foo' or 'bar'

Modifier	Description
(?i)	Case insensitive

Mass Config Push

Basics

Mass Config Push is used to send a set of commands to a group (or all) of devices in your network. This allows you to roll out mass configuration changes across your network, or query command outputs from a large set of devices. Before you can push to devices, each device must be in a "discovered" state (normally done during device addition into the NCM). Config push will not be performed on undiscovered devices, and they will be grouped in the **Un-discovered devices** group.

To perform a push, you first need to create a **Push Preset** from the **Mass Config Push** page. After a preset is created, you can click **Run now** to push the provided commands to the selected devices.

Output Grouping

The commands you specify in the **Push Preset** are sent to the devices, one line at a time (for more information on how NCM interacts with the device, see How commands are sent to devices). The output of the executed commands are evaluated on all devices, and a new **Output Group** is created for each unique output during the push. If 2 devices reply with the same output, both devices are assigned to a single output group belonging to that unique output.

For example, if you push a command to 200 devices, but across all devices, there will only be 2 unique outputs to this command, the NCM will create only 2 output groups, and assign each device to the right output group. Output grouping means you don't need to inspect outputs of 200 devices after a config push - saving you time, and making navigation in large push outputs easy.

State of device before Push - require "enable" or "configure" CLI mode

You have the option of letting the NCM ensure that each device in the **Push Preset** is in the desired CLI mode before your specified commands are pushed. You can check **Require enable mode** and/or **Require configure mode** in the **Push Preset**, and the NCM will make sure the device is in the specified mode before executing your commands. Modes available on devices are discovered during Discovery, using mode credentials configured in the **Credentials** screen. You can check which modes the NCM was able to discover on devices using the **Devices** > **Info** page.

If a device can not be switched into the specified CLI mode before the push, **Config Push** will NOT push the provided commands to the device, and any devices failing the mode switch will be grouped in appropriate **mode not supported** output groups.

The **enabl**" and **configure** mode names are only symbolic - each device type has its own modes. In general, the **enable** mode means the 2nd highest possible CLI mode on the device, and the **configure** mode the 3rd highest possible CLI mode on the device (if the device type supports this). Taking Cisco as an example, the **user exec** mode would be 1st possible login mode, **privilege exec** would be the 2nd (enable) mode, and **configure mode** would be the 3rd (configure).

Please see our Device mode table article for more info on how the modes in the push preset map to various vendors/device types.

Command parsing

In a **Push Preset**, you provide a command set to the NCM, which will be parsed for submission modifiers and control sequences, and then sent to the device(s).

If you do not provide any submission modifiers or control sequences, the NCM will split the provided command set line by line, and then send it to the selected device(s) one by one.
Basically, you give the NCM a number of commands, and it will distribute them to your network.

For example, specifying push commands like this will work as expected, and the NCM will handle pagination ("-- more --") natively:

```
show version
```

```
show ip interface brief
```

```
show vlan brief
```

For details on exactly how commands are sent to devices, please see How commands are sent to devices.

Command sending modifiers and control sequences

NCM supports command submission modifiers and sending control sequences.

There are 2 command submission modifiers possible:

```
$[enter]
```

```
$[no-enter]
```

These modifiers have to be put at the end of the line, and control if <Enter> will be sent after the command line.
The default behavior is to send <Enter> after each command line, UNLESS "$[no-enter]" is specified, or the command is a control sequence.
(please check the Command parsing examples section for more details)

Normally, NCM waits for some know output (for example a prompt) after each sent command line (see How commands are sent to devices for more info).
There are 2 modifiers that can influence this behavior, with "$[wait]" being the default behavior.

```
$[wait]
```

```
$[no-wait]
```

Specifying "$[no-wait]" will cause NCM to instantly continue sending the next command line without waiting for any device outputs.
This can be useful when for example the device doesn't respond with any data after sending some command/line, and you want NCM to continue sending data to the device.
(please check the Command parsing examples section for more details)

The NCM supports sending various control sequences to the device, in this format:

```
$[0x01]
```

As mentioned above, <Enter> is not by default sent after a control sequence, but this can be forced by appending "$[enter]" to the end of the line.
Full list of command sequences available in this article: VT100 Control / Command sequences.

Both submission modifiers and command sequences support escaping by prefixing them with an additional "\":

```
\$[0x01]  = "$[0x01]" will be sent literally to the device(s), instead of a control sequence

\$[enter] = "$[enter]" will be sent literally to the device(s), instead of modifying command submission behavior
```

Please check the Command parsing examples section for more details on how modifiers and control sequences work.

How commands are sent to devices

Config Push has a few stages:

- connect to device, login to the CLI
- switch the device to desired CLI mode (if require **enable** or **configure** mode is set)
- send each command line from the **Commands** to the device (more info below)
- collect all output from device
- disconnect the CLI session
- create a new output group for this device OR assign the device into an existing output group

The main (and most important) portion of **Config Push** is sending the actual commands provided in the **Commands** set to the devices.

The NCM works as follows:

1. from the command set, send the next command line to the device, followed by **Enter**.
2. collect all output, up until a known (recognizable) output occurs (unless "$[no-wait]" specified)
3. **IF** "$[no-wait]" specified, **GO TO** step 1
4. **IF** device asks if it should continue with output (more info below), NCM answers "Y", **GO TO** step 2
5. **IF** device outputs a question (such as a "[y/n]" question - more info below), stop, **GO TO** step 1.
6. **IF** device asks for a password, or another "secret", stop, **GO TO** step 1
7. **IF** device pages through output ("-- more --"), NCM sends pagination key (space for most device types) to proceed to next page, **GO TO** step 2
8. **IF** device asks to confirm end of output, ("press Q to finish"), NCM sends the appropriate key to finish output, **GO TO** step 2
9. **IF** a prompt is received, stop, **GO TO** step 1

As per the description above, pagination ("-- more --"), and output termination which some devices require ("Output finished, press Q to continue") will be handled completely automatically.

For **step 3**, NCM will automatically answer "Y" to questions like "Show all items?" or "Do you want to show sensitive items?" or "Confirm command...", etc.
For **step 4**, NCM recognizes any generic questions asked by devices that end in the "[y/n]" or "(yes/no)" prompts, but also things like "Press enter to confirm change", etc.
For **step 5**, usual "Password:" or "Password for..." and similar password prompt are recognized

To make this simple - you can provide the same sequence of commands and key-strokes to the NCM as you would when interacting with the device CLI, and the NCM will properly interact with the device as expected.

Command parsing examples

Push commands:

```
write memory

y
```

Device output (keys sent by NCM in green):

device# write memory<enter>
This may take some time, continue? [y/n] y<enter>
.....
device#

For the "y" confirmation, it will be sent as "y" and then Enter. Here is an example of how to send a control sequence, and send the "y" without enter:

commit

```
$[0x1A]

y$[no-enter]
```

Device output (keys sent by NCM in green):

```
device# commit<enter>
Press CTRL + Z to continue, or Q to quit... <CTRL+Z>
Are you sure you want to contiue? [y/n] y
..... done
device#
```

Sometimes, you want the NCM to send commands to the device even if the device doesn't output any known outputs after sending a command line.
For example, many devices don't respond with anything if you need to provide a text block. You can avoid "INTERACTION_ERROR" in this case like this:

```
banner motd ^

This is the start of the banner$[no-wait]

And this is a 2nd line of the banner$[no-wait]

This is the end of the banner^
```

It is important to NOT specify "$[no-wait]" after the last line, since the device will respond with a prompt after the last line, and you want the NCM to wait for the prompt and collect the output.

The interaction with the device will look like this:

```
device (config)# banner motd ^<enter>
Enter TEXT message.  End with the character '^'.
This is the start of the banner<enter>
And this is a 2nd line of the banner<enter>
This is the end of the banner^<enter>
device (config)#
```

Debugging failed pushes - "INTERACTION_ERROR" group

If there is an "INTERACTION_ERROR" output group in your Push results, this means NCM was not able to identify any known/expected output from some devices (see How commands are sent to devices). If this happens, NCM will stop pushing to the device(s) for safety reasons, and place these devices in the "INTERACTION_ERROR" group.

The easiest way to find out why a push is failing is to enable Device Output Logging in "Zones > Your_zone > Debug Mode > Device output logging". After this is enabled, re-run your push and wait for it to fail.
After the push fails, download the Device Output Log file, and check what exactly in the device communication caused NCM to fail.

You can also create a Support Ticket on our Portal and attach the log file - and we will be happy to investigate why the push failed.

Device mode table

This table shows some examples of how NCM treats **enable** and **configure** mode equivalents for various device types. Please note not all device types supported by NCM are present in this table - we hope it should be clear from these examples how **enable** /**configure** mode switching behaves.

This table is most useful when you want NCM to bring a device to a certain CLI mode in **Mass Config Push** before your commands are executed.
Please see this wiki article State of device before Push - require "enable" or "configure" CLI mode for more details on setting the CLI mode before a Config Push.

Device type	1st mode - "base" mode (user-exec equivalent)	Command	2nd mode - "enable" mode (privileged-exec equivalent)	Command	3rd mode - "configure" mode (configure equivalent)
Cisco IOS	`device>(user-exec mode)`	`enable`	`device# (priv-exec mode)`	`configure terminal`	`device (config)# (configure mode)`
3com / HP / H3C / Huawei Comware OS	`<switch>(base mode)`	`system-view`	`[switch] (system mode)`	•	•
Arista	`device>`	`enable`	`device#`	`configure`	`device (config)#`
Aruba / HP ArubaOS / ProCurve / ProVision	`switch>`	`enable`	`switch#`	`configure`	`switch (config)#`
Cumulus Linux	`cumulus@cumulus:~$`	`sudo su`	`root@cumulus:/home/cumulus#`	•	•
F5 TMOS (BigIP, BigIQ, etc.)	`[root@f5:Act:Standalone] config #`	`tmsh`	`(cfg-sync Standalone)(Act)(/Common)(tmos)#`	•	•

HP V1910 / 1920	`<switch>(base mode)`	`_cmdline-mode on` `system-view`	`[switch]` (unlocked and in system mode)	•	•
Juniper JunOS	`root@device:RE:0%` (login to FreeBSD)	`cli`	`root@device>(base JunOS mode)`	`configure`	`root@device#` (JunOS configure mode)
OPNsense	`user@OPNsense:~ %`	`sudo su`	`root@OPNsense:~ #`	•	•
Palo Alto PAN-OS	`user@pa-fw(active)>`	`configure`	`user@pa-fw(active)#`	•	•
Ruckus SmartOS ZoneDirector / Unleashed	`ruckus>`	`enable` `force`	`ruckus#`	`config`	`ruckus (config)#`
Ubiquiti EdgeRouter	`ubnt@ubnt:~$`	`configure`	`ubnt@ubnt:~#`	•	•
Vyatta / VyOS	`vyatta@vyatta:~$`	`configure`	`vyatta@vyatta:~#`	•	•

VT100 Control / Command sequences

NCM Mass Config Push allows you to send various command sequences (VT100 control characters) to the device using the push.

Please use these variables in Config Push when sending control sequences:

Control sequence	NCM variable	Control sequence	NCM variable
CTRL + A	$[0x01]	CTRL + N	$[0x0E]
CTRL + B	$[0x02]	CTRL + O	$[0x0F]
CTRL + C	$[0x03]	CTRL + P	$[0x10]
CTRL + D	$[0x04]	CTRL + Q	$[0x11]
CTRL + E	$[0x05]	CTRL + R	$[0x12]
CTRL + F	$[0x06]	CTRL + S	$[0x13]
CTRL + G	$[0x07]	CTRL + T	$[0x14]
CTRL + H	$[0x08]	CTRL + U	$[0x15]
CTRL + I	$[0x09]	CTRL + V	$[0x16]
CTRL + J	$[0x0A]	CTRL + W	$[0x17]
CTRL + K	$[0x0B]	CTRL + X	$[0x18]
CTRL + L	$[0x0C]	CTRL + Y	$[0x19]
CTRL + M	$[0x0D]	CTRL + Z	$[0x1A]

Device variables

NCM supports declaring per-device variables, which can be used in Config Push for more automation flexibility.

Defining device variables

Device variables can be defined for device(s) in the **Devices** screen. You will find a **Variables** button here, which opens the variable edit window.

The Variable edit window has 2 separate modes:

- single device variable editing mode
- bulk (multi-device) variable editing mode

If you open the Variables window WITHOUT any device being currently selected, you will enter into the bulk edit mode. If you open the window when multiple devices are selected, you will enter the bulk edit mode with data pre-filled for devices which were selected. Finally, if a single device is selected, you will enter into single device variable edit.

Single device variable edit

In this mode, you can set variables for the specific device which was selected when you opened the variable edit window. You can simply specify variables in a "key=value" format. If any variables are already present on the device, they will be loaded into the edit window for you. Here is an example of how variable configuration on a single device would look like:

```
foo = bar
variable_name = variable_value
hostname = router
```

The above configuration would set 3 variables for the selected device, "foo", "variable_name" and "hostname". When saving variables, omitting any existing variables removes those variables. To simplify - when you save variables for a device, all existing variables will be removed, and then variables will be set to the device as provided in the edit window.

Bulk (multi-device) variable edit

In bulk mode, you need to specify for which device you wish to set variables. Each device is specified as a section, and after opening a section for a specific device, you can set its variables. You can then move on to another device by opening another section. Here is an example:

```
[device_address_1 @ Z1]
variable_name = variable_value
foo = bar

[192.168.1.1 @ Z2]
hostname = router
```

In the example above, we set variables for 2 devices:

- device "device_address_1" in a Zone with ZoneID "Z1"
- device "192.168.1.1" in a Zone ZoneID "Z2"

As mentioned in a previous section, if you open the variable edit window without any devices selected, it will be in an empty bulk-edit mode. You don't need to manually write out the device sections. You can select multiple devices in the Device table, and open the variable edit window to have the devices pre-filled for you, including their existing variables.

Please note that when saving devices in bulk edit mode, omitting variables for devices which already have variables present will remove those variables. To simplify - if you save variables for devices in bulk edit mode, all existing variables from each specified device will be removed, and then variables will be set to the devices as provided from the new configuration.

Variable formatting

The variable name, as well as the variable value will have both their leading and tailing white-space removed. All of these declarations as valid, and equal:

```
hostname=router1
hostname = router1
hostname= router1
```

All of the above will result in a variable with name "hostname" and value "router1" created. It is not possible to declare variables which contain leading or tailing spaces. Spaces inside both variable names and values are however fully supported.

Using variables in Config Push

Variables can be used in Config Push Commands to substitute unique per-device values when commands are being pushed to devices. This allows you to create a single Push Preset which can send unique commands to each device, or allow for unique per-device values to be inserted inside commands. Variables can be referred to in Config Push in the following format: "${variable_name}".

Here is an example which would utilize a variable with name "snmp_community" to create an SNMPv2 community on MikroTik RouterOS:

```
/snmp community
add security=none write-access=no name=${snmp_community}
```

Alternatively, here is an example of creating a user on Cisco IOS, where the password for the user is read from a device variable:

```
! simple user creation
username network-admin password ${admin-password}

! privileged user creation
username network-admin privilege 15 password ${admin-password}
```

Variable substitution and the "\" escape character

If you wish to send a sequence beginning with "${" to the device, you can escape the variable declaration like this:

```
/interface bridge
add name=lo1 comment="This is a comment with \${some strange characters}"
```

In the example above, NCM would not substitute the text above for a variable's value. The comment for the loopback interface would be set to "This is a comment with ${some strange characters}".

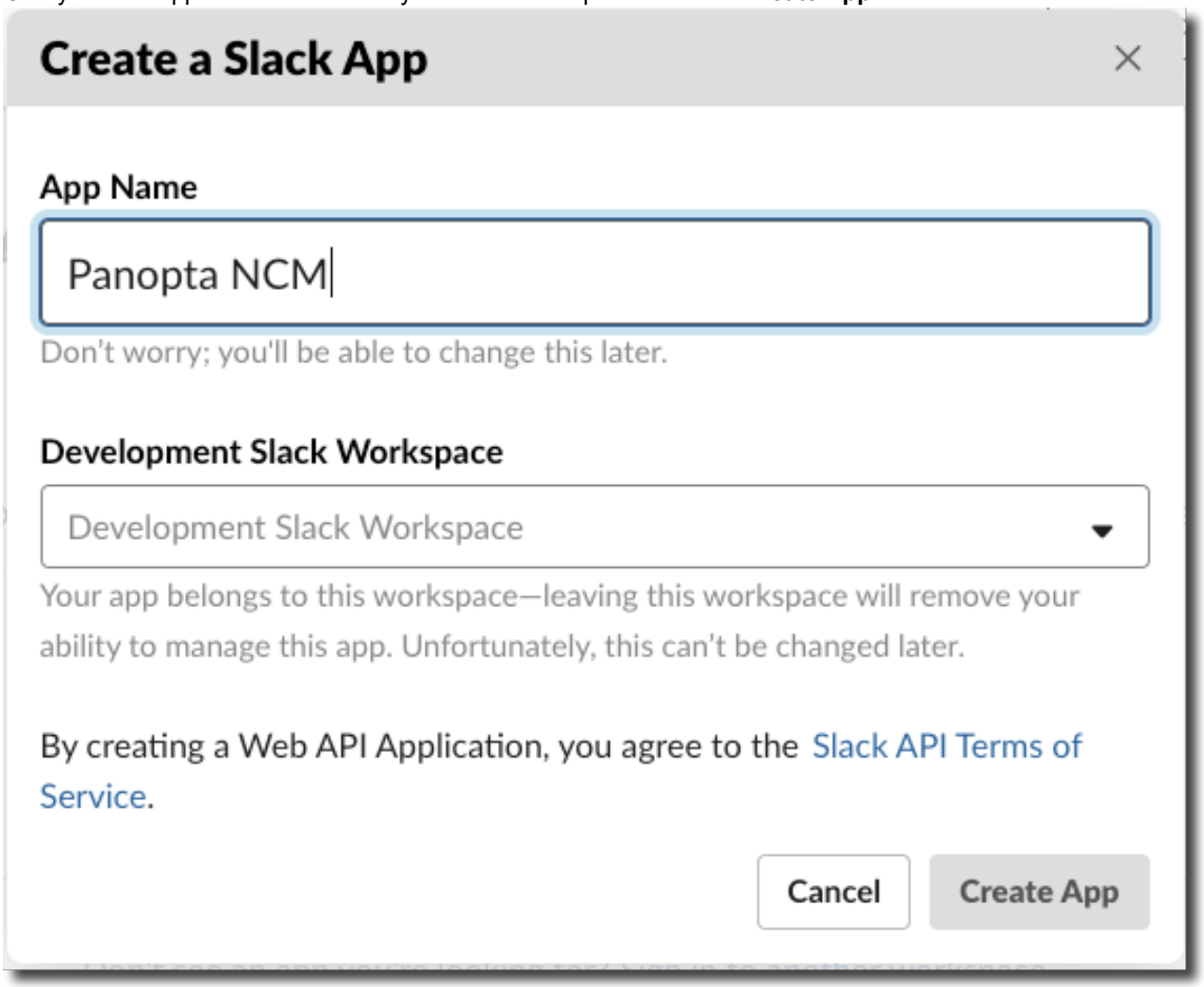
Variable chaining and Push Behavior Modifiers

Variables chaining is NOT supported. In other words, you can NOT refer to other variables from within different variables. Config Push Behavior Modifiers however CAN be declared inside variables.

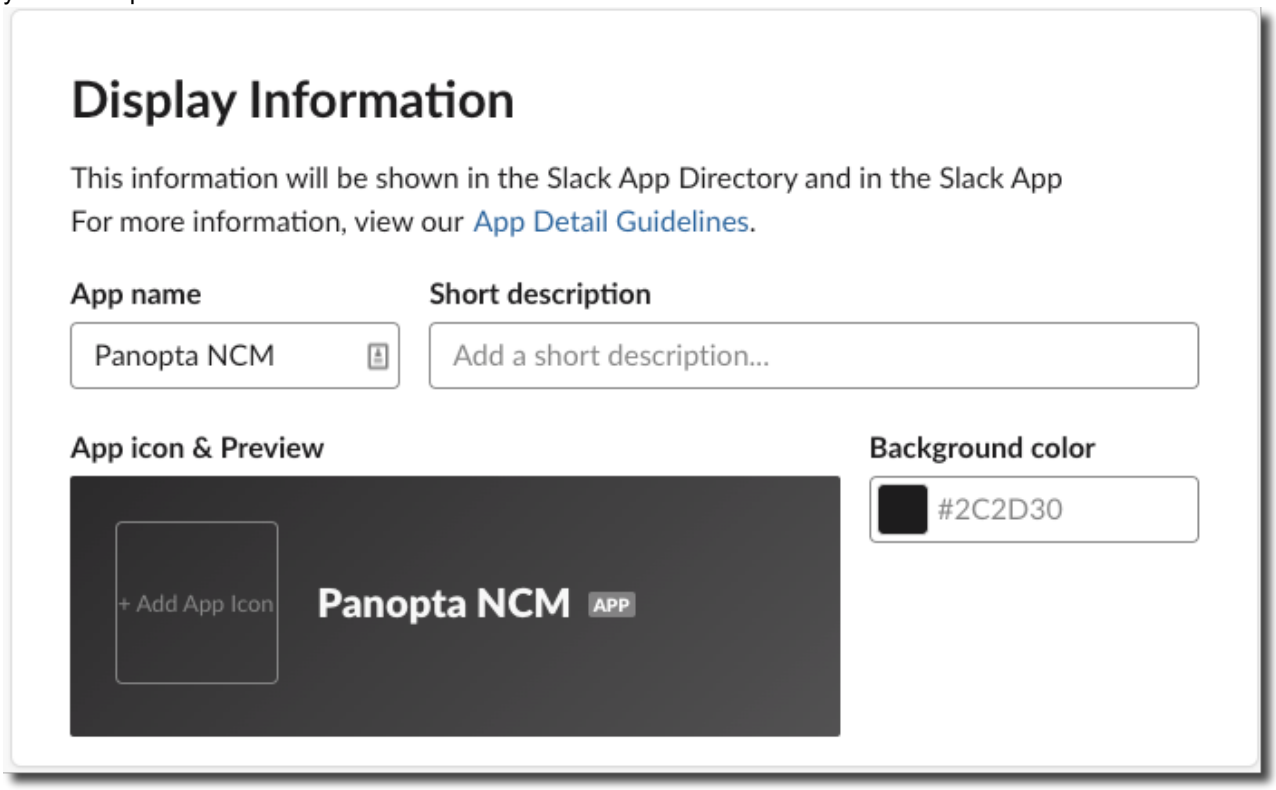
```
# this is NOT possible
var1=value
var2=some_other_${var1}
```

NCM Slack configuration

This article shows how to configure a Slack App for usage with the NCM.

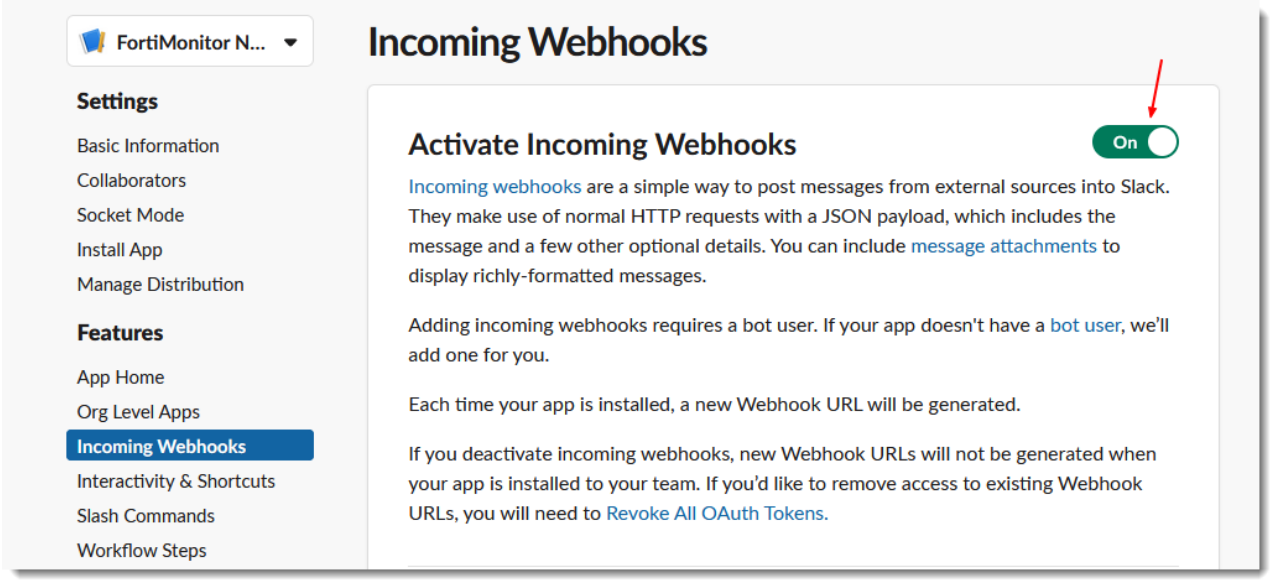
1. Open https://api.slack.com/apps and click **Create an App** to start:
   Give your new app a name and select your Slack Workspace then click **Create App**.
2. 

Scroll down to the **Basic Information** section. You can add the name and the logo that your Slack Bot will use in your Workspace:

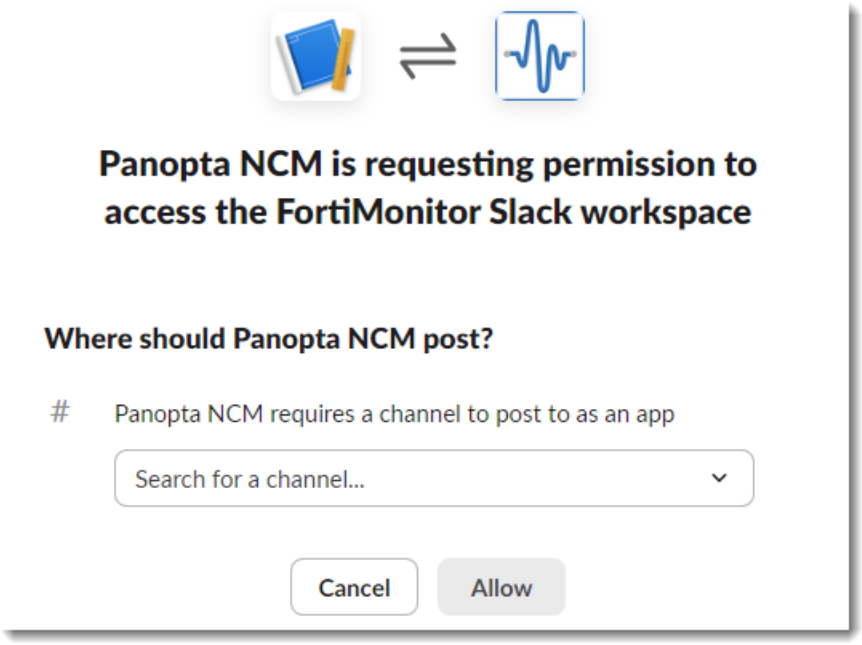


3.

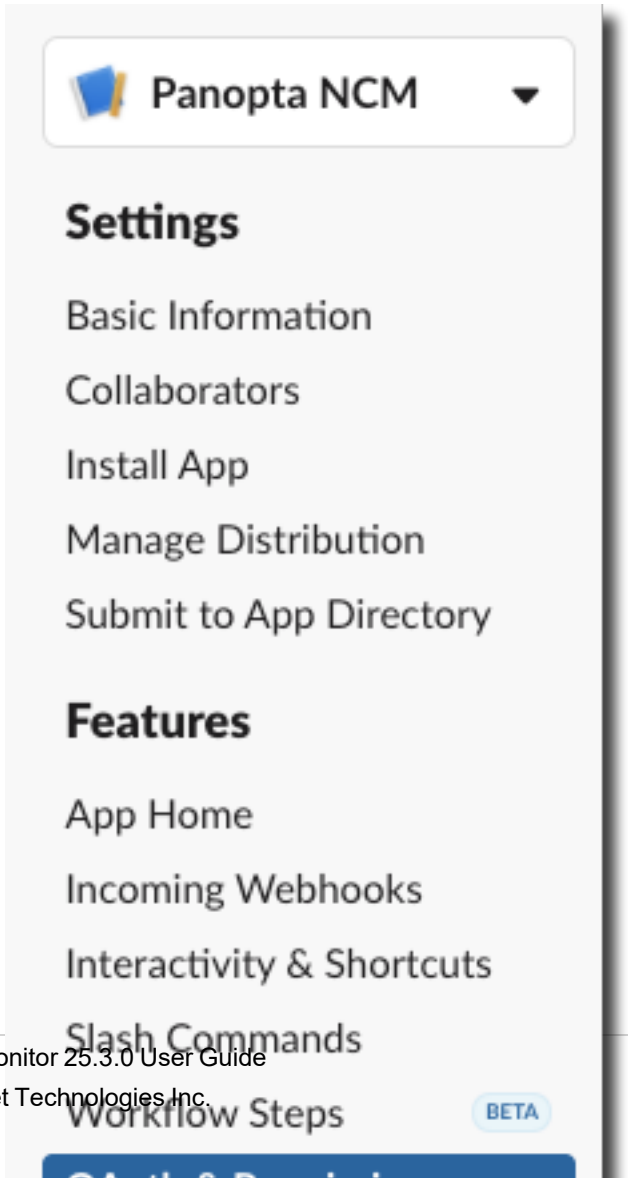
4. Enable incoming webhooks.



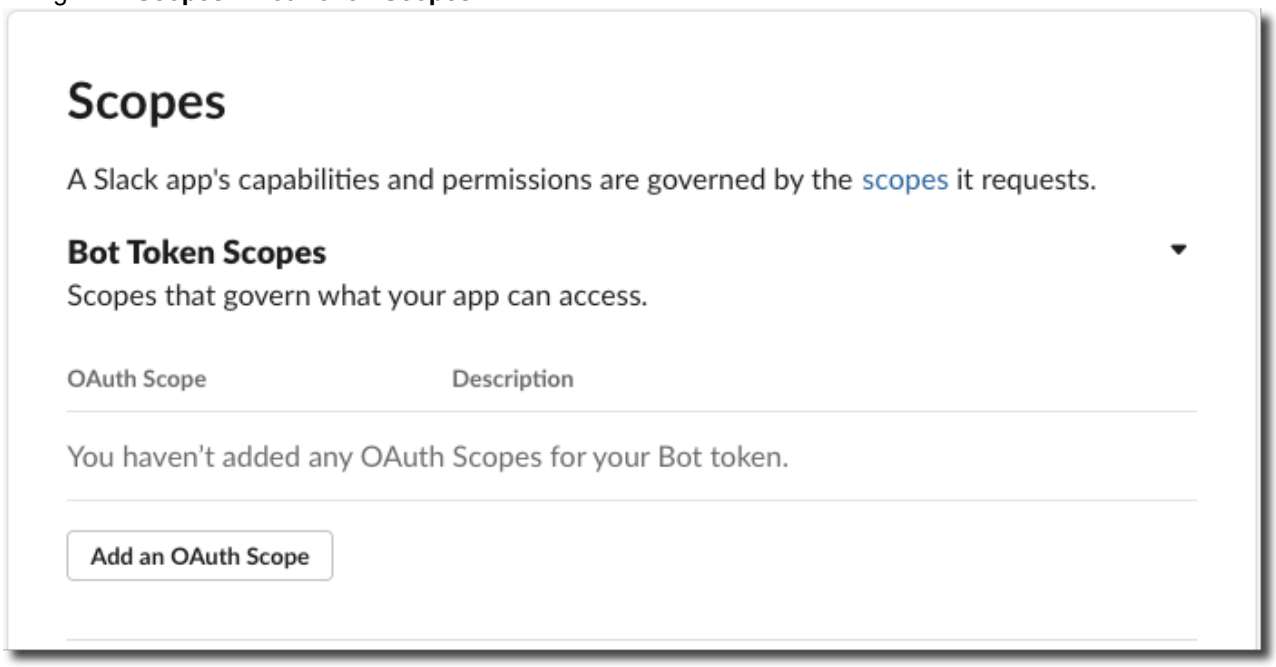
5. Select the channel where the app should be applied to.



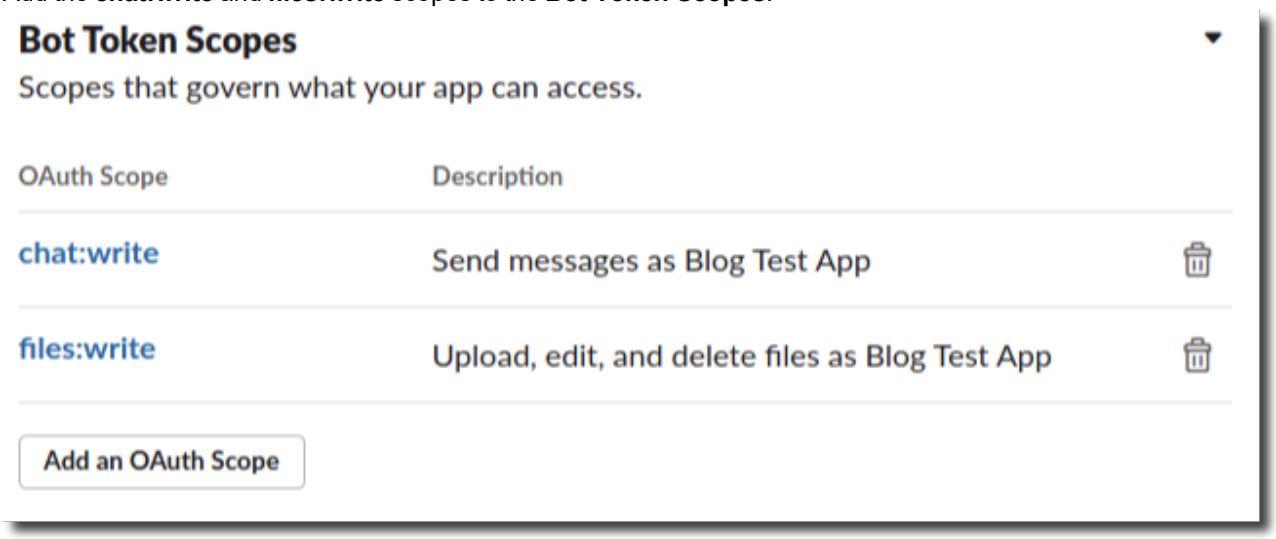
In the left menu, click **OAuth & Permissions**.



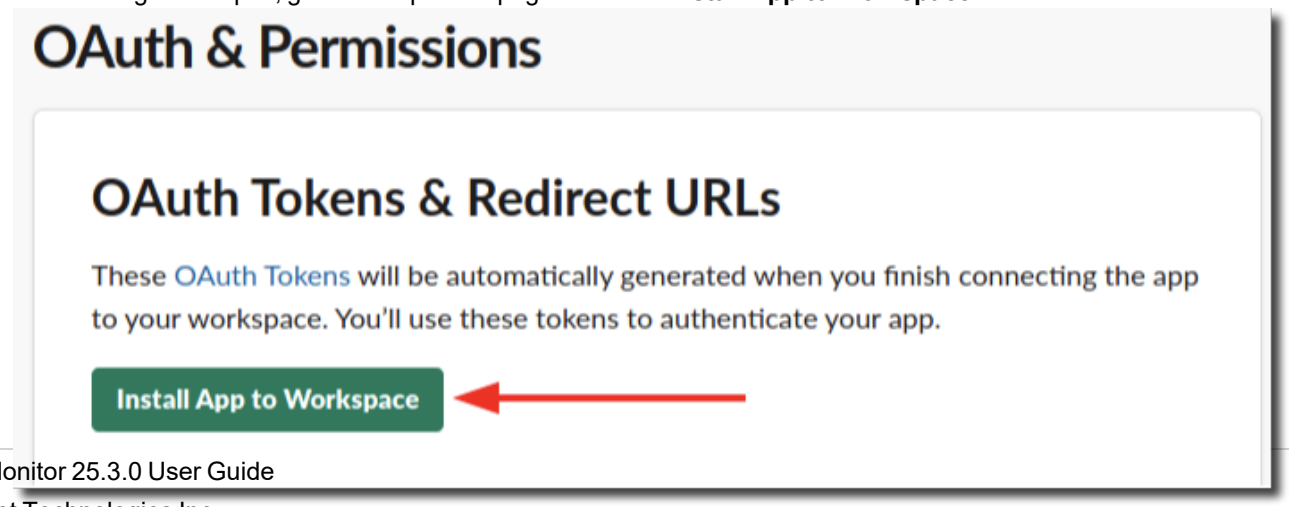
Navigate to **Scopes** > **Bot Token Scopes**.



7. Add the **chat:write** and **files:write** scopes to the **Bot Token Scopes**.

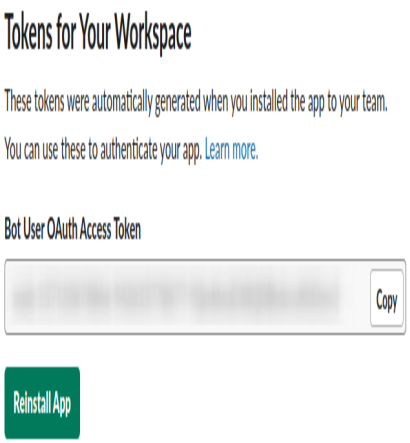


8. After adding the scopes, go to the top of the page then click **Install App to Workspace**.

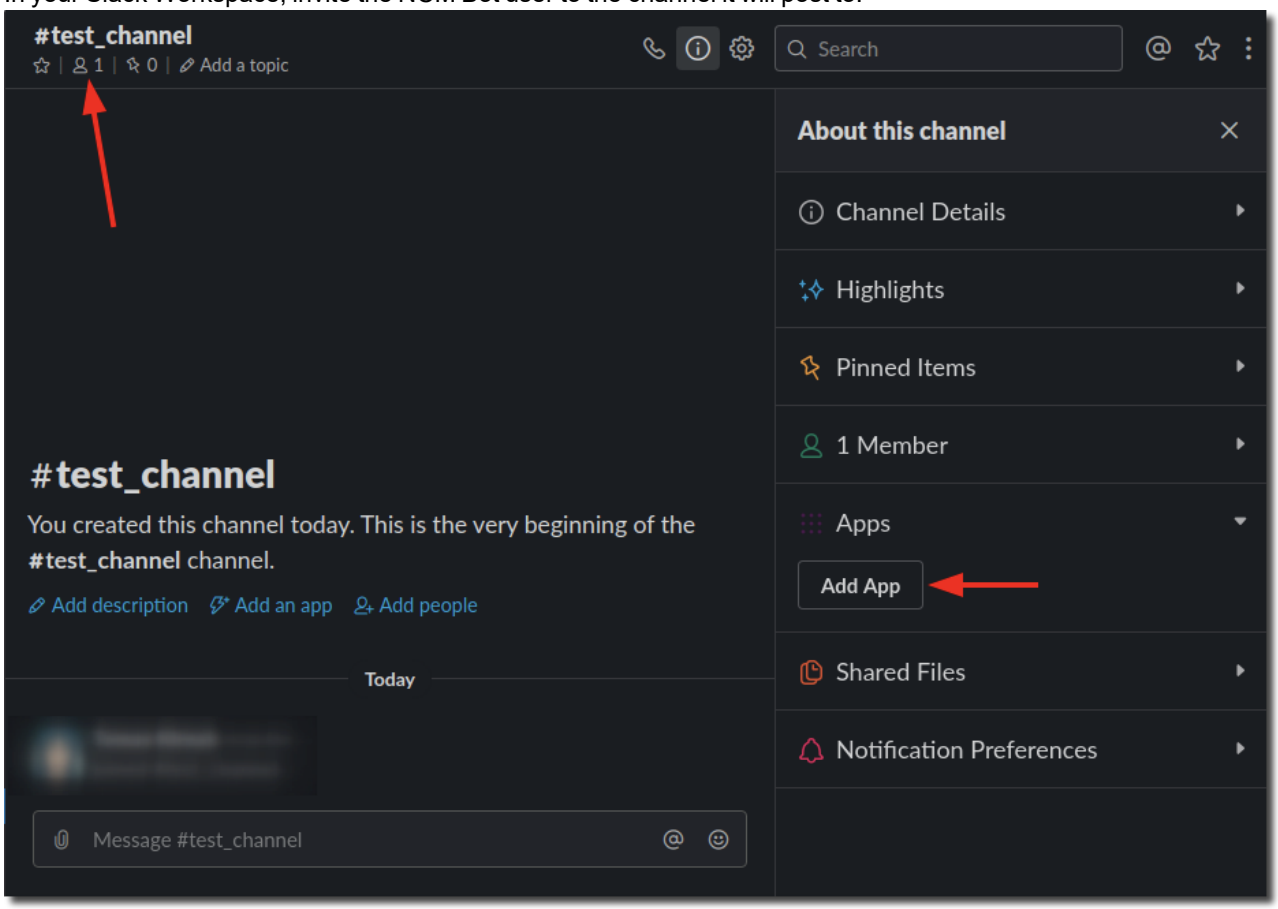


9.

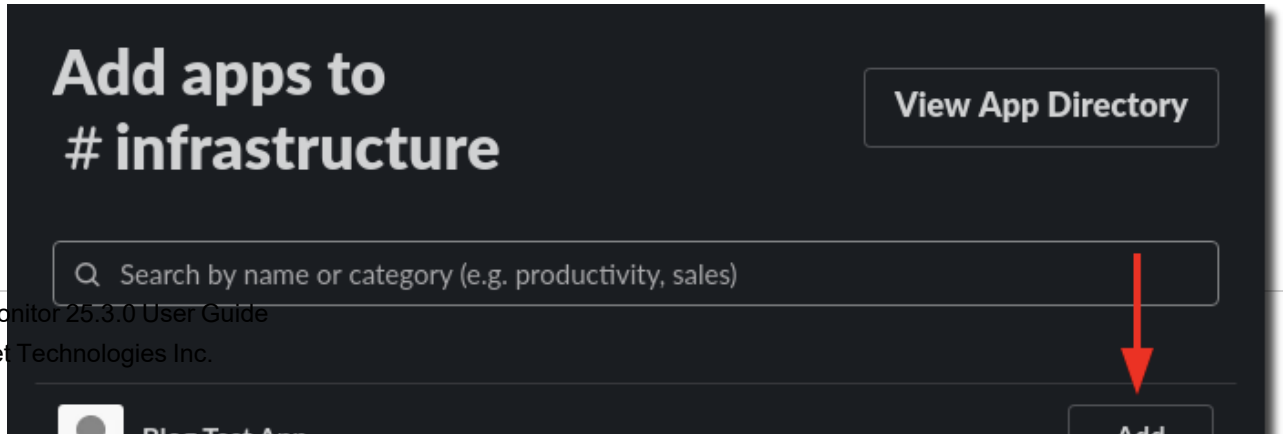
You will get a **Bot User OAuth Access Token**. Copy the token.



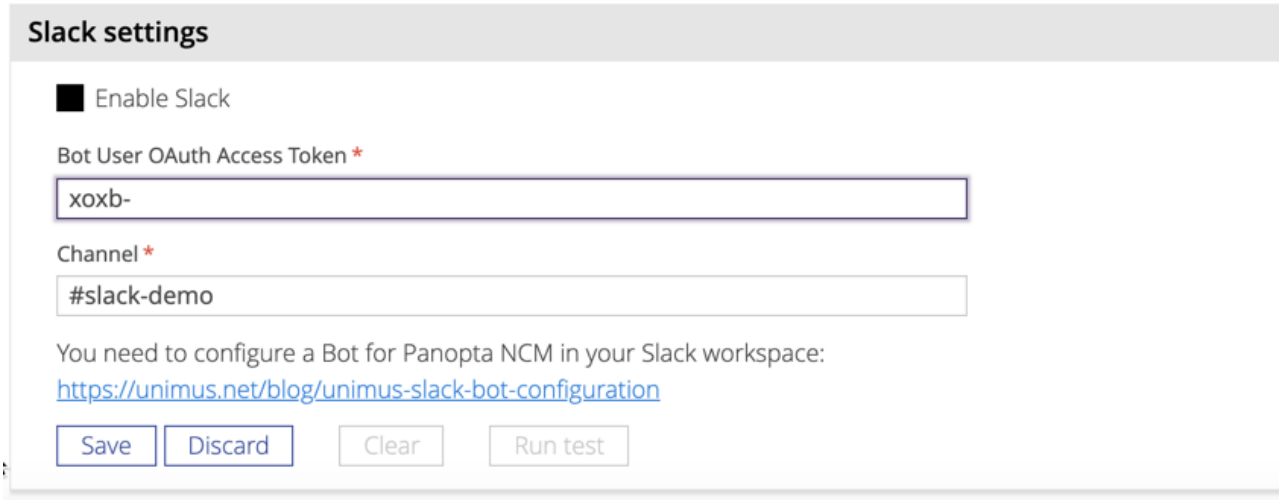
10. In your Slack Workspace, invite the NCM Bot user to the channel it will post to.



11.



Now navigate to **Notifications** > **Slack settings** in your NCM server:

12. 

13. Paste in your **Bot User OAuth Access Token** and your channel. You can use "#channel" to post in a channel, or "@user" to DM messages to a person.

Don't forget to **Save** your settings after pasting them in.
You can use the **Run test** feature to check if your notifications are working.

If the test works, the NCM Slack Sender is fully configured.
You will now receive Config Change Notifications into Slack as soon as the NCM detects a config change in your network.

System login

When a user attempts to log in to the system, the local user database is consulted.
If a corresponding account is found, the authentication method of that account is used to check the password.

For locally authenticated accounts, the password is simply checked against the local user database.
If the password is correct, user is logged in.

If Radius is enabled, a Radius Access Request is performed.
If Radius responds with an Access Accept message, the user is logged in to the system.

If the Radius client in the NCM is not enabled, authentication for this user fails.

Access accounting

After a successful login, an accounting record is always created in the local database.
Even Radius-based logins are accounted in the local DB.

If Radius is enabled, an Radius Accounting Request is performed. The failure of Radius accounting is not considered as an error.
This means a user will be allowed to log-in, even if Radius accounting fails.

This is done because even logins using the local database are accounted against Radius (if Radius is enabled).
If the NCM did not allow allow login after Radius accounting failed, system access would not be possible in case of Radius failure.

An accounting record in the local DB is always created, so this mechanism does not compromise the auditability of the system access.

Discovery

What does discovery do?

To make device management easier and to automate bulk operations, NCM uses a discovery mechanism.

The discovery mechanism works in stages:

Detect available connector

All available ports for all enabled connectors are checked.

Secure connectors are always preferred to insecure connectors.
For example, if a device is available over both Telnet and SSH, SSH will be used.

Detect credentials

Credential discovery works differently depending if the device is set to "Discover" credentials, or has "Bound" credentials.

If credential discovery enabled:
All credentials configured in the 'Credentials' settings are checked against the device.
Credentials are checked against the device in random order.

Whichever credentials work on the device first will be used for any future operations with this device.

When credentials bound to device:
Only bound credentials are tested against the device
If bound credentials do not work, the job fails

Detect device vendor and model

After correct connector and credentials are known, NCM discovers the device.

The Vendor, Type and Model of the device is discovered.

Detect available CLI modes on the device

After NCM knows exactly which device it's working with, it discovers available CLI modes (enable / configure mode - for example "privileged exec" and "configure" on Cisco).

When is Discovery performed?

Device addition into the system

The discovery mechanism is what makes only the address of the device required when adding a new device into NCM. NCM will automatically discovered everything else about the device.

Import

When 400 devices are imported into NCM, NCM will automatically discover all the necessary details about each of the devices, and start backing them up automatically.
If proper connectors and credentials are configured, no additional steps other than importing the devices are needed.

This saves time for the administrators, and automates the work-flow.

Scheduled backups

NCM performs a discovery before each scheduled backup.
This is to make sure NCM knows current information about the device (connector, credentials, vendor, model, etc.).

The discovery before backup is necessary to make sure NCM doesn't user wrong commands on the device.
In certain situations (ex. when a device is changed for another device) it's possible commands to generate backups on some vendors can actually cause configuration changes for other vendors.
Due to this (and other edge-cases) NCM does a discovery before every backup.

Credential changes

Discovery mechanism will also be used if any device operation fails.
For example, if credentials which were previously used on a device are no longer valid on that device, NCM will re-run discovery.

This means that if credentials used on 400 devices need to be changed, the only change needed in NCM is to reconfigure the credentials in 'Credentials' settings.
Discovery mechanism will take care of the rest.

Device hardware changes

Discovery will automatically be re-run when device hardware change is detected.
For example, when a Cisco device is replaced with a MikroTik, or a HP device with an UBNT device, NCM will handle this automatically.

This means that if devices are changed around the network, NCM will automatically adjust to the situation, without the need for user interaction.

How to find out why device didn't discover?

There are 2 places in NCM where you can see why discovery failed.

- Dashboard > Latest failed jobs
  You can select the discovery job, and press 'Show log'.

- Logs
  '/var/log/NCM'

If you can't identify why you device didn't discover properly from either of these, feel free to contact support.
We will always be happy to help with any issues.

Scheduling

All operations in NCM work on Schedules.

You can define any number of schedules, and each schedule can have different periodicity.
(cron expression, or pre-made hourly, daily, monthly, etc.)

The Default Schedule

There must always be at least one Default Schedule in NCM.
(and one is automatically created for you when NCM runs for the 1st time)

You can't delete the Default Schedule.
If you wish to delete it, you must first choose a different schedule to be the Default Schedule.

Any operations that don't have an explicit schedule will use the default schedule.

You can change which schedule is the default schedule in the **Schedules** view.
Any operations which use the default schedule will properly be rescheduled to the new default schedule.

Explicit schedules

If you wish some devices to be polled differently (or some operations to run differently) from the Default Schedule, you can assign an explicit schedule to a device/operation.

Simply create a schedule in the **Schedules** view, and assign it to the device.
(either during the device creation, or **Edit** the device)

The same applies to NSM Sync, Mass Config Push, and Network Scan as well - you can use an explicit schedule in each of the respective features.

If you delete a schedule that is used like this, all operations that were using the schedule will be re-scheduled to the Default Schedule.

User security and access roles

Account security roles

Each system user has a role (Administrator, Operator, Read-only, None).
Access to various parts of the application is limited according to users access role.

This article describes the access that each role has.

System access table

	Administrator	Operator	Read-only	None
Login	✓	✓	✓	X
Access to all features	✓	✓ *	read-only access *	X
Change any settings	✓	✓ *	X	X
License settings	✓	read-only access	X	X
User management	✓	X	X	X

- - see details for additional limitations

Administrator

Accounts with the **Administrator** access role have full access to NCM and all features within.

In other words, **Administrator** users are not limited in any way.

Operator

Accounts with the **Operator** role have full read/write access to NCM, except:

- Operators have no access to the **User management** screen
- Operators can not delete **Device Tags** (as this can affect access policies)
- Access to **License settings** is read-only (can see, but can't change license key)
- Access to **Sensitive data stripping** is read-only

We recommend that most users have **Operator** set as their access role.

Users authenticated from Radius are assigned the **Operator** role.

Access for operator accounts can be further restricted using **Device access tags**.
Please see this wiki article for more information: Device access restrictions.

Read-only

Read-only role accounts have read-only access to NCM - they can not configure or change any settings.

Additionally, read-only accounts have these limitations:

- Read-only accounts have no access to the **User management** screen
- Read-only accounts have no access to the **License settings** menu
- Read-only account do not have access to **Show Password** and **Show All Passwords** in the **Credentials** screen

Access for read-only accounts can be further restricted using **Device access tags**.
Please see this wiki article for more information: Device access restrictions.

None

Accounts with the **None** role have no access to the application - they can not even log in.

This role is meant to deny access to NCM for a particular account, without the need to delete that account.

Device access restrictions

Device Tags can be used to further limit access of accounts.

If an account has no tags defined, it can see all devices.

Affected features

- **Devices** view
- **Zones** view
- **Device Tags** view
- **Backups** view
- **Config search** view
- **Mass config push** view
- **Credentials** > **Show usage**
- **Schedules** > **Show scheduled tasks**
- **Other settings** > **Per-Tag connectors**
- **Other settings** > **Sensitive data stripping**

Assigning tags to devices

We can create as many tags as needed using the **Add** button in the **Device tags** table.

After a tag is created, we can tag devices with this tag using the **Tag devices** table.
Selecting a tag and pressing **Un-tag devices** will allow us to remove devices from this tag.
(if an account is already access-limited by tags, they can only Tag/Untag devices with tags they have access to)

On the **Devices** view, you can also see a clickable **Tag** icon which allows for tagging/untagging devices.

Tagging devices from Zones

Zones can add tags to all devices that belong to the Zone.
To use this, simply add a tag to a zone in the **Zones** view, and all devices in that zone will be tagged with the selected tag (s)

Assigning tags to accounts

Tags can be applied for access restrictions to any account, to limit the access of that account only to devices with that tag.
If tags are applied to an account, the account can also only see zones that also have this tag applied in the **Zones** view.

By default, users have access to all devices present in the NCM.

You can add tag limitations to accounts in the **Device access** table in **User management**.
To do this, create rules (account / tag bindings). You can add as many rules as needed to achieve proper access separation.

Usage example

We want user 'Bob' to only have access to the WiFi APs in NCM.
For example, when Bob does a Config search for 'password', he would see results only in configs of the APs.

First we create user 'Bob' with 'Read-only' access role.

Next we create the 'APs' device tag in **Device tag"** screen.
After the tag is created, we give access to the devices with the 'APs' tag to Bob.
We navigate to **User management** > **Device access**, click **Add** and add to Bob's account the 'APs' tag.

Now we need to tag the right devices with the 'APs' tag.
In **Device tags** screen, we select the tag, and select **Tag devices**.
We add the tag to the appropriate devices.

After this, our 'Bob' user will only see the devices that are tagged with the 'APs' tag when using Config search, or any other NCM features.

Accounts with restricted access

Accounts with restricted access will not see devices they don't have access to in any parts of the NCM.
Zones which are not tagged with any tags that the account has access to will also not be visible.

For a complete list of all NCM features which will be access restricted, please check the Device access restrictions article.

Since the NCM uses a tag-based approach to per-device access restriction, there are a few specific behaviors that can benefit from clarification.

- Operator level accounts can NOT delete Tags - this is a security measure as deleting Tags can have an effect on access policies.
  (for example, deleting a tag that is the only tag restricting access for an account would remove access restrictions from an account - see below)

By default, accounts have access to all devices present in NCM.
When device access restrictions are applied to an account, this causes some specific behavior:

- all tagging operations (in **Device tags**, **Devices > Tags**, **Zones > Manage tags**, etc.) for access restricted accounts are limited to only tags the account has access to
  (an access limited account can only tag and untag with tags associated with that account)
- access restricted accounts can only see Zones which have tags they have access to
  (only zones tagged with at least one tag the account is restricted with will be visible in the **Zones** view)

Sensitive data stripping

NCM supports stripping sensitive data from backups of devices.
When this feature is enabled, passwords, pre-shared-keys, and other sensitive data will not be stored in backups of devices.

This can be enabled in "Other settings > Sensitive data stripping".
You can enable stripping sensitive data globally ("Default sensitive data stripping policy"), or per-Tag.

Per-Tag policy always over-rides the default policy for devices that the Tag applies to.
If a single device belongs to Tags that specify both the "Never strip" and the "Always strip" policy, the more secure option ("Always strip") will be applied.

Always check for desired behavior

When using this feature, **always verify if all sensitive data is properly being stripped**.
Also verify if data which should be present is not getting stripped when it should not be.

Supported devices

Currently, sensitive data stripping is supported on these devices:

```
Cisco ASA
Cisco IOS
Cisco IOS XR
Cisco Nexus
Cisco NXOS (generic NXOS)
```

If a backup is ran on a device which is not yet supported and it's configured for sensitive data stripping, the backup job will fail.
(fail reason will be "SENSITIVE_DATA_STRIPPING_ERROR")

We are periodically adding support for more devices to the above list.
If you want to use sensitive data stripping with any devices not listed above, **please let us know**.

Integrations

This page provides a list of supported integrations across the following categories: alerting, cloud monitoring, third-party integration, Security Fabric, and SNMP.

Alerting

- Campfire Chat
- HipChat
- Jira and Jira Service Desk
- Microsoft Teams
- OpsGenie
- PagerDuty
- Salesforce
- ServiceNow
- Slack
- StatusPage.io
- Webhooks
- Zendesk

Third-Party Integrations

- Arista CloudVision
- Alkira

Cloud Monitoring

- AWS
- Azure
- Google Cloud Platform
- Kubernetes

Security Fabric

SNMP MIBs

- Citrix
- Cisco
- Synology
- Ubiquiti Networks
- MikroTik
- Netgear
- Nimbus Software
- CyberPower
- Cambium Networks
- HP
- VMware
- Fortinet
- Palo Alto Networks
- Dell
- SonicWall
- NetApp
- Alcatel
- Aerohive
- Checkpoint
- Juniper
- Nutanix
- Barracuda
- CyberArk
- Mitel
- Broadcom
- APC
- Vertiv
- Polycom
- Riverbed
- HPE
- Ruckus
- LSI
- Axis Communications AB
- Server Technology, Inc.
- Broadcom Brocade
- Silver Peak
- F5

- Unitrends
- Raritan
- Lantronix
- Watchguard
- Nimble Storage
- Brocade
- Panduit
- AVTECH
- Tripplite
- Extreme Networks
- OpenGear
- Meru Networks
- PulseSecure
- HP Printers
- Avaya
- Arista
- Kemp Technologies
- Zscaler
- Transition Networks
- IBM
- Eaton
- Sentry
- Chatsworth Products
- Huawei
- Enlogic PDU
- Network Technologies Inc
- Deep Sea Electronics
- Isilon Systems
- Infinet Wireless
- Hikvision
- Avi Networks
- Infoblox
- Versa Networks
- Metaswitch Networks
- Cleo
- Net SNMP
- Nasuni
- Hitachi Vantara
- Hangzhou H3C Technologies
- A10 Networks
- QNAP
- Blue Coat Systems
- Monnit

- NetBotz
- Morningstar
- Lester Electrical
- Alstom
- Digi International

FAQ

What do you count as a Service?

A service is any network check (for example, a basic ping test or a complete HTTP transaction with content verification) or any metric monitored with our agent or SNMP polling.

Check Frequency

Any outage, even a short one, immediately impacts your business - customers who aren't able to reach your website or email your company will look to your competitors. Checking all of your services every 60 seconds means that we detect even short outages and give you the ability to respond quickly, as well as detecting situations where there are repeated short-duration outages that add up to substantial downtime but might otherwise get missed with less frequent checking.

If you are a service provider using FortiMonitor to validate your service level agreements (SLAs), high check frequencies are critical. If your SLA guarantees 99.9% only allows for just over 10 minutes of downtime per week. If you only check your site once every 10 minutes, any detected outage, even a very short one, will be counted as your quota for downtime for the week. Higher-frequency checks means your availability metrics have a greater resolution and allow you to more accurately represent your SLA performance.

Note, with our system the timing of outage notifications is completely separate from the frequency of checks - if you don't want to be alerted to short-duration outages you can setup your notification schedule so that alerts are only sent after, say, 10 minutes. Each outage, however short, will still be included in daily, weekly, and monthly performance reports and availability calculations.

How do you minimize false alerts?

Our system uses a custom algorithm to confirm every outage before beginning any notifications. After we first detect a failed service check, confirmation checks are performed from a number of geographically nearby monitoring locations, and only after a quorum of these locations confirm the outage does the notification process begin.

Once the service has been restored and is returning successful checks, the same process is repeated to confirm the resolution. With this approach, any spurious failures are eliminated and alerts are only sent on real, confirmed problems.

Due to the design of our system, the entire confirmation process is handled quite quickly, typically within 20 seconds of the initial failure detection.

What are you checking when you check my servers?

We support a range of different check types for all of the common services that we support. At the most basic, we are checking to see if a service on a given port is listening for connections - this is referred to as a "port check" in the control panel. For many of the most used services, including HTTP, HTTPS, IMAP, POP, SMTP, and DNS, we also support more advanced types of checks where we, for example, download a full page from the web server using a URL that you provide or perform an IMAP login session with credentials you provide.

We also support arbitrary TCP/IP port checks on whatever ports you specify, as well as Ping checks. If you have a specific application you need monitored, or are looking for more thorough checking for one of the services that we currently support, please contact our support team to discuss the specifics - we are continually expanding the portfolio of checks that we perform and most likely can accommodate your request.

From where is my site monitored?

Each service that is monitored has a primary location that we check from. Once an outage is detected, we immediately perform checks from our other locations around the world to confirm the outage before any notification begins.

If you would like to regularly check from multiple locations you can set up checks for the same service from multiple locations in the control panel. These services will be clearly marked with the other location in the control panel.

You can find a complete list of all of our monitoring locations here.

Can you monitor my custom application?

We can monitor any application that has a remotely-accessible network interface. At the very minimum, we can do basic port checks to ensure that the application is listening on a given port or ports. More in-depth monitoring is possible, especially if your application has a HTTP-like protocol (using our standard HTTP check functionality).

More advanced checks can also be incorporated into our system by our engineers, and we are currently working on a monitoring extension API that will allow you to design and implement your own checking logic specific to your application. Contact our support team if you are interested in pursuing this.

Does FortiMonitor do SaaS monitoring?

We do, and we will be glad to help you with your SaaS. If the SaaS that you need to monitor is web-based then you can set up a multi-step HTTP check. To learn more about how to set a multi-step HTTP check see our support page on that topic. The FortiMonitor Monitoring Agent can also be very useful for SaaS monitoring. Since the Agent can collect data directly from your server, you can build custom plugins to check metrics specific to that application. If neither of those solutions meets your needs then feel free to contact our support team. We will work with you to create a solution for your specific needs.

Can FortiMonitor collect information needed to check an SLA?

Yes we can. The nature of this service will change based on the nature of your SLA. If your SLA promises that your service will be up and reachable a certain percentage of the time, then you can set up a simple HTTP check of that service. This should give you all the information you need to check the agreement, to learn how to set up a simple HTTP check see this article. However, if monitoring your SLA requires a more involved solution then we encourage you to contact our support team.

I have servers behind a firewall, can I monitor those?

If some or all of your infrastructure is behind a firewall and not reachable by our external monitoring nodes, then the OnSight vCollector is what you need. Since the OnSight vCollector can be installed behind the firewall, it can securely and accurately monitor your servers. Results from the OnSight vCollector are pushed to our central monitoring infrastructure for analysis, reporting, and alert generation.

Does FortiMonitor offer a partnership program?

We're always looking for companies to partner with to either extend the range of our monitoring capabilities or to offer our monitoring services to their customer base in either a white-label or co-branded manner. Please contact our sales team to discuss further.

Do I need the Server Agent and the OnSight vCollector?

The Server Agent and the OnSight vCollector perform two separate functions. This article will outline the purpose of each, and then explain how they interact and what the benefit is of using both.

The FortiMonitor Agent installs directly on your server (whether virtual or bare metal) and provides host-level resource utilization and application-specific metrics. The Server Agent gives you details like CPU, memory, disk space utilization, bandwidth, etc, and also allows you to monitor processes and services. We also have a number of plugins available to monitor applications like web servers, databases, caches, and so on.

This is critical information for identifying the source of incidents because you can review the activity on the host before and while the outage was happening.

The OnSight vCollector functions the same as one of our public monitoring probes, running uptime checks and synthetic monitoring against your publicly available servers and web endpoints, but the Onsight vCollector sits behind your firewall and monitors on your private network. The OnSight can run uptime and synthetic checks on your servers behind the firewall, monitoring services like HTTP/HTTPS, RDP, database connectivity, and port availability.

OnSight can also function as a proxy for servers that have the Server Agent installed, but no connection to the outside internet. OnSight can also monitor network devices with SNMP polling and receive SNMP Traps.

You don’t need to use OnSight to use the Agent, or vice versa, but using both gives you a more full view of your infrastructure with the different kinds of monitoring each offers.

Data Retention and Compliance

FortiMonitor compliance

FortiMonitor infrastructure runs on the Google Cloud Platform and falls under the same compliance as GCP. The following links provide deeper information on security and compliance maintained by Google:

Google Security Model
Google Infrastructure Compliance Overview

Encryption

All data is encrypted in transit using TLS 1.2 or higher encryption.
All data is also encrypted at-rest using 256-bit AES. This includes all authentication information used for performing synthetic and application checks.

Data retention

FortiMonitor retains data related to servers and devices (names, IP's, etc.), system metrics, check results, and outage/incident-related information.

Outages, incidents, and hourly aggregates of all metrics are stored for 24 months. 60-second resolution of data is stored online for 6 months. After 6 months, they are rolled up into hourly summaries.

Data deletion requests

Upon account termination, you can request deletion of your data using the **Account Cancellation** option in FortiCare.

Get support

FortiCare provides access to support services such as creating and managing support tickets. To get started, go to https://support.fortinet.com/.

You can also go to https://support.fortinet.com/ticket/index.aspx to create a ticket directly.