New Features Guide

FortiOS 8.0.0

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

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

FORTINET BLOG
https://blog.fortinet.com

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

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

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

FORTIGUARD LABS
https://www.fortiguard.com

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

FEEDBACK
Email: techdoc@fortinet.com

April 21, 2026
FortiOS 8.0.0 New Features Guide
01-800-1177878-20260421

TABLE OF CONTENTS

Change Log **8**
Overview **9**
GUI **10**
General usability enhancements 10
Summary panel in Log Details 10
GUI support for preferred outbound route map options 13
Seven-day setup period for GUI and CLI configuration 14
VDOM selection for Central Management in the GUI 15
GUI support for configuring proxy ARP on VLAN interface 16
Improve switch management 17
Display Known Exploited Vulnerabilities from FortiClient 20
Custom GUI themes and admin-level personalization 26
Enhanced GUI support for Inline-CASB SaaS Application 32
FortiView 35
Graphical display for application performance monitoring analytics 35
Network **41**
General 41
IPv6 Multicast Routing Enhancement with BSR Support Added 41
BGP neighbor naming support 48
Increased AS path item for BGP route maps 51
IPv6 PCP support for DNAT46 in NAT64 deployments 52
Customizable CoS marking for locally generated ARP packets 55
Firewall address support for IPAM rules 59
Graceful BGP shutdown 64
TCP congestion control enhancement with BBR 72
Session helper statistics 74
Enhanced EVPN support for VXLAN Anycast gateway and type-5 routing 75
Enhanced PIM support for IPv6 across all VRFs 91
GPON support for FortiGates with SFP/SFP+ ports 97
802.1X, MAB, dynamic VLAN, and address object support for software switch 99
Dynamic BGP learning of ISDB reputation IPs 116
Enhanced multicast session key for virtual wire pair 119
Next-hop self-support for VPNv4 and VPNv6 route reflectors 121
IPv6 122
IPv6 Link-local configuration enhancement 122
IPv6 proxy address and address group object support 123
Explicit and transparent proxy 124
Secure explicit proxy with client certificate blocklist enforcement 125
SD-WAN **128**
Overlays and underlays 128
SD-WAN underlay bandwidth steers traffic 128
ToS-based SD-WAN duplication matching 135
Stop packet duplication upon reaching bandwidth utilization threshold 141
Selective SD-WAN duplication 147

Use secondary tunnel for FEC redundant parity packets ... 152
ToS matching and negate options on adaptive FEC profiles ... 159
Control SD-WAN interface usage based on monthly traffic volume (quota) ... 166
Confine packet duplication across multiple overlays to the same spoke ... 173
On-demand duplication at the Hub using remote health-check ... 178
Performance SLA ... 187
SD-WAN speed test scheduling improvements ... 188
Policy and objects ... 196
Policies ... 196
Upstream SSL/TLS support for virtual load-balancing servers ... 196
Objects ... 199
Custom tags for addresses, policies, and dynamic tag address groups ... 199
Zero Trust Network Access ... 212
General ... 212
ZTNA configuration simplification ... 212
ZTNA service connector ... 235
Security posture and EMS connectors ... 242
Support tags in dual stack IPv4/IPv6 ZTNA policies ... 242
Security profiles ... 246
Others ... 246
File filter warning action for HTTP downloads ... 246
Local URL filtering with custom FortiGuard categories ... 250
New classification framework for application signatures and FortiView support ... 256
Support for hybrid PQC SSL deep inspection in proxy mode ... 272
Enhanced UTM protection with WebSocket traffic inspection ... 275
FortiGuard-free rating for local and external categories ... 282
Expanded CASB SaaS application database ... 284
Protecting LLM and GenAI ... 286
Application control support for generative AI ... 286
Agentic AI protocol support in FortiOS ... 286
VPN ... 293
IPsec and Agentless VPN ... 293
New Cloud SDN Orchestration VPN wizard for AWS ... 293
Support matching firewall policies and policy routes based on source IP geography of dial-up IPsec remote users ... 307
Send security posture tags from FortiClient to FortiOS directly as JWT ... 311
TLS 1.3 based VPN over TCP ... 316
Native VPN remote access support in VPN wizard ... 323
Default to IPAM addressing for remote access VPNs in VPN Wizard ... 341
Display interface and tunnel alias in various diagnose and get commands ... 346
Support SM3/SM4 cryptographic algorithms for IKEv1/ IKEv2 ... 350
Show only recommended IKE proposals by default ... 356
User and authentication ... 358
Authentication ... 358
Enable secure LDAP connection by default in the GUI ... 358

LAN Edge ... **360**
Wireless ... 360
Support FortiAP management through IPv6 ... 360
Improve CAPWAP stability ... 362
Support dynamic redirect URLs from Cisco ISE ... 364
Support IPsec traffic offloading from the FortiAP ... 367
Enhance DARRP with FortiAIOps ... 369
Support up to 16 VAPs on FortiAP G and K-series models ... 372
Support Wi-Fi 7 MLO on FortiAP K-series models ... 374
Support FQDN in Layer 3 ACLs ... 377
Add GUI support for 6 GHz Channel Utilization ... 379
Support customizable DHCP Option 82 configurations ... 380
Optimize and update default DARRP parameters ... 383
Add GUI support for capturing wireless packets ... 384
Support tuning Clear Channel Assessment threshold values ... 387
Allow FortiAP radio and FortiWiFi local radio to be disabled ... 391
Support DVLAN assignments from RADIUS and VLAN pooling together ... 394
Support multiple FortiAP groups per VLAN entry for VLAN pooling ... 396
Add RADIUS authentication survivability for 802.1X SSIDs ... 396
Automated FortiAP location positioning system ... 399
Support FortiGuest assignments of RADIUS attributes to MPSK groups ... 404
Switch controller ... 407
Add maximum power control for PoE ports ... 408
Increase number of supported switches ... 409
Advertise and control EEE using LLDP in the switch controller ... 409
Support trusted host settings for managed switches ... 410
Control when custom commands are pushed ... 412
Provide private data encryption for managed switches ... 414
Support MAC move for managed switches ... 415
Enable advanced switching features in the GUI ... 420
Control the number of 802.1X-authenticated clients per port ... 422
Support IGMP-snooping static group configuration ... 422
Dynamic MAB session handling ... 423
Enhance NAC/DPP port control and QoS/PoE support ... 424
Add global port-selection criteria for some switch models ... 426
Using NAC with 802.1X authentication in the FortiOS GUI ... 427
System ... **431**
General ... 431
Configurable timeout for log file system check ... 431
Administrator access hardening ... 433
SCP file transfers ... 434
VM license grace period monitoring enhancement ... 437
Configurable FQDN Host header for FortiGate proxy communication ... 438
Additional commands supported before device registration ... 440
Legal third party disclosure panel ... 440
Per VDOM replacement message customization ... 442
Configuration revisions and logout backup default changes ... 444
FortiAI assistant and CLI Code Lab ... 446

Enhancements to required upgrade when firmware license is invalid or device is EOES 458
High availability 467
IPv6 support for in-band management IP 468
Support IPv6 multicast route synchronization in HA 469
FGSP source IP and interface selection 470
HA monitoring support for software switch member interfaces 472
FGCP and FGSP synchronizes full-cone expectation sessions when session-pickup-expectation enabled 475
Enhanced FGCP monitoring with interface group awareness 481
GUI support for HA actions, health status, and config diff 485
FGSP synchronizes firewall-authenticated users 487
HA role shown in CLI prompt 492
SNMP 492
SNMP support for new IP pool statistics OIDs 492
FortiGuard 494
Enhanced FortiGuard server selection and load balancing 495
FortiGuard rating server prioritization improvements 503
API 506
API support to manage firmware for FortiGate extended devices 506
Security Fabric 512
Fabric settings and connectors 512
Cloud-based Fabric Feed synchronization 512
Label-based sandbox exemptions for sensitive data control 517
Simultaneous FortiSandbox Cloud and on-premise integration 520
External SDN connectors 536
Support for Cisco ACI External EPG Subnets in Direct Connector 536
Negation support in ACI direct connector address filters 542
Log and report 547
Logging 547
Enhanced TACACS+ accounting log detail 547
Custom log format support for syslog server 549
Secure log upload enhancements with SFTP and LZ4 support 558
Cloud 562
Public and private cloud 562
AWS SDN connector EKS filtering support 562
FortiOS Carrier 565
Selective GTP FGSP sync: only S10 tunnels 565
Selective GTP FGSP sync: only synchronize GTP tunnels 569
New RAT types 573
3G to 4G/5G handover support 575
Increase GTP tunnel limit per profile to 50M tunnels 577
GTP profile tunnel limit 577
FortiASIC 579
Hardware acceleration 579
Improvements for NP7-offloaded GRE tunnels 579

NP7 offloading of TCP and UDP sessions denied by firewall policies to reduce CPU usage ... 581
NP7 support for traffic shaping based on dynamic RADIUS VSAs ... 582
Hyperscale firewall ... 582
New SNMP counters to support CGNAT IP pool monitoring ... 582
Hyperscale HA hardware session synchronization improves support for PBR sessions ... 586
Hyperscale CGNAT EIF session timer options ... 587
Operational Technology ... 592
System ... 592
MACsec support for FortiGate Rugged Models with hardware switch ... 592
FortiGate Rugged digital I/O trigger logging enhancement ... 593
Index ... 601
8.0.0 ... 601
GUI ... 601
Network ... 601
SD-WAN ... 602
Policy and objects ... 602
Zero Trust Network Access ... 603
Security Profiles ... 603
VPN ... 603
User & Authentication ... 604
LAN Edge ... 604
System ... 605
Security Fabric ... 605
Log & Report ... 606
Cloud ... 606
FortiOS Carrier ... 606
FortiASIC ... 606
Operational Technology ... 607

Change Log

Date	Change Description
2026-04-21	Initial release.

Overview

This guide provides details of new features introduced in FortiOS 8.0. For each feature, the guide provides detailed information on configuration, requirements, and limitations, as applicable. Features are organized into the following sections:

- GUI
- Network
- SD-WAN
- Policy and objects
- Zero Trust Network Access
- Security profiles
- VPN
- User and authentication
- LAN Edge
- System
- Security Fabric
- Log and report
- FortiOS Carrier
- FortiASIC

For a list of features organized by version number, see Index on page 601.

GUI

This section includes information about FortiOS GUI related new features:

- General usability enhancements on page 10
- FortiView on page 35

General usability enhancements

This section includes new features related to general usability enhancements:

- Summary panel in Log Details on page 10
- GUI support for preferred outbound route map options on page 13
- Seven-day setup period for GUI and CLI configuration on page 14
- VDOM selection for Central Management in the GUI on page 15
- GUI support for configuring proxy ARP on VLAN interface on page 16
- Improve switch management on page 17
- Display Known Exploited Vulnerabilities from FortiClient on page 20
- Custom GUI themes and admin-level personalization on page 26
- Enhanced GUI support for Inline-CASB SaaS Application on page 32

Summary panel in Log Details

This information is also available in the FortiOS 8.0 Administration Guide:

- Forward Traffic log page

For *Forward Traffic* logs, the top of the *Log Details* pane now includes a summary panel. The summary panel shows the log timestamp, policy name, and a summary of security events with hover-enabled tooltips and access to quick actions. Previously this information was available under the *Log Details* > *Security* tab.

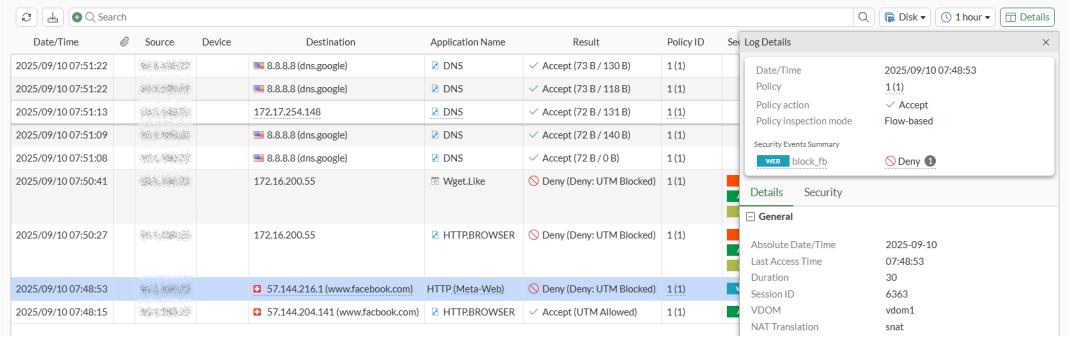
The summary panel:

- Is available for *Memory*, *Disk*, *FortiAnalyzer*, and *FortiGate Cloud* log sources.

  For *FortiGate Cloud* log source, the summary panel will always display *No security events found* because forward traffic logs from FortiGate Cloud do not contain the utmref field, which is necessary for populating security event data.
- Is only available in *Log Details* when the selected log has security events associated with it
- Only displays for non-implicit deny policy logs

To view the summary in Log Details:

1. Go to *Log & Report* > *Forward Traffic*.
2. Set the log source to *Memory*, *Disk*, or *FortiAnalyzer*.
3. Select a UTM log entry, and click *Details*. The *Log Details* pane is displayed with a summary panel at the top:

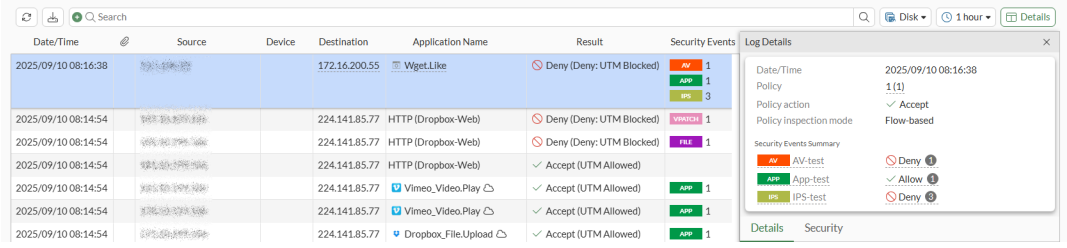


Option	Description
Date/Time	Date and time of the log entry
Policy	Name of the policy used
Policy action	Policy action (Accept or Deny)
Policy inspection mode	Policy inspection mode (Flow-based or Proxy-based)
Security Events Summary	Lists the type and name of the security profile(s) used, final action (Allow or Deny), and the number of security event logs for the event. If the same security profile has multiple security events, and if at least one (1) of the events is categorized as *Deny*, then the final action is *Deny*, and the number bubble indicates the number of deny logs out of the total logs.

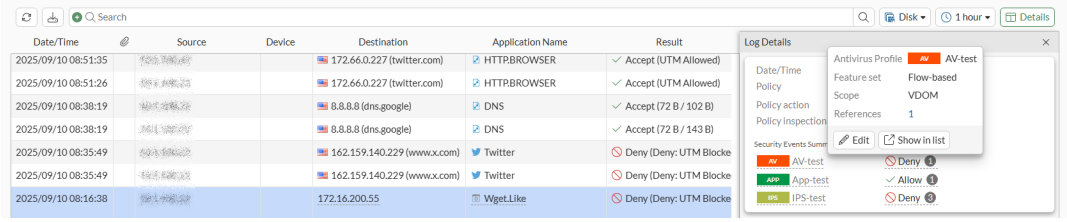
In the *Security Events Summary* section:

- When multiple security profiles (for example, APP, AV, and IPS) log security events, all security profile names and the final action (Allow or Deny) are displayed. The number bubble indicates how many security event logs are available for the event.

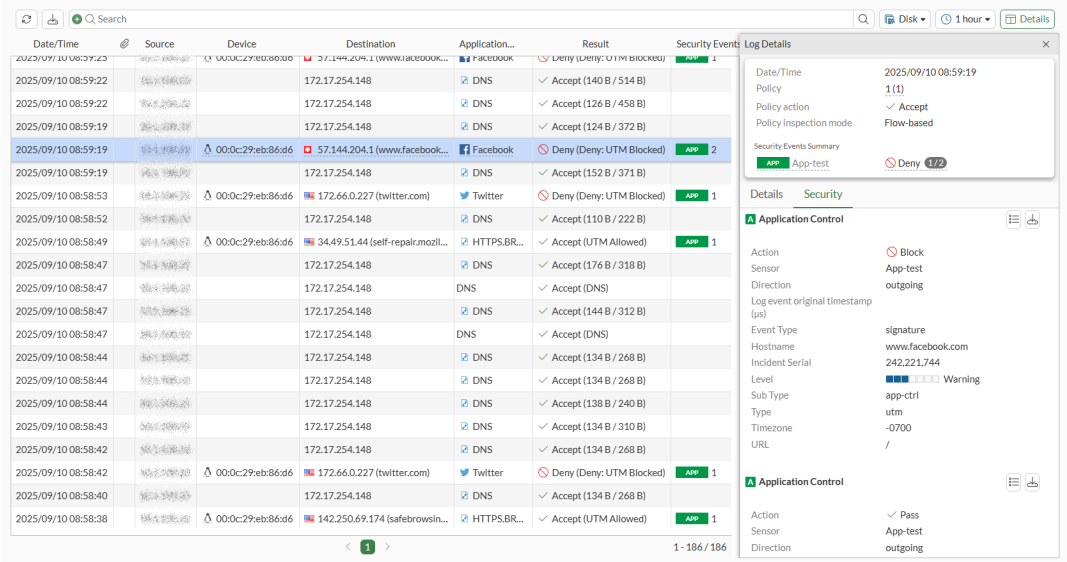
  In this example, multiple profiles (AV, APP, and IPS) logged security events, and each has one (1) security event log, except for IPS, which has three (3):



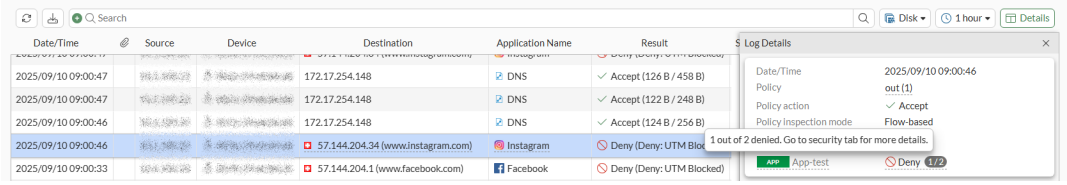
- Hover over any security profile to display a tooltip of information and access the *Edit* and *Show in list* buttons:



- When multiple security events are listed for the same security profile, the final verdict is *Deny*, if at least one of the events is categorized as *Deny*. More information about the deny logs is available on the *Security* tab.

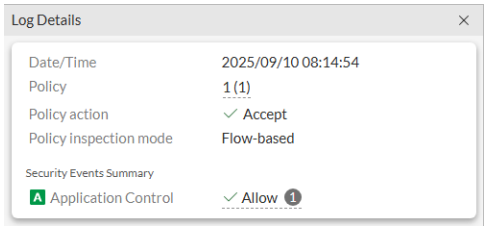


- Hover over the number bubble. A tooltip displays how the number of denied logs and *Go to security tab for more details*.



If any event has multiple logs with same security profile, but all logs are with same action (Allow or Deny), then the tooltip for the number bubble only displays *Go to security tab for more details*.

- When the security profile no longer exists, then the *Security Events Summary* section displays the generic security event name of *Application Control*.



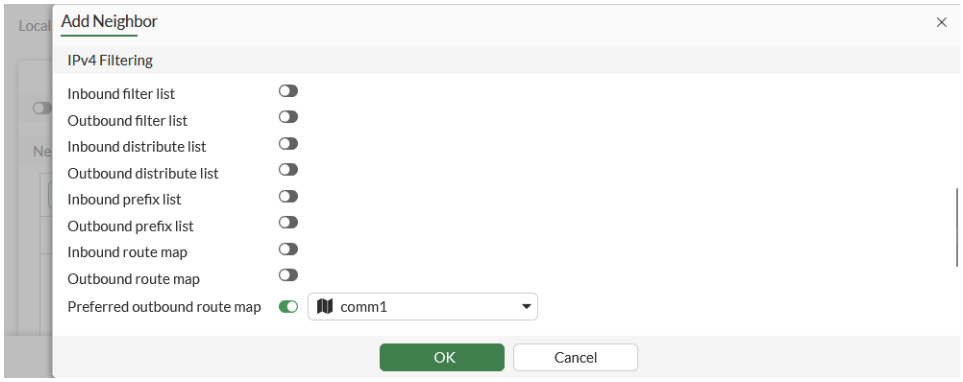
GUI support for preferred outbound route map options

Enhanced the routing configuration by adding GUI support for `route-map-out-preferable` and `route-map-out6-preferable`, which were previously only configurable using the CLI.

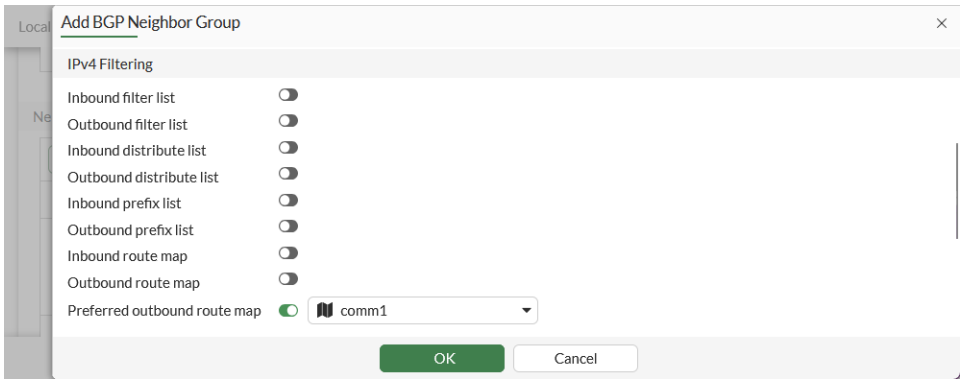
To configure a preferred outbound route in the GUI:

1. Configure a route map. See Route maps.
2. Go to *Network* > *BGP* and click *Create New* in the *Neighbors* or *Neighbor Groups* tables.
3. Enable *Active IPv4* to configure *IPv4 Filtering*.
4. Enable *Preferred outbound route map*, then select a route map from the drop-down list.

   IPv4 neighbor:

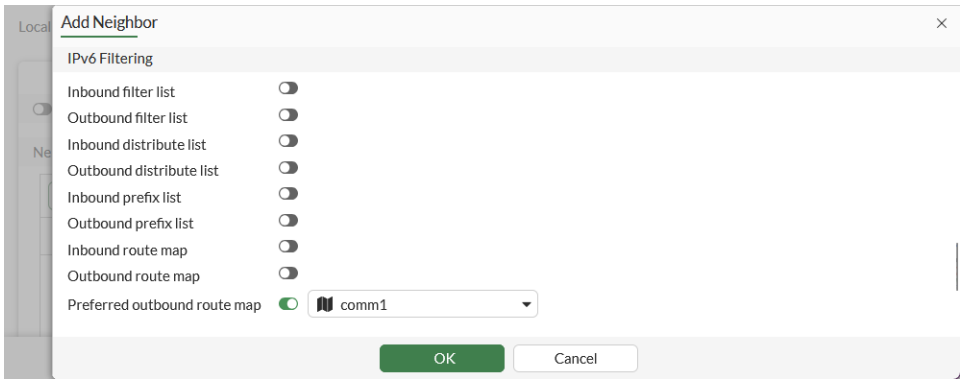


   IPv4 neighbor group:

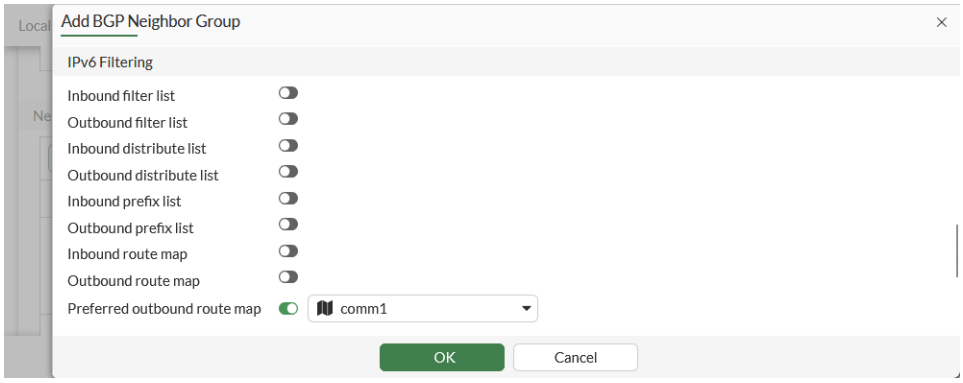


5. Enable *Active IPv6* to configure *IPv6 Filtering*.
6. Enable *Preferred outbound route map*, then select a route map from the drop-down list.

   IPv6 neighbor:



IPv6 neighbor group:



7. Configure the remaining settings as needed, then click *OK*.

Seven-day setup period for GUI and CLI configuration

Newer FortiGate models running FortiOS 8.0.0 now allow seven days to configure the FortiGate before device registration to FortiCare is enforced. When the seven-day setup period expires, you must register the FortiGate to FortiCare to continue accessing the GUI and CLI. The setup period allows, for example, distributors to configure FortiGates before sending the devices to end users for low-touch or zero-touch provisioning.

See Enforce FortiCare registration after new GUI login for information about registration requirements.

The seven-day setup period starts when the FortiGate is first powered up and counts down while the device is powered up. Log out of FortiOS and power down FortiGate to pause the countdown. The setup period reflects the cumulative device uptime and ends after seven days or when the FortiGate is registered to FortiCare, whichever occurs first.

End users can receive FortiGates after the seven-day setup period has expired. In this case, registration to FortiCare will be required at first login.

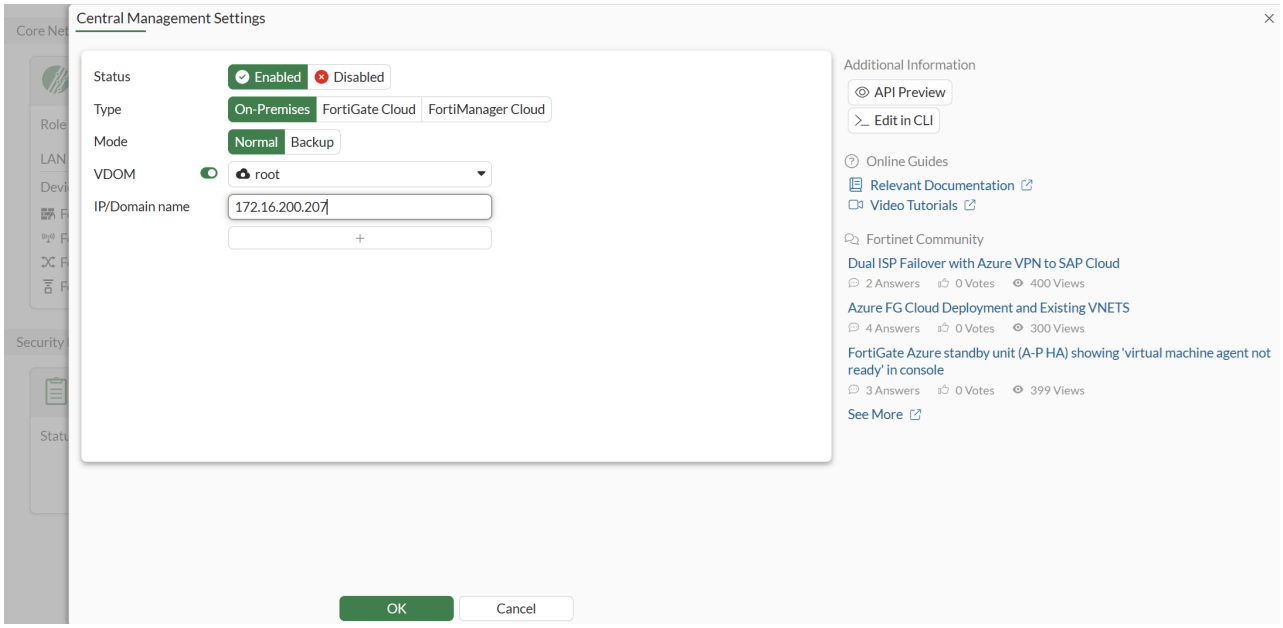
For more information about this feature, see Seven-day setup period for GUI and CLI configuration.

VDOM selection for Central Management in the GUI

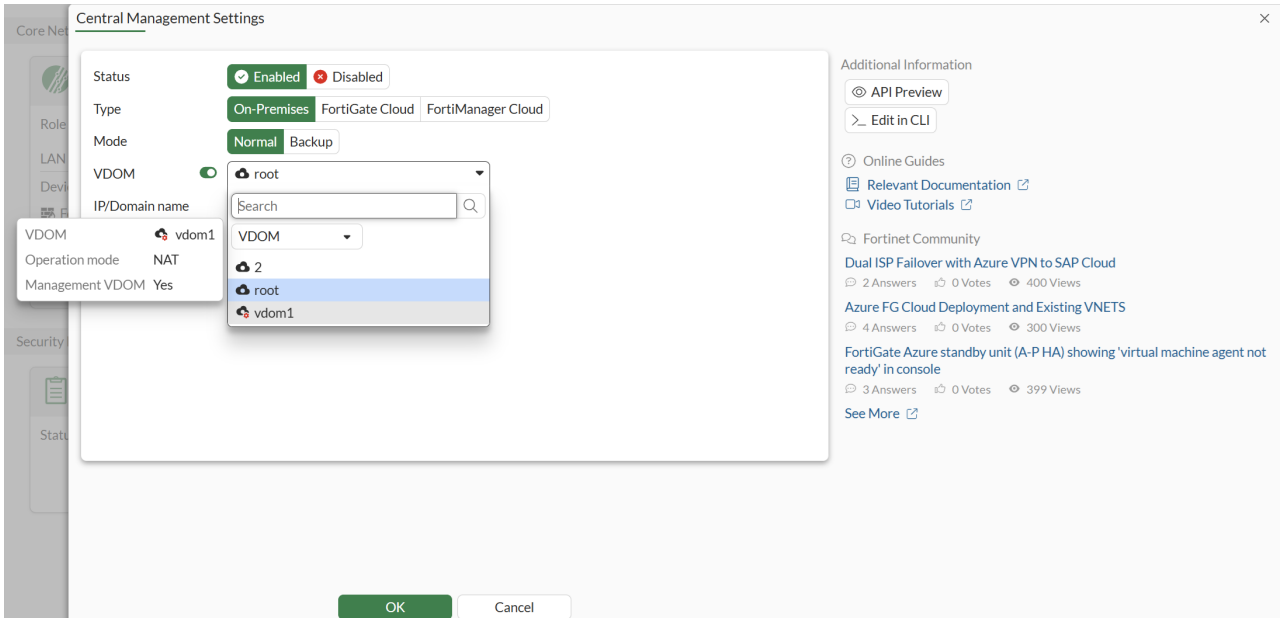
A VDOM dropdown has been added to the Fabric Central Management GUI page for FortiGate devices with multi-VDOM enabled. This provides a more intuitive configuration experience by allowing VDOM selection directly in the GUI.

To configure the central management VDOM in the GUI:

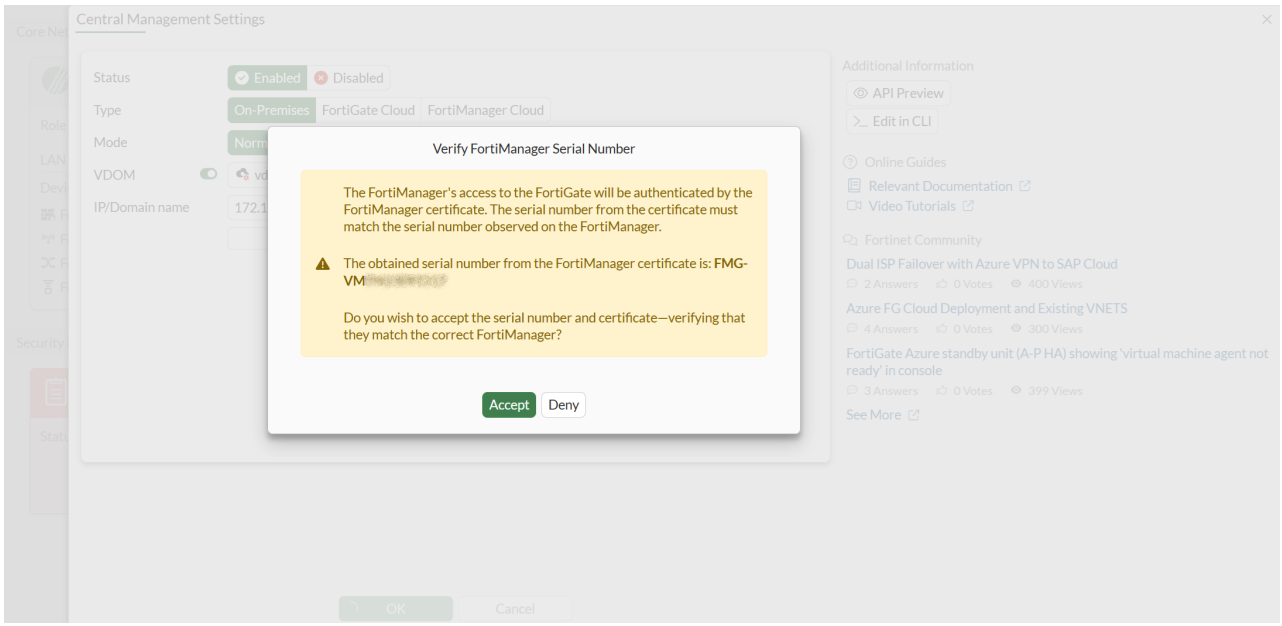
1. Configure central-management in the GUI:



2. Select the VDOM that has a route to the FortiManager:



3. Verify the FortiManager serial number:



4. Confirm the VDOM configuration in the CLI:

```
# show system central-management
config system central-management
    set type fortimanager
    set serial-number "FMG-VM**********"
    set fmg "172.16.200.207"
    set vdom "vdom1"
    set enc-algorithm default
end
```

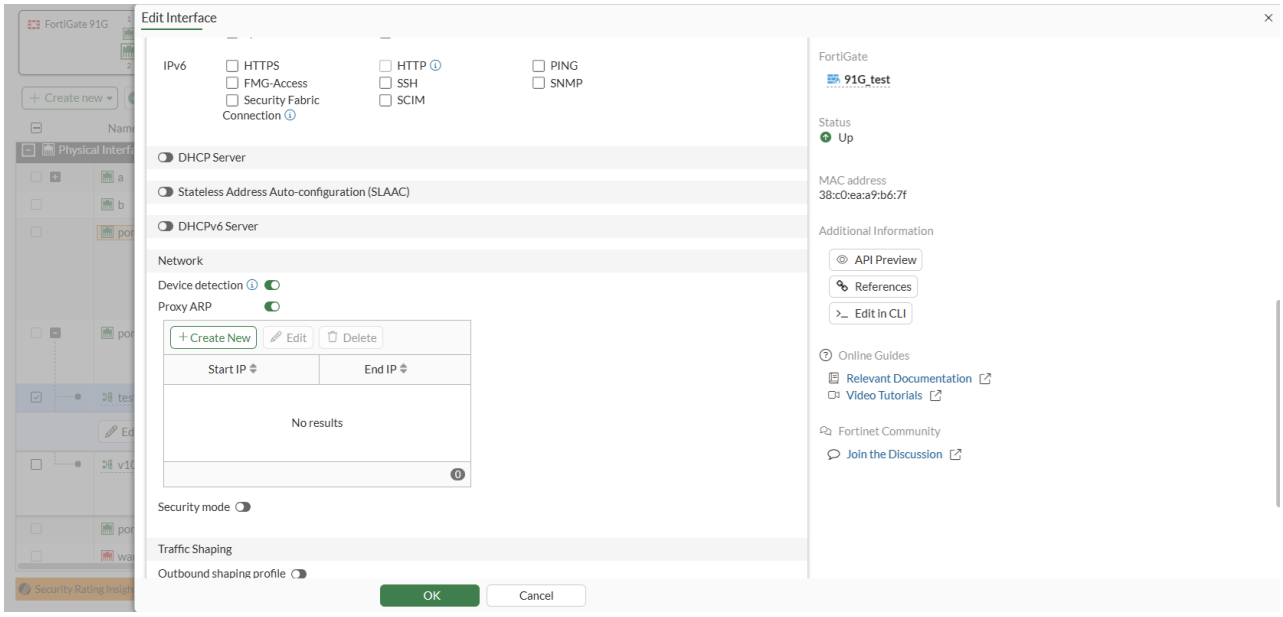
GUI support for configuring proxy ARP on VLAN interface

To facilitate configuring proxy ARP address ranges for FortiSwitch VLANs, the GUI now supports adding proxy arp address ranges for VLAN type interfaces.

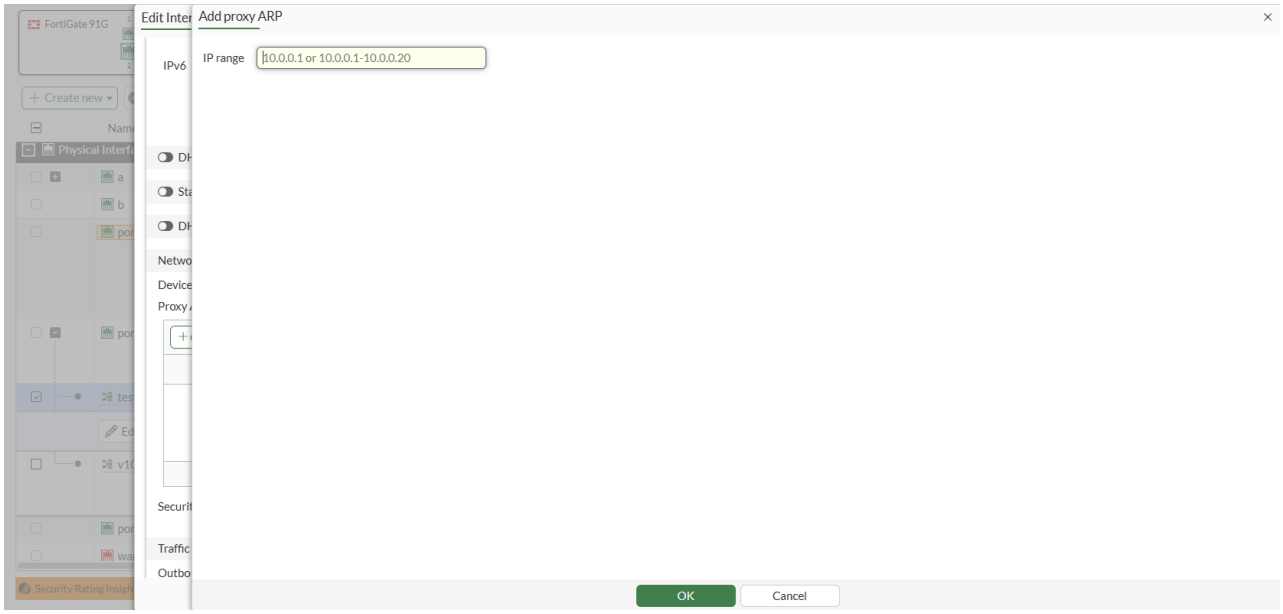
In a micro-segmentation scenario, you can enable blocking intra-VLAN traffic on a FortiSwitch and utilize proxy ARP and firewall policy to define which hosts are allowed to respond to each other. See Blocking intra-VLAN traffic.

Administrators can use the proxy ARP configuration option in a VLAN interface to create, edit, and delete these address ranges.

1. Go to *Network* > *Interfaces* and edit a VLAN interface to enable Proxy ARP.



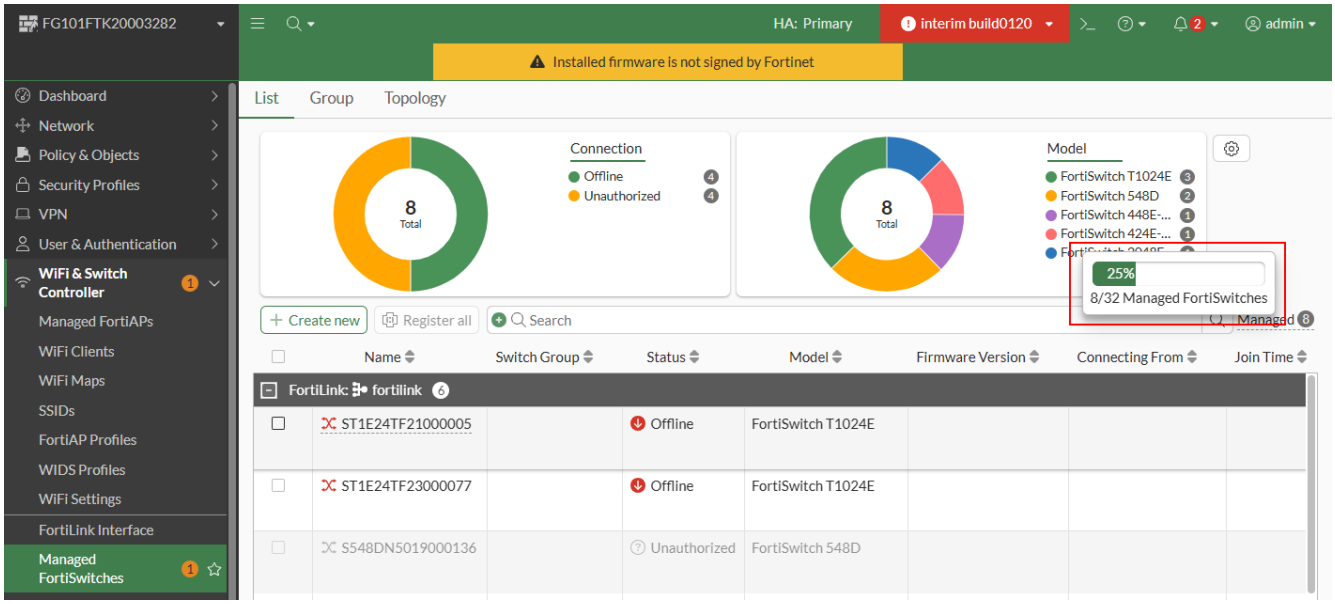
2. Click *Create New* in the *Proxy ARP* table to add or edit an IP address range.



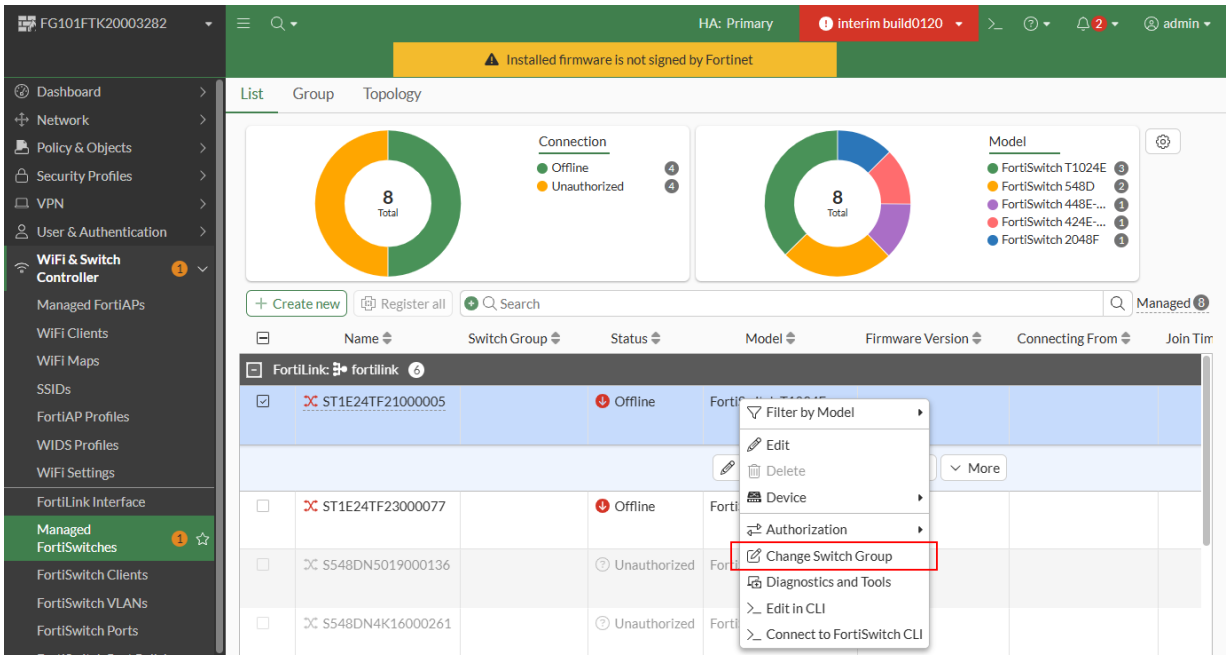
Improve switch management

The FortiGate GUI has been improved to make switch management easier.

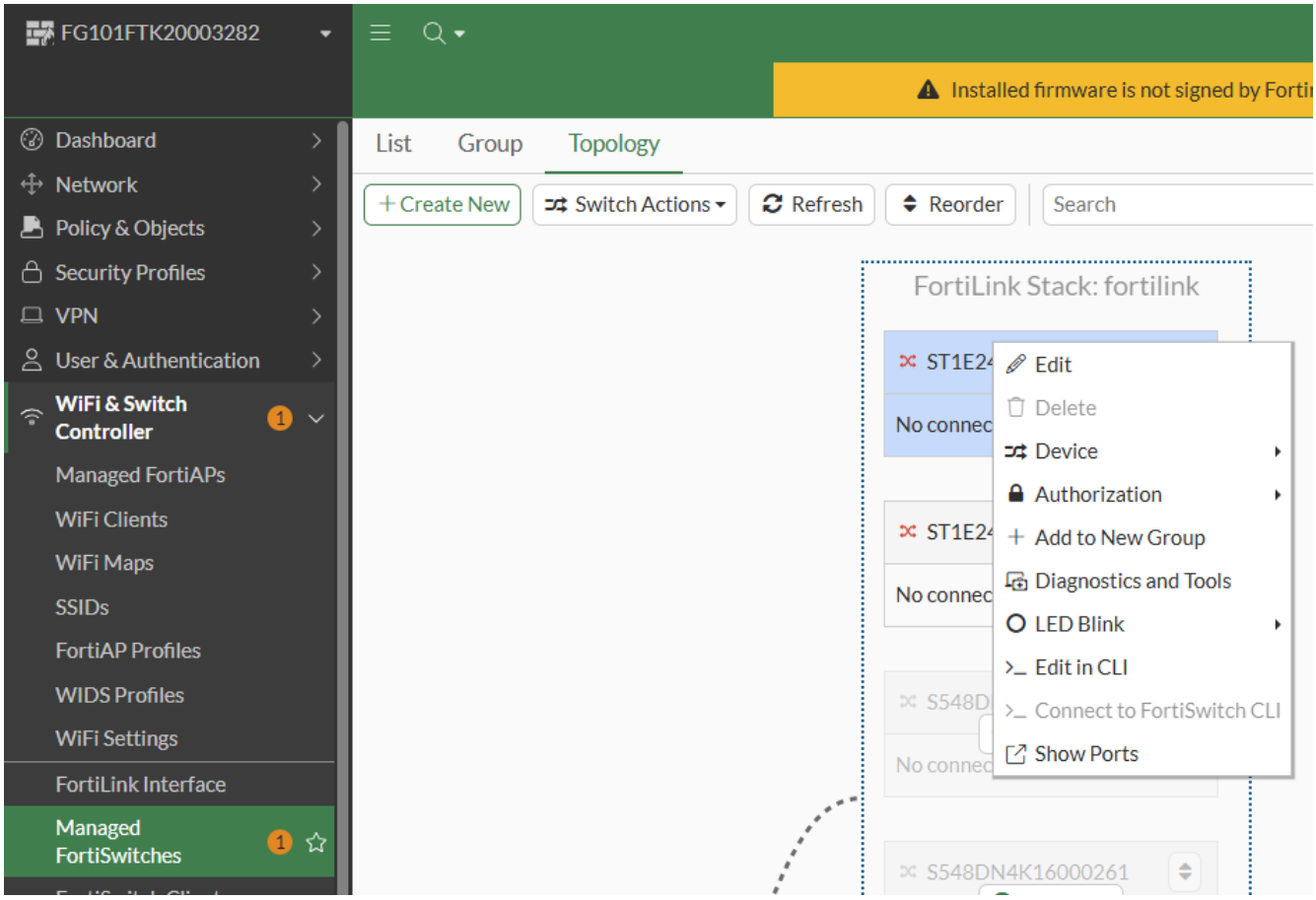
A tooltip in the *WiFi & Switch Controller page* > *Managed FortiSwitches* page shows the current number of managed FortiSwitch units and the maximum number supported of managed FortiSwitch units.



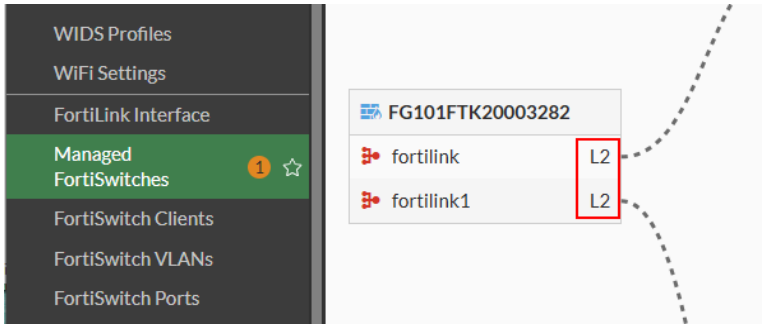
You can now change the group that the managed switch belongs by right-clicking on the switch in the *WiFi & Switch Controller page* > *Managed FortiSwitches* page.



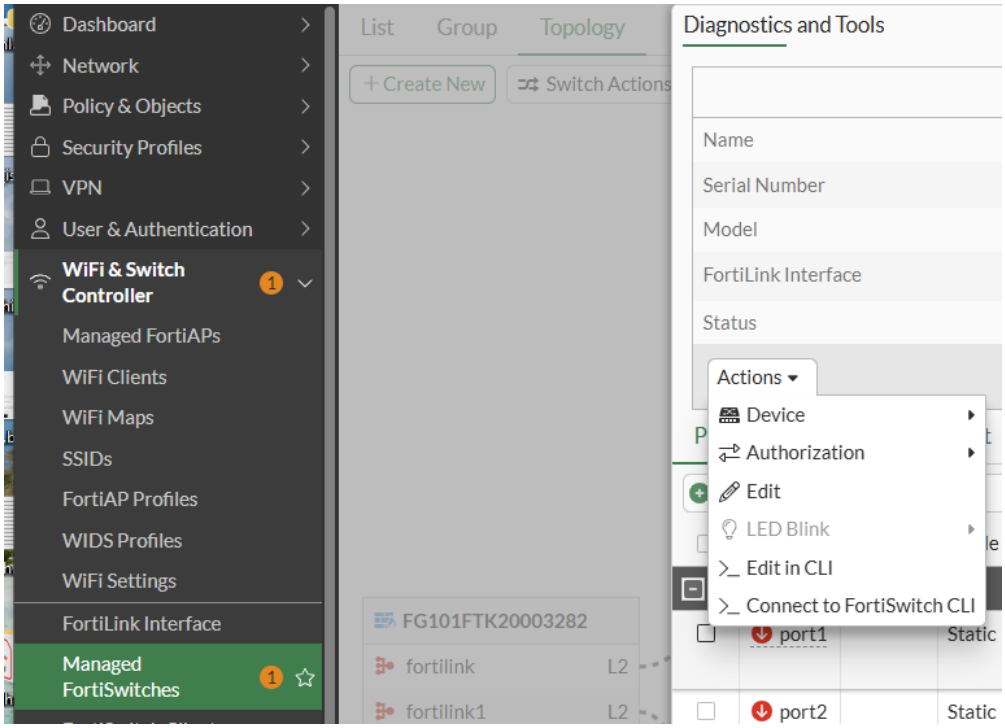
You now have the same commands available when you right-click a switch in the *Topology* view as in the *List* view.



You can now see the type of FortiLink connection in the *Topology* view.



You now have more commands available in the *Actions* dropdown menu in the *Diagnostics and Tools* pane.



When there is only one FortiLink interface, there is no longer an option to select the FortiLink interface in the *Diagnostics and Tools* pane.

Display Known Exploited Vulnerabilities from FortiClient

This information is also available in the FortiOS 8.0 Administration Guide:
- KEV information from FortiClient

FortiGate can now detect and display CISA Known Exploited Vulnerabilities (KEV) for FortiClient devices managed by FortiClient EMS in the *Security Fabric > Asset Identity Center* and the *Dashboard > Assets & Identities > Assets* widget in the GUI. Drilling down displays vulnerability details.

When FortiClient endpoints managed by FortiClient EMS run vulnerability scans and detect vulnerabilities, this information is synchronized to EMS and subsequently to FortiGate. When FortiGate receives detected vulnerabilities, it looks up the vulnerabilities against the KEV database and flags them accordingly. As a result, the FortiOS GUI can display the number of KEVs found on an endpoint and provide details about the vulnerabilities. The security rating check for FortiClient vulnerabilities also fails when a KEV vulnerability is detected.

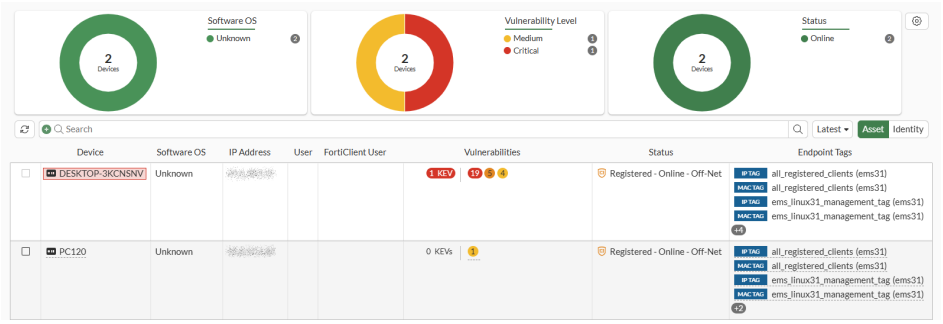
A new diagnose command is available:

```
# diagnose user-device-store device memory FortiClient-vulnerability-query <mac> <ip>
```

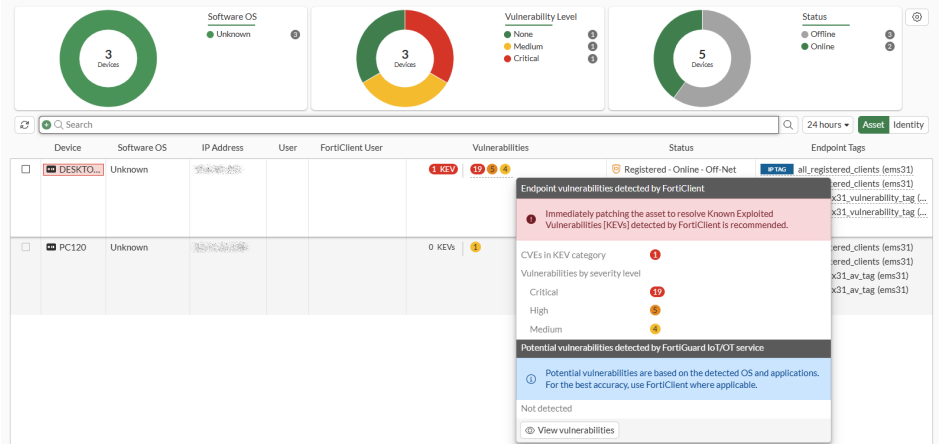
Option	Description
`diagnose user-device-store device memory FortiClient-vulnerability-query <mac> <ip>`	Query FortiClient vulnerability information by MAC and IP address. Takes 0-3 argument(s).

To view information in Security Fabric > Asset Identity Center:

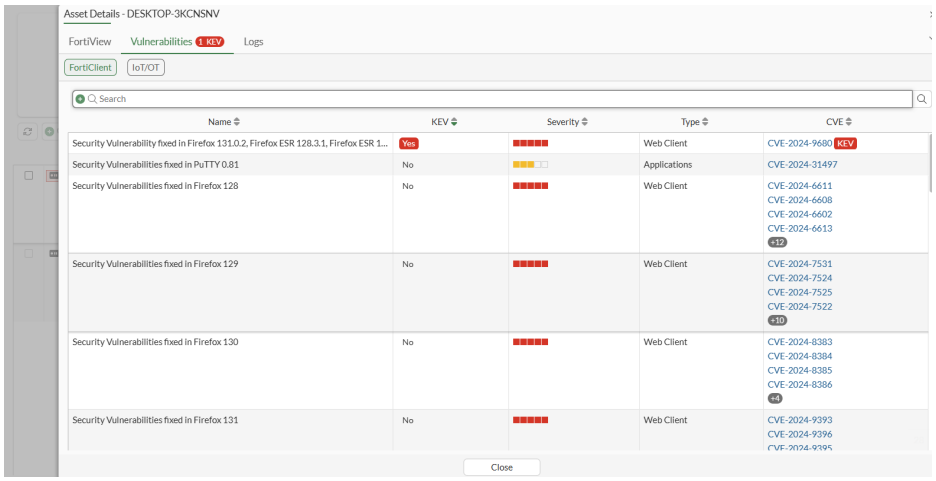
1. Go to *Security Fabric > Asset Identity Center* and view the *Vulnerabilities* column for KEV information detected by FortiClient.



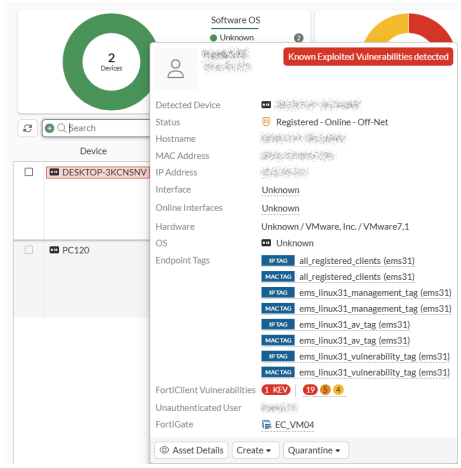
2. In the *Vulnerabilities* column, hover over a vulnerability to display a tooltip to view the KEV category and vulnerabilities by severity level.



3. Click *View vulnerabilities* to display additional information.

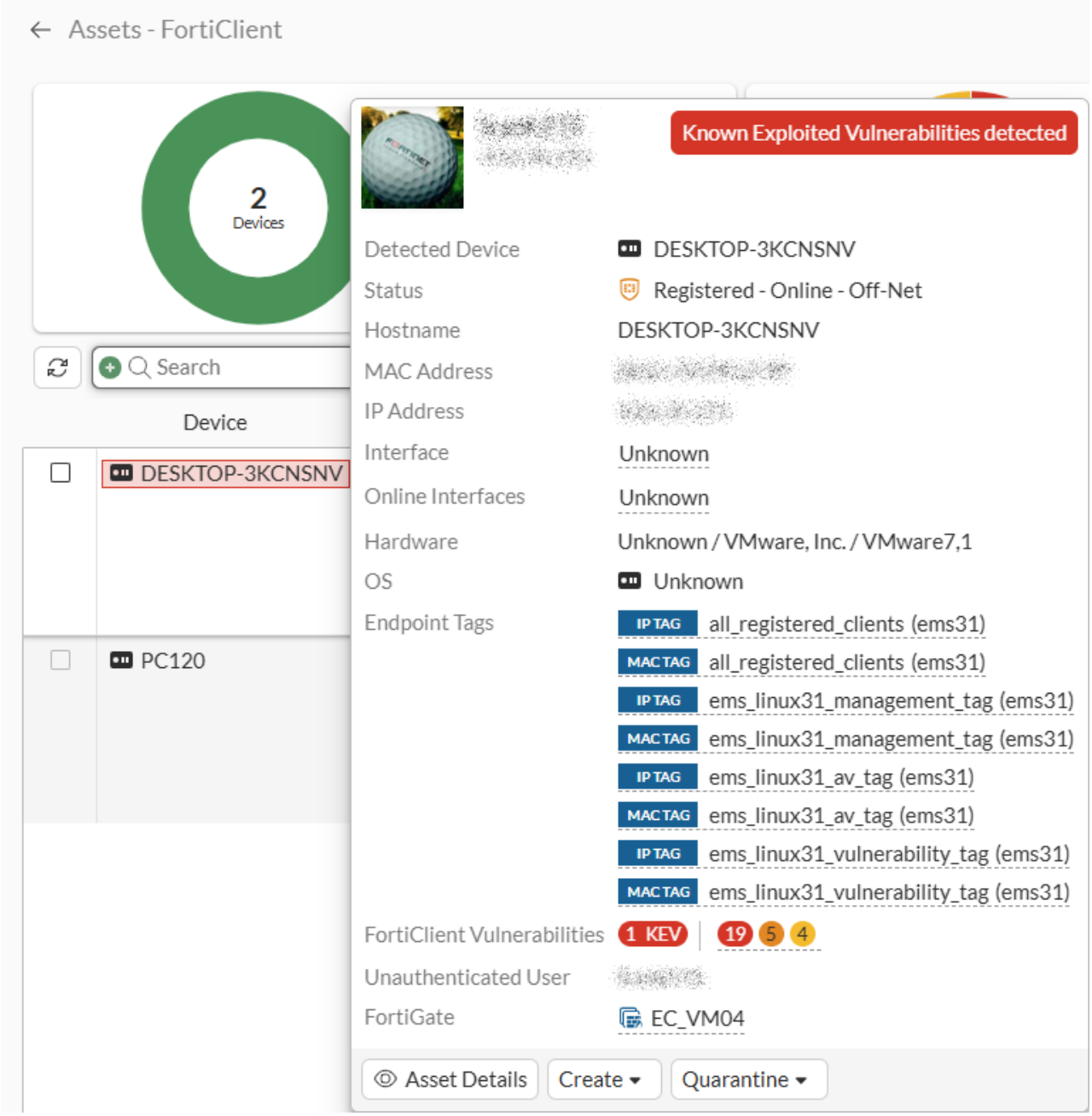


4. Click *Close*.
5. In the *Device* column, hover over a device name to display a tooltip of information about known, detected exploited vulnerabilities.

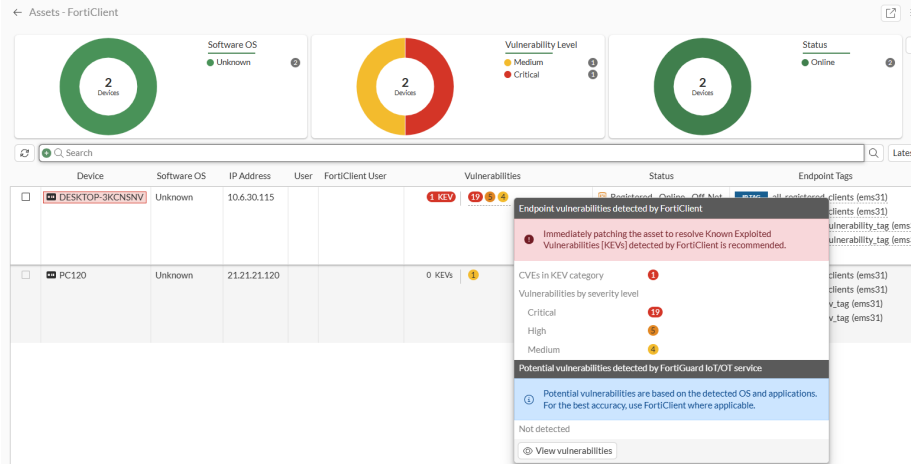


To view information in Dashboard > Assets & Identities:

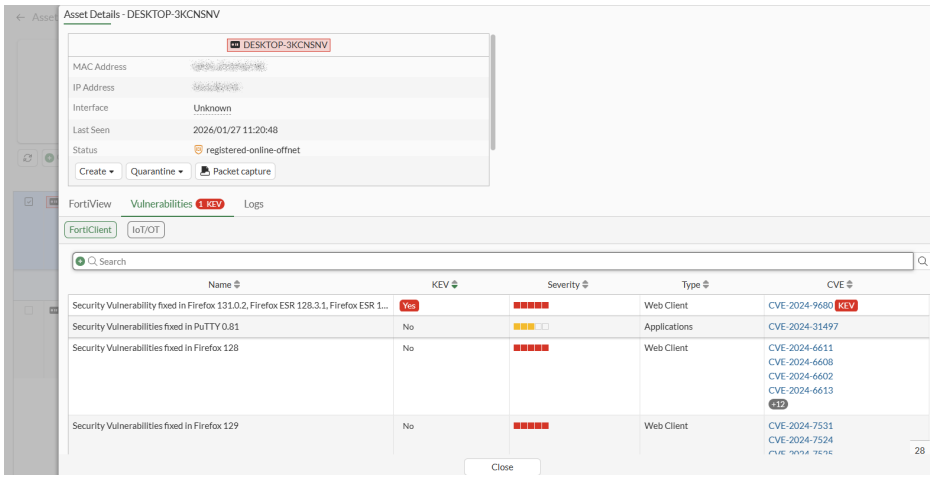
1. Go *to Dashboard > Assets & Identities > Assets - FortiClient* widget.
2. In the *Device* column, hover over a device name to display a tooltip of information about known, detected exploited vulnerabilities.



3. In the *Vulnerabilities* column, hover over a vulnerability to display a tooltip to view the KEV category and vulnerabilities by severity level.



4. Click *View vulnerabilities* to display additional information.



5. Click *Close*.

To query FortiClient vulnerability information by MAC and IP address:

```
# diagnose user-device-store device memory FortiClient-vulnerability-query <mac
address> <IP address>Record #1:

        fct_info
                'scan_time' = '1769477815'
        fct_vulnerability
                'vuln_id' = '5685'
                'severity' = '2'
                'category' = 'Applications'
                'vuln_name' = 'Security Vulnerabilities fixed in PuTTY 0.81'
                'cve_ids' = 'CVE-2024-31497'
        fct_vulnerability
                'vuln_id' = '80382'
                'severity' = '4'
                'category' = 'Web Client'
                'vuln_name' = 'Security Vulnerabilities fixed in Firefox 128'
                'cve_ids' = 'CVE-2024-6611'
```

```
                'cve_ids' = 'CVE-2024-6608'
                'cve_ids' = 'CVE-2024-6602'
                'cve_ids' = 'CVE-2024-6613'
                'cve_ids' = 'CVE-2024-6604'
                'cve_ids' = 'CVE-2024-6603'
                'cve_ids' = 'CVE-2024-6615'
                'cve_ids' = 'CVE-2024-6612'
                'cve_ids' = 'CVE-2024-6600'
                'cve_ids' = 'CVE-2024-6601'
                'cve_ids' = 'CVE-2024-6614'
                'cve_ids' = 'CVE-2024-6607'
                'cve_ids' = 'CVE-2024-6610'
                'cve_ids' = 'CVE-2024-6606'
                'cve_ids' = 'CVE-2024-6609'
                'cve_ids' = 'CVE-2024-6605'
        fct_vulnerability
                'vuln_id' = '80711'
                'severity' = '4'
                'category' = 'Web Client'
                'vuln_name' = 'Security Vulnerabilities fixed in Firefox 129'
                'cve_ids' = 'CVE-2024-7531'
                'cve_ids' = 'CVE-2024-7524'
                'cve_ids' = 'CVE-2024-7525'
                'cve_ids' = 'CVE-2024-7522'
                'cve_ids' = 'CVE-2024-7529'
                'cve_ids' = 'CVE-2024-7518'
                'cve_ids' = 'CVE-2024-7528'
                'cve_ids' = 'CVE-2024-7521'
                'cve_ids' = 'CVE-2024-7519'
                'cve_ids' = 'CVE-2024-7526'
                'cve_ids' = 'CVE-2024-7520'
                'cve_ids' = 'CVE-2024-7527'
                'cve_ids' = 'CVE-2024-7530'
                'cve_ids' = 'CVE-2024-7523'
        fct_vulnerability
                'vuln_id' = '81006'
                'severity' = '4'
                'category' = 'Web Client'
                'vuln_name' = 'Security Vulnerabilities fixed in Firefox 130'
                'cve_ids' = 'CVE-2024-8383'
                'cve_ids' = 'CVE-2024-8384'
                'cve_ids' = 'CVE-2024-8385'
                'cve_ids' = 'CVE-2024-8386'
                'cve_ids' = 'CVE-2024-8389'
                'cve_ids' = 'CVE-2024-8382'
                'cve_ids' = 'CVE-2024-8387'
                'cve_ids' = 'CVE-2024-8381'
        fct_vulnerability
                'vuln_id' = '81485'
                'severity' = '4'
                'category' = 'Web Client'
                'vuln_name' = 'Security Vulnerabilities fixed in Firefox 131'
```

```
                'cve_ids' = 'CVE-2024-9393'
                'cve_ids' = 'CVE-2024-9396'
                'cve_ids' = 'CVE-2024-9395'
                'cve_ids' = 'CVE-2024-9400'
                'cve_ids' = 'CVE-2024-9401'
                'cve_ids' = 'CVE-2024-9402'
                'cve_ids' = 'CVE-2024-9403'
                'cve_ids' = 'CVE-2024-9394'
                'cve_ids' = 'CVE-2024-9398'
                'cve_ids' = 'CVE-2024-9399'
                'cve_ids' = 'CVE-2024-9392'
                'cve_ids' = 'CVE-2024-9397'
                'cve_ids' = 'CVE-2024-9391'
        fct_vulnerability
                'vuln_id' = '81654'
                'severity' = '4'
                'category' = 'Web Client'
                'vuln_name' = 'Security Vulnerability fixed in Firefox 131.0.2, Firefox ESR
128.3.1, Firefox ESR 115.16.1'
                'cve_ids' = 'CVE-2024-9680'
                'kevs' = 'CVE-2024-9680'
        fct_vulnerability
                'vuln_id' = '81684'
                'severity' = '2'
                'category' = 'Web Client'
                'vuln_name' = 'Security Vulnerability fixed in Firefox 131.0.3'
                'cve_ids' = 'CVE-2024-9936'
```

Custom GUI themes and admin-level personalization

This information is also available in the FortiOS 8.0 Administration Guide:

- Setting GUI color themes

The *System* > *Settings* options are now organized across *General*, *Access*, *Device*, and *Preference* tabs. The *Preference* tab includes a *System theme* option to select a predefined theme or create a custom theme. Predefined and custom themes can then be associated with the FortiOS system, admin profiles, or individual administrators, enabling clearer environment distinction, improved usability, and stronger security awareness in multi-classification deployments.

Banner messages associated with a severity can be created for custom themes using the CLI.

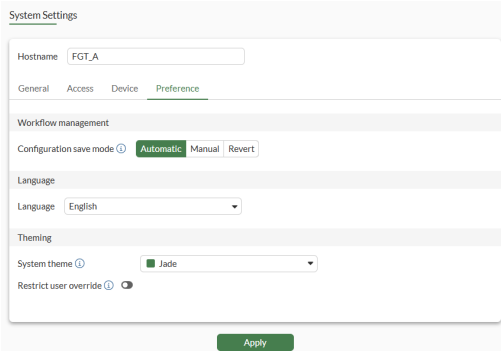
GUI

You can create one or more custom GUI themes for administrators. Then you can assign different predefined or custom themes to the system, admin profiles, and administrators.

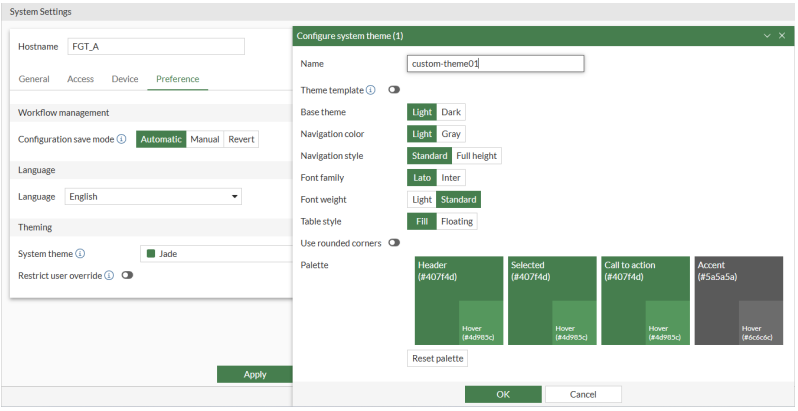
Administrators can also choose a theme.

To create a system theme in the GUI:

1. Go to *System* > *Settings* > *Preference*.



2. Beside *System theme*, click the list, and select Create (+). The *Configure system theme* pane is displayed.



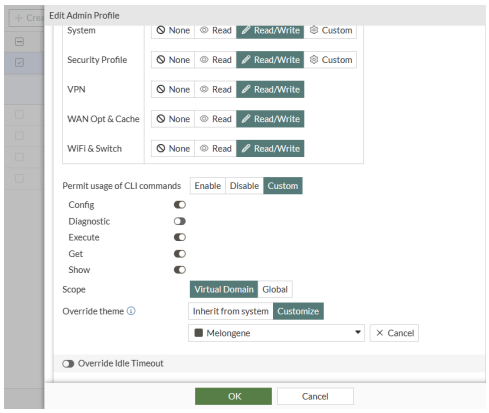
3. (Optional) Enable *Theme template* and select an existing template to customize.
4. Set the options, and click *OK*. The custom theme is selected.
5. (Optional) *Enable Restrict user override* to restrict theme customization to administrators assigned the super_admin profile.
6. Click *Apply* to apply the changes to the GUI.

To assign a theme to an administrator profile in the GUI:

1. Go to *System* > *Admin Profiles*.
2. Open an existing administrator profile for editing, or click *Create new* to create a new profile.
3. Set *Override Theme* to *Customize* and select a theme, or click Create (+) to create a custom theme.
4. Click *OK* to save the settings.
   The selected theme is used for administrators assigned this profile.

To assign a theme to individual administrators in the GUI:

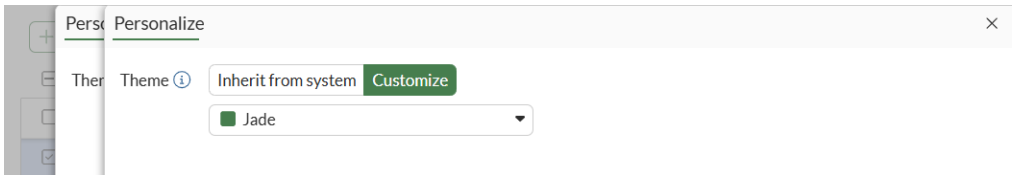
1. Go to *System* > *Administrators* and double-click an administrator to open it for editing.
2. Set *Override Theme* to *Customize* and select a theme, or click Create (+) to create a custom theme.



3. Click *OK* to save the settings.
   The selected theme is used for the administrator.

To personalize the GUI:

1. From the top-right corner, click your administrator name and select *Personalize*. The Personalize dialog box is displayed.



2. Set *Theme* to *Customize* and select a theme, or click Create (*+*) to create a custom theme.
3. Click *OK* to save the settings.
   The selected theme is used when you log in to FortiOS.

CLI

You can use the CLI to create one or more custom GUI themes. Then you can assign different predefined or custom themes to the FortiOS system, admin profiles, and administrators.

In the CLI, you can also create a banner message associated with a severity to display in a custom GUI theme.

To create a custom GUI theme in the CLI:

```
config system theme
    edit "custom theme 1"
        set nav-color gray
        set nav-style full-height
        set font inter
        set font-weight light
        set table-style float
        set border-radius enable
        set header-color "#e64747"
        set selected-color "#2cbaa2"
        set call-to-action-color "#ba2cb5"
```

```
        set accent-color "#df1616"
        set banner-msg ''
    next
end
```

To assign a custom theme to an administrator profile in the CLI:

```
config system accprofile
    edit "prof_admin"
        set gui-theme custom
        set gui-custom-theme custom theme 1
    next
end
```

To assign a custom theme to individual administrators in the CLI:

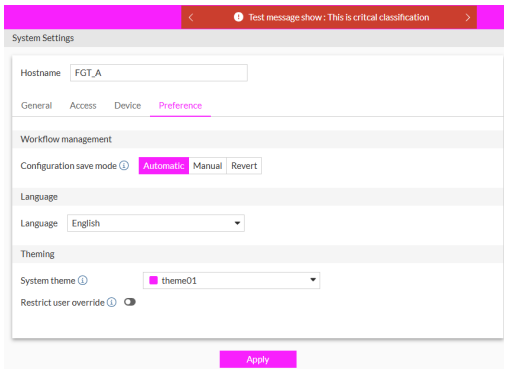
```
config system admin
    edit "admin"
        set gui-theme custom
        set gui-custom-theme custom theme 1
    next
end
```

To configure a banner for a custom theme in the CLI:

1. Configure the banner message and severity:

   ```
   config system theme
       edit theme01
           set banner-msg "Test message show: This is critical classification"
           set banner-msg-severity critical
       next
   end
   ```

   The message displays in the banner for the custom theme:



CLI syntax

A new command is available to configure custom GUI themes:

```
config system theme
    edit <name>
        set theme-template {jade | neutrino | mariner | graphite | melongene | jet-stream |
security-fabric | retro | dark-matter | onyx | eclipse | none}
        set base-theme {light | dark}
        set nav-color {light | gray}
        set nav-style {standard | full-height}
        set font {lato | inter}
        set font-weight {light | standard}
        set table-style {fill | float}
        set border-radius {enable | disable}
        set header-color <string>
        set selected-color <string>
        set call-to-action-color <string>
        set accent-color <string>
        set banner-msg <string>
        set banner-msg-severity {info | warning | critical}
    next
end
```

Option	Description
`config system theme`	Configure custom GUI themes.
`<name>`	Name of the theme.
`theme-template {jade | neutrino | mariner | graphite | melongene | jet-stream | security-fabric | retro | dark-matter | onyx | eclipse | none}`	Set the theme template to use in place of the base theme: • `jade`: Jade theme. • `neutrino`: Neutrino theme. • `mariner`: Mariner theme. • `graphite`: Graphite theme. • `melongene`: Melongene theme. • `jet-stream`: Jet stream theme. • `security-fabric`: Security Fabric theme. • `retro`: FortiOS v3 retro theme. • `dark-matter`: Dark matter theme. • `onyx`: Onyx theme. • `eclipse`: Eclipse theme. • `none`: No theme template.
`base-theme {light | dark}`	Set the base theme to use as a template: • `light`: Light theme. • `dark`: Dark theme.
`nav-color {light | gray}`	Set the color of the navigation bar: • `light`: Light colored navigation bar. • `gray`: Gray colored navigation bar.
`nav-style {standard | full-height}`	Set the navigation bar style: • `standard`: Use the standard navigation bar style.

Option	Description
	• `full-height`: Use the full-height navigation bar style.
`font {lato | inter}`	Set the font family: • `lato`: Use the Lato font. • `inter`: Use the Inter font.
`font-weight {light | standard}`	Set the font weight: • `light`: Use a thin font-weight. • `standard`: Use a standard font-weight.
`table-style {fill | float}`	Set the table style: • `fill`: Tables will fill the available space. • `float`: Tables will be separated by a margin.
`border-radius {enable | disable}`	Enable/disable border radius: • `enable`: Enable having curved borders for buttons and input fields. • `disable`: Disable having curved borders for buttons and input fields.
`header-color <string>`	Hexadecimal color code for the header color.
`selected-color <string>`	Hexadecimal color code for the selected color.
`call-to-action-color <string>`	Hexadecimal color code for the call to action color.
`accent-color <string>`	Hexadecimal color code for the accent color.
`banner-msg <string>`	Optional message to appear on the FortiGate header.
`banner-msg-severity {info | warning | critical}`	Severity color of the banner message: • `info`:Banner message on a blue background. Default severity. • `warning`: Banner message on a yellow background. • `critical`: Banner message on a red background.

A new parameter is available to assign a custom GUI theme to the system:

```
config system global
    set gui-custom-theme <string>
    set gui-restrict-theme-change {enable | disable}
end
```

Option	Description
`gui-custom-theme <string>`	Custom theme that overrides the default FortiGate themes.
`gui-restrict-theme-change {enable | disable}`	Enable/disable restricting editing and assigning override themes to super admins.

New parameters are available to assign a GUI theme to an admin profile:

```
config system accprofile
    edit <name>
```

```
        set gui-theme-type {predefined | custom}
        set gui-custom-theme <string>
        set gui-theme <string>
    next
end
```

Option	Description
`gui-theme-type {predefined | custom}`	Use predefined themes or custom themes: • `predefined`: Use predefined theme. • `custom`: Use custom theme
`gui-custom-theme <string>`	Name of the custom GUI theme. Available when `gui-theme-type` is set to `custom`.
`gui-theme <string>`	Set the predefined theme that overrides the default FortiGate theme. Available when `gui-theme-type` is set to `predefined`.

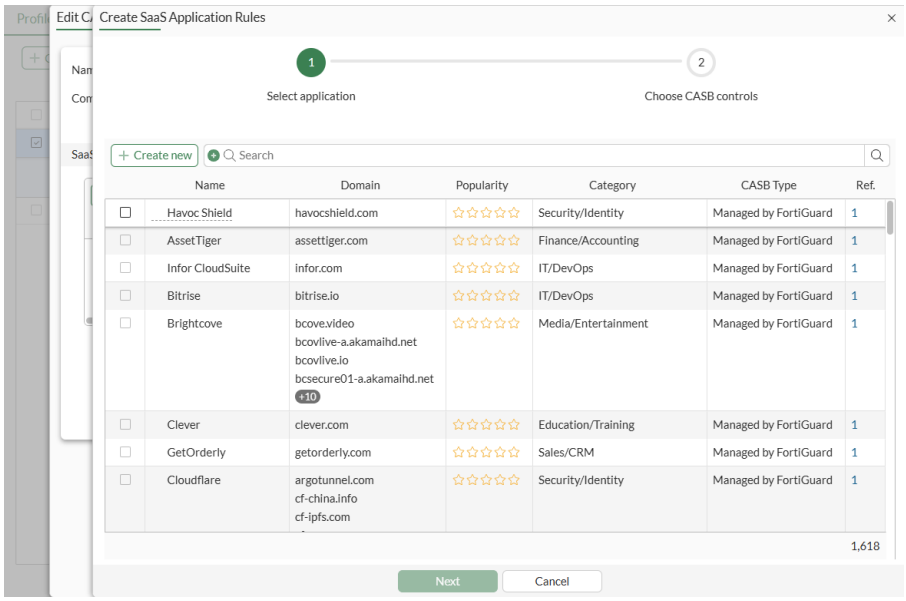
New parameters are available to assign a GUI theme to an administrator:

```
config system admin
    edit <name>
        set accprofile <name>
        set gui-theme-type {predefined | custom}
        set gui-custom-theme <string>
        set gui-theme <string>
    next
end
```

Option	Description
`gui-theme-type {predefined | custom}`	Use predefined themes or custom themes: • `predefined`: Use predefined theme. • `custom`: Use custom theme
`gui-custom-theme <string>`	Name of the custom GUI theme. Available when `gui-theme-type` is set to `custom`.
`gui-theme <string>`	Set the predefined theme that overrides the default FortiGate theme. Available when `gui-theme-type` is set to `predefined`.

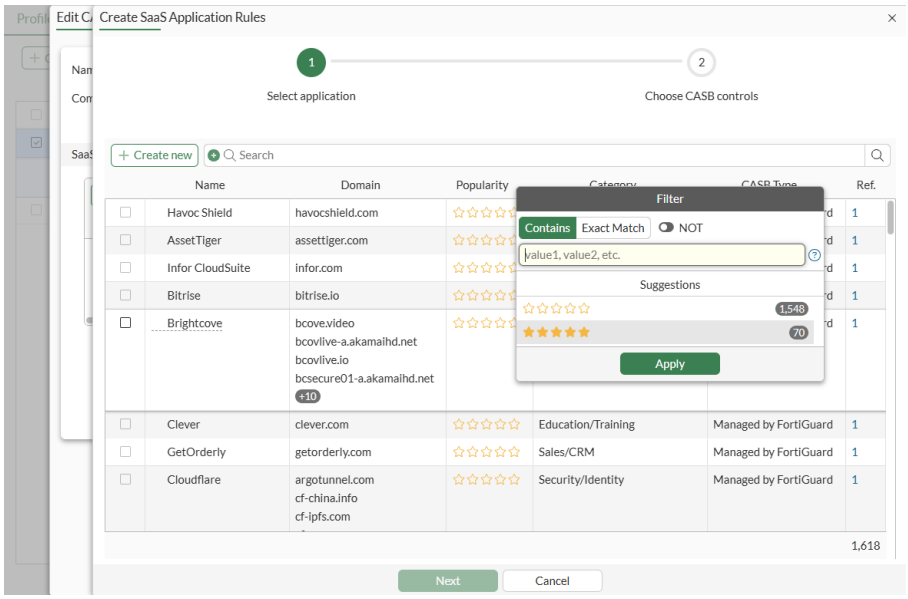
Enhanced GUI support for Inline-CASB SaaS Application

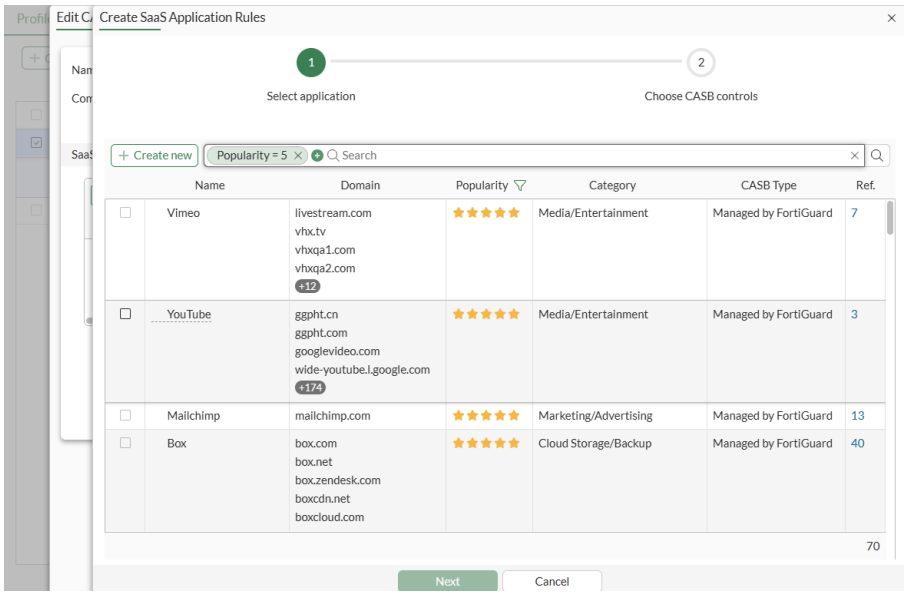
The SaaS application list now allow multi-select, bulk add actions, and includes columns and filters for *Category* and *Popularity*. These allow for faster CASB profile creation, especially when managing a large number of SaaS entries.



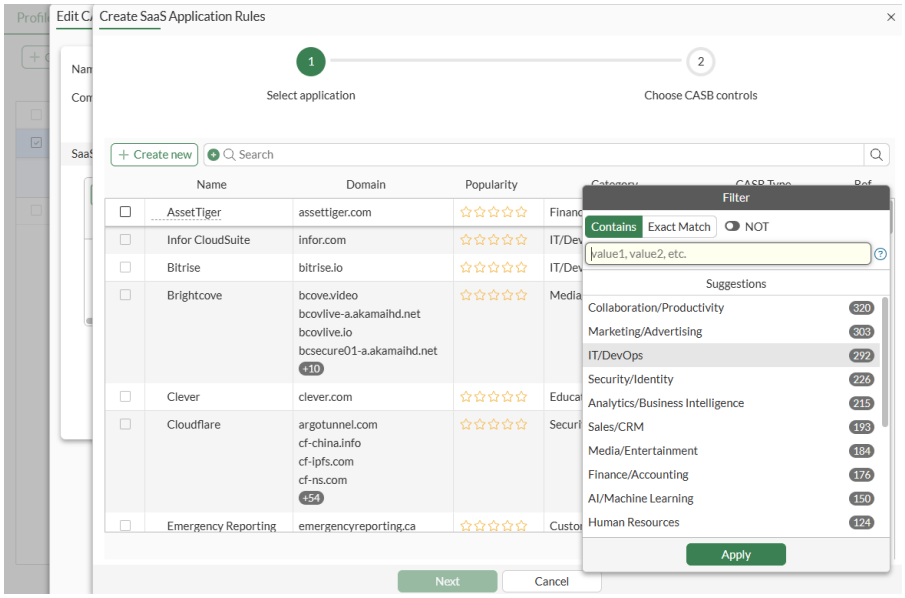
The list can be filtered by *Popularity* and *Category*:

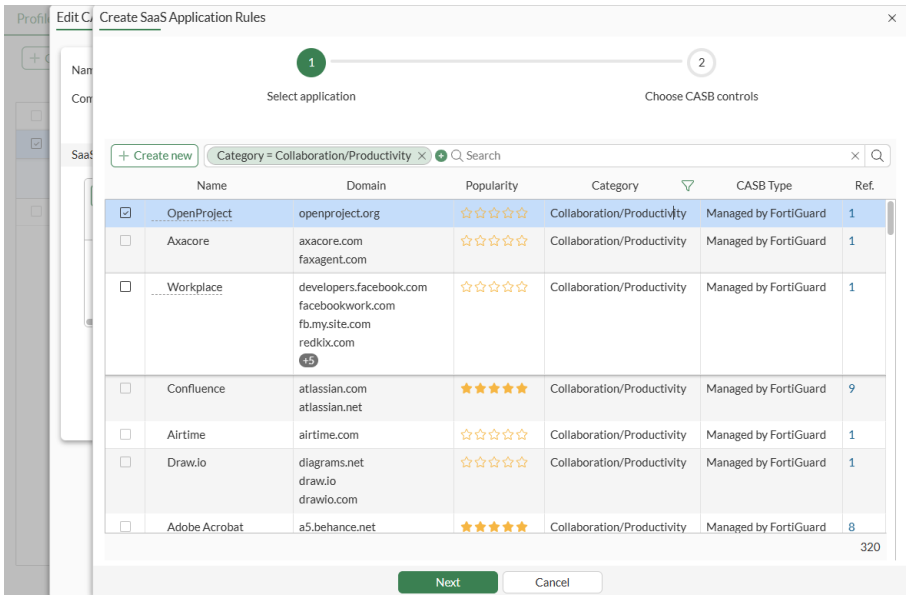
- Popularity




Create SaaS Application Rules
1
Select application
2
Choose CASB controls
+ Create new
Popularity = 5
Search
Name
Domain
Popularity
Category
CASB Type
Ref.
Vimeo
livestream.com
vhx.tv
vhxqa1.com
vhxqa2.com
+12
Media/Entertainment
Managed by FortiGuard
7
YouTube
ggpht.cn
ggpht.com
googlevideo.com
wide-youtube.l.google.com
+174
Media/Entertainment
Managed by FortiGuard
3
Mailchimp
mailchimp.com
Marketing/Advertising
Managed by FortiGuard
13
Box
box.com
box.net
box.zendesk.com
boxcdn.net
boxcloud.com
Cloud Storage/Backup
Managed by FortiGuard
40
70
Next
Cancel

- Category


Create SaaS Application Rules
1
Select application
2
Choose CASB controls
+ Create new
Search
Name
Domain
Popularity
AssetTiger
assettiger.com
Infor CloudSuite
infor.com
Bitrise
bitrise.io
Brightcove
bcove.video
bcovlive-a.akamaihd.net
bcovlive.io
bcsecure01-a.akamaihd.net
+10
Clever
clever.com
Cloudflare
argotunnel.com
cf-china.info
cf-ipfs.com
cf-ns.com
+54
Emergency Reporting
emergencyreporting.ca
Filter
Contains
Exact Match
NOT
value1, value2, etc.
Suggestions
Collaboration/Productivity 320
Marketing/Advertising 303
IT/DevOps 292
Security/Identity 226
Analytics/Business Intelligence 215
Sales/CRM 193
Media/Entertainment 184
Finance/Accounting 176
AI/Machine Learning 150
Human Resources 124
Apply
Next
Cancel



FortiView

This section includes new features related to FortiView:

- Graphical display for application performance monitoring analytics on page 35

Graphical display for application performance monitoring analytics

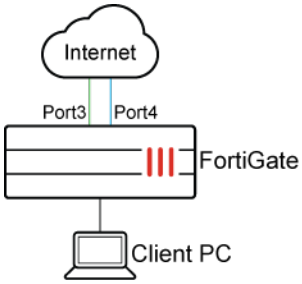
This information is also available in the FortiOS 8.0 Administration Guide:
- Application performance monitoring analytics

The new *FortiView SD-WAN Application Performance* widget leverages the passive WAN application performance monitoring of SD-WAN to log and monitor application performance metrics (see Application performance monitoring). These metrics are aggregated into five new FortiView report types:

- Application performance overview: Overview of different performance metrics of monitored applications.
- Application response time: Shows response time of a given application.
- Application connection stability: Shows application connection stability in terms of jitter.
- Application retransmission: Shows application traffic packet retransmission.
- Application reliability monitor: Shows packet loss and abrupt connection termination by TCP reset.

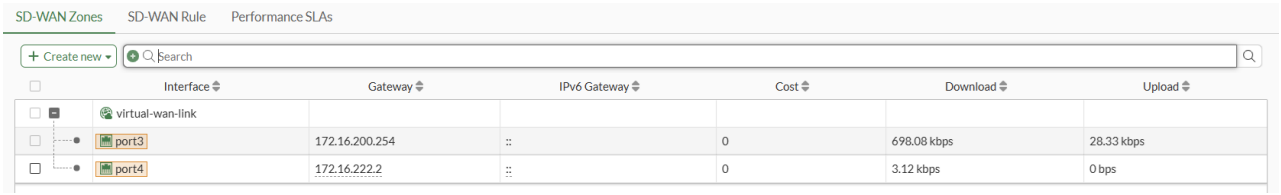
A new GUI option, *Log application health metrics*, is added to the firewall policy logging options section to enable the required logging for this widget.

Example



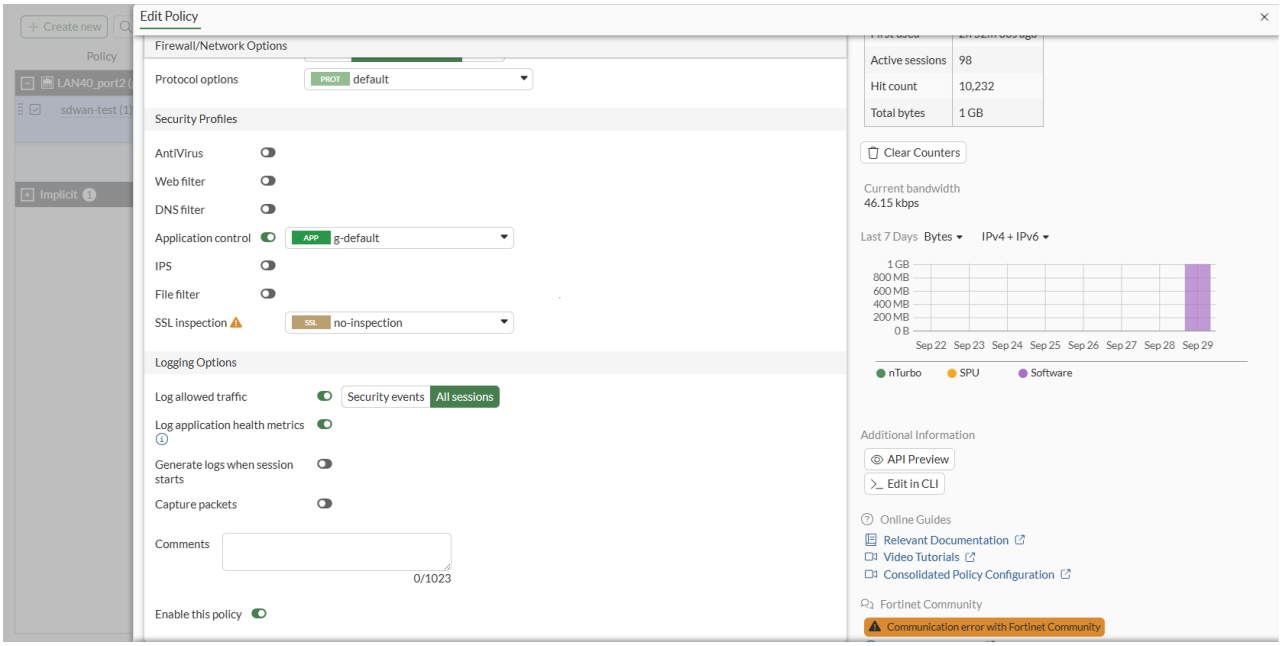
To test the application performance monitoring analytics in the GUI:

1. Create the SD-WAN zone and assign WAN ports.



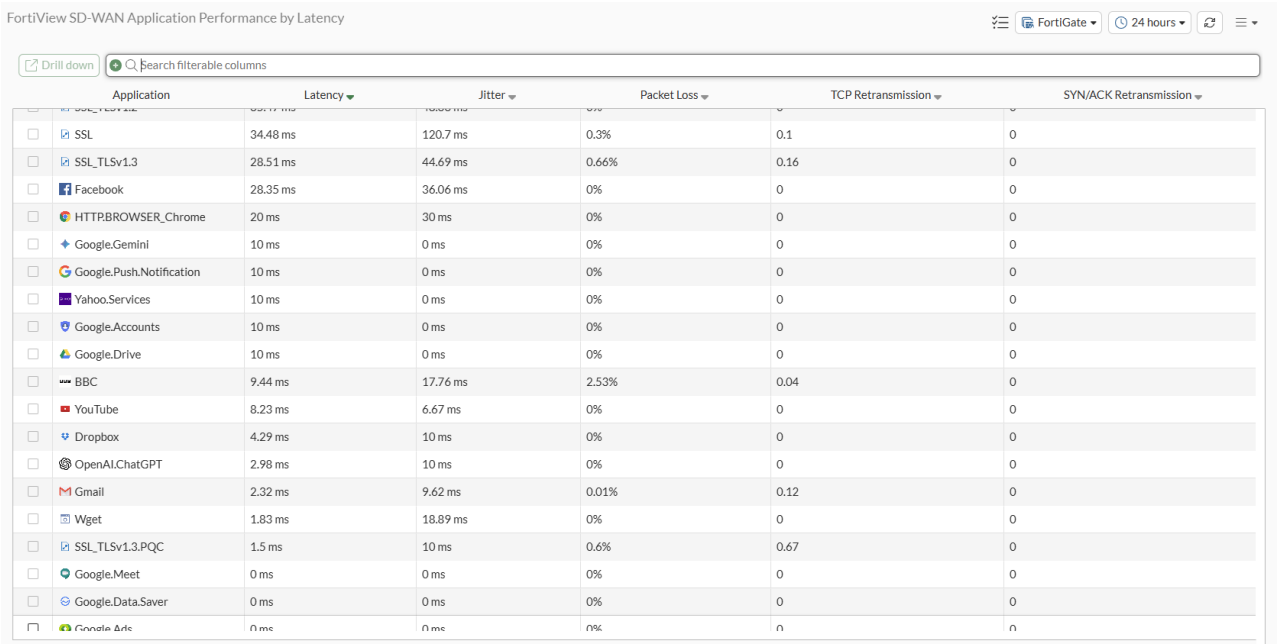
2. Create a policy with the following settings:
   - Set the *SD-WAN zone* as the destination interface.
   - Attach an *Application Control* profile to specify which applications are monitored.
   - Enable *Log application health metrics* under *Logging Options*.

     This replaces the *Passive health check* option under *Firewall > Network Options*. Enabling this toggle enables both *passive-wan-health-measurement* and *app-monitor* values in the policy. When it is disabled, both are turned off.



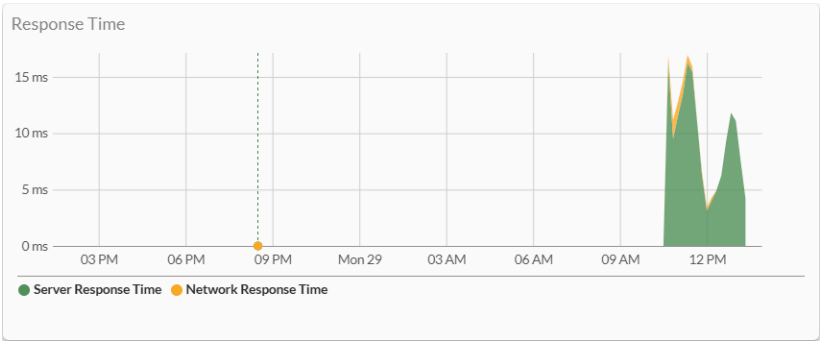
3. From a client PC, generate traffic using various applications, such as YouTube or ChatGPT.

4. Go to *Dashboard* > *FortiView* and add the *FortiView SD-WAN Application Performance Monitor*.
   Verify that filtering by *Applications* works and check that sorting by *Latency*, *Jitter*, *Packet Loss*, *TCP Retransmission*, and *SYN/ACK Retransmission* is functional.

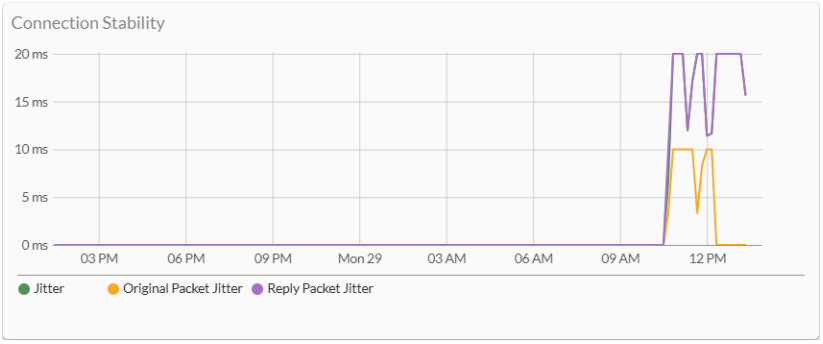


Application	Latency	Jitter	Packet Loss	TCP Retransmission	SYN/ACK Retransmission
SSL	34.48 ms	120.7 ms	0.3%	0.1	0
SSL_TLSv1.3	28.51 ms	44.69 ms	0.66%	0.16	0
Facebook	28.35 ms	36.06 ms	0%	0	0
HTTP.BROWSER_Chrome	20 ms	30 ms	0%	0	0
Google.Gemini	10 ms	0 ms	0%	0	0
Google.Push.Notification	10 ms	0 ms	0%	0	0
Yahoo.Services	10 ms	0 ms	0%	0	0
Google.Accounts	10 ms	0 ms	0%	0	0
Google.Drive	10 ms	0 ms	0%	0	0
BBC	9.44 ms	17.76 ms	2.53%	0.04	0
YouTube	8.23 ms	6.67 ms	0%	0	0
Dropbox	4.29 ms	10 ms	0%	0	0
OpenAI.ChatGPT	2.98 ms	10 ms	0%	0	0
Gmail	2.32 ms	9.62 ms	0.01%	0.12	0
Wget	1.83 ms	18.89 ms	0%	0	0
SSL_TLSv1.3.PQC	1.5 ms	10 ms	0.6%	0.67	0
Google.Meet	0 ms	0 ms	0%	0	0
Google.Data.Saver	0 ms	0 ms	0%	0	0

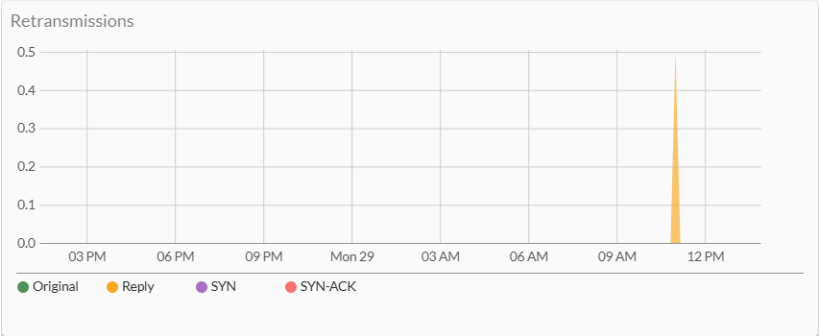
5. Drill down into an entry and verify that the following four charts are available:
   - **Response Time**



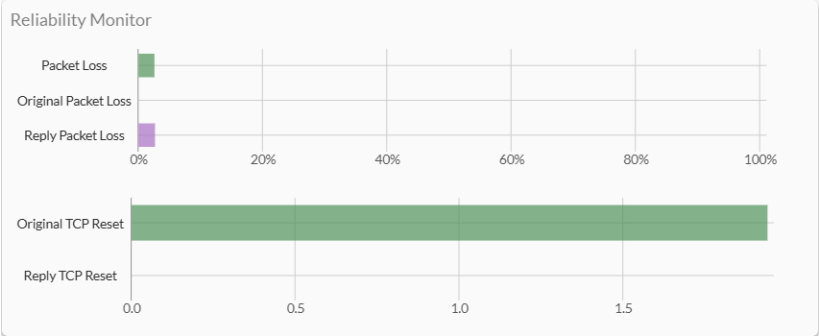
   - **Connection Stability**



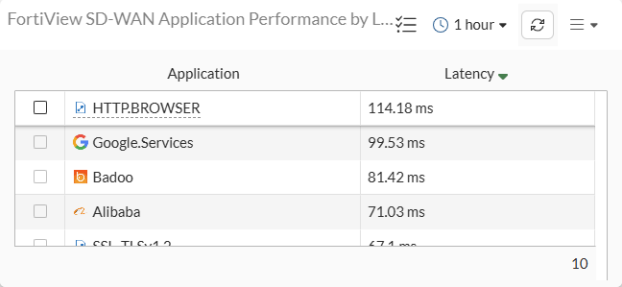
   - **Retransmissions**



- **Reliability Monitor**



6. Go to the Dashboard and confirm that the *FortiView SD-WAN Application Performance* widget can be added.



To test the application performance monitoring analytics in the CLI:

1. Create the SD-WAN zone and assign WAN ports:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port3"
            set gateway 172.16.200.254
        next
        edit 2
            set interface "port4"
```

```
            set gateway 172.16.222.2
        next
    end
end
```

2. Create a policy:

```
config firewall policy
    edit 1
        set name "sdwan-test"
        set srcintf "port2"
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set passive-wan-health-measurement enable
        set app-monitor enable
        set utm-status enable
        set application-list "g-default"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

3. Configure a health-check:

```
config system sdwan
    config health-check
        edit "test_h"
            set detect-mode passive
            set recoverytime 10
            set embed-measured-health enable
            set members 1 2
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 50
                next
            end
        next
    end
end
```

4. Configure an SD-WAN service and rule:

```
config system sdwan
    config service
        edit 1
            set name "sdwan_test"
```

```
            set mode sla
            set internet-service enable
            set internet-service-name "8X8-8X8.Cloud"
            set internet-service-app-ctrl 31077
            config sla
                edit "test_h"
                    set id 1
                next
            end
            set priority-zone "virtual-wan-link"
            set passive-measurement enable
        next
    end
end
```

5. From a client PC, generate traffic using YouTube.

Network

This section includes information about network related new features:

- General on page 41
- IPv6 on page 122
- Explicit and transparent proxy on page 124

General

This section includes information about general network related new features:

- IPv6 Multicast Routing Enhancement with BSR Support Added on page 41
- BGP neighbor naming support on page 48
- Increased AS path item for BGP route maps on page 51
- IPv6 PCP support for DNAT46 in NAT64 deployments on page 52
- Customizable CoS marking for locally generated ARP packets on page 55
- Firewall address support for IPAM rules on page 59
- Graceful BGP shutdown on page 64
- TCP congestion control enhancement with BBR on page 72
- Session helper statistics on page 74
- Enhanced EVPN support for VXLAN Anycast gateway and type-5 routing on page 75
- Enhanced PIM support for IPv6 across all VRFs on page 91
- GPON support for FortiGates with SFP/SFP+ ports on page 97
- 802.1X, MAB, dynamic VLAN, and address object support for software switch on page 99
- Dynamic BGP learning of ISDB reputation IPs on page 116
- Enhanced multicast session key for virtual wire pair on page 119
- Next-hop self-support for VPNv4 and VPNv6 route reflectors on page 121

IPv6 Multicast Routing Enhancement with BSR Support Added

FortiGate now supports BSR (Bootstrap Router) for IPv6 multicast, enabling dynamic RP discovery and aligning IPv6 multicast capabilities with existing IPv4 PIM-SM deployments. BSR is a mechanism in PIM-SM that automates the discovery and advertisement of Rendezvous Points (RPs), simplifying configuration and improving scalability for large-scale multicast environments.

New commands to support multicast flows for IPv6 to be used for static-group in PIM6 interface:

```
config router multicast6-flow
    edit <name>
        config flows
            edit <id>
                set *group-addr <IPv6 addr>
            next
        end
    next
end
```

Command or option	Description
`config router multicast6-flow`	Configure IPv6 multicast-flow.
`config flows`	IPv6 multicast-flow entries.
`id`	Flow ID.
`*group-addr <IPv6 addr>`	Multicast group IP address.

New commands to support PIM6 RP configurations for PIM6 interface:

```
config router multicast6
    config interface
        edit <name>
            set rp-candidate {enable | disable}
            set static-group <string>
            set rp-candidate-group <string>
            set rp-candidate-priority <value>
            set rp-candidate-interval <integer>
           next
    end
end
```

Command or option	Description
`config router multicast6`	Configure IPv6 multicast.
`config interface`	Protocol Independent Multicast (PIM) interfaces.
`rp-candidate {enable | disable}`	Enable/disable compete to become RP in elections: • `enable`: Compete for RP elections. • `disable`: Do not compete for RP elections.
`static-group <string>`	Statically set IPv6 multicast groups to forward out.
`rp-candidate-group <string>`	Multicast groups managed by this RP.
`rp-candidate-priority <value>`	Router's priority as RP (0 - 255, default = 192).
`rp-candidate-interval <integer>`	RP candidate advertisement interval (1 - 16383 sec, default = 60).

New commands to support PIM6 BSR configurations without VRF:

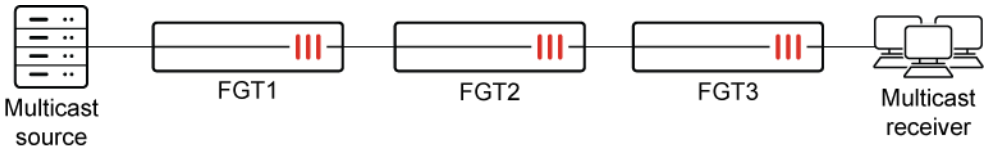
```
config router multicast6
    config pim-sm-global
        set bsr-candidate {enable | disable}
        set bsr-interface <string>
        set bsr-priority <integer>
        set bsr-hash <integer>
        set bsr-allow-quick-refresh {enable | disable}
        set cisco-crp-prefix {enable | disable}
        config rp-address
            edit <id>
                set *ip6-address <IPv6 addr>
                set group <string>
            end
        set cisco-ignore-rp-set-priority {enable | disable}
        set spt-threshold {enable | disable}
        set spt-threshould-group <string>
    end
end
```

Command or option	Description
`config pim-sm-global`	PIM sparse-mode global settings.
`bsr-candidate {enable | disable}`	Enable/disable allowing this router to become a bootstrap router (BSR).
`bsr-interface <string>`	Interface to advertise as candidate BSR.
`bsr-priority <integer>`	BSR priority (0 - 255, default = 0).
`bsr-hash <integer>`	BSR hash length (0 - 32, default = 10).
`bsr-allow-quick-refresh {enable | disable}`	Enable/disable accept BSR quick refresh packets from neighbors.
`cisco-crp-prefix {enable | disable}`	Enable/disable making candidate RP compatible with old Cisco IOS: • `enable`: Do not allow sending group prefix of zero. • `disable`: Allow sending group prefix of zero.
`config rp-address`	Statically configure RP addresses.
`*ip6-address <IPv6 addr>`	RP router IPv6 address.
`group <string>`	Groups to use this RP.
`cisco-ignore-rp-set-priority {enable | disable}`	Use only hash for RP selection (compatibility with old Cisco IOS): • `enable`: Ignore RP-SET priority value. • `disable`: Do not ignore RP-SET priority value.
`spt-threshold {enable | disable}`	Enable/disable switching to source specific tree: • `enable`:Switch to Source tree when available. • `disable`: Do not switch to Source tree when available.

Command or option	Description
`spt-threshould-group <string>`	Groups allowed to switch to source tree.

Example

This example about large-scale deployments shows how BSR (Bootstrap Router) can be used to dynamically learn the RP info and avoid mis-configurations.



In this example, three FortiGates are configured to support IPv6 multicast routing using PIM-SM. FGT1 and FGT2 are configured as RP and BSR candidates for IPv6 multicast groups, while port4 on FGT3 is connected to a multicast receiver:

- FGT1:

```
config router access-list6
    edit "list6"
        config rule
            edit 1
                set prefix6 ff05::10/128
            next
        end
    next
    edit "list11"
        config rule
            edit 1
                set prefix6 ff05::11/128
            next
        end
    next
end

config router multicast6
    set multicast-routing enable
    config interface
        edit "port1"
            set rp-candidate enable
            set rp-candidate-group "list11"
        next
        edit "port2"
            set rp-candidate enable
            set rp-candidate-group "list6"
            set rp-candidate-priority 200
        next
        edit "fortilink"
```

```
        next
    end
    config pim-sm-global
        set bsr-candidate enable
        set bsr-interface "port1"
    end
end
```

- FGT2

```
config router access-list6
    edit "list6"
        config rule
            edit 1
                set prefix6 ff05::10/128
            next
        end
    next
end

config router multicast6
    set multicast-routing enable
    config interface
        edit "port1"
            set rp-candidate enable
            set rp-candidate-group "list6"
        next
        edit "npu0_vlink0"
        next
    end
    config pim-sm-global
        set bsr-candidate enable
        set bsr-interface "port1"
    end
end
```

- FGT3

```
config router multicast6
    set multicast-routing enable
    config interface
        edit "npu0_vlink1"
        next
        edit "port4"
        next
    end
end
```

Verification of BSR Election and RP Mapping in IPv6 PIM Sparse Mode Across FortiGates:

- FGT1:

```
# get router info6 multicast pim sparse-mode bsr-router
PIMv2 Bootstrap information for VRF 0
  BSR address: 2000:172:16:200::4
  Uptime:      02:02:12, BSR Priority: 0, Hash mask length: 126
  Expires:     00:01:12
  Role: Candidate BSR
  State: Candidate BSR

  Candidate RP: 2000:172:16:205::1(port2)
    Advertisement interval 60 seconds
    Next Cand_RP_advertisement in 00:00:15

  Candidate RP: 2000:172:16:200::1(port1)
    Advertisement interval 60 seconds
    Next Cand_RP_advertisement in 00:00:02


# get router info6 multicast pim sparse-mode rp mapping
PIM Group-to-RP Mappings for VRF 0
Group(s): ff05::10/128
  RP: 2000:172:16:200::4
    Info source: 2000:172:16:200::4, via bootstrap, priority 192
         Uptime: 00:54:11, expires: 00:02:19
  RP: 2000:172:16:205::1
    Info source: 2000:172:16:200::4, via bootstrap, priority 200
         Uptime: 00:55:18, expires: 00:02:19
Group(s): ff05::11/128
  RP: 2000:172:16:200::1
    Info source: 2000:172:16:200::4, via bootstrap, priority 192
         Uptime: 00:55:18, expires: 00:02:19


# get router info6 multicast pim sparse-mode rp-hash ff05::11
    VRF 0 RP: 2000:172:16:200::1
    Info source: 2000:172:16:200::4, via bootstrap


# get router info6 multicast pim sparse-mode rp-hash ff05::10
    VRF 0 RP: 2000:172:16:200::4
    Info source: 2000:172:16:200::4, via bootstrap


# get router info6 multicast pim sparse-mode nexthop
Flags: N = New, R = RP, S = Source, U = Unreachable
VRF Destination          Type  Nexthop   Nexthop              Nexthop  Nexthop Metric Pref
Refcnt
                               Num     Addr                 Ifindex  Name
_____________________________________________________________________________________
  0 2000:172:16:200::1   .R..  1         2000:172:16:200::1   7               0      0     1
  0 2000:172:16:200::4   .RS.  1         2000:172:16:200::4   7               0      0     3
  0 2000:172:16:205::11  ..S.  1         2000:172:16:205::11  8               0      0     1
```

```
# get router info6 multicast pim sparse-mode mroute
IP Multicast Routing Table

(*,*,RP) Entries: 0
(*,G) Entries: 1
(S,G) Entries: 1
(S,G,rpt) Entries: 0
FCR Entries: 0

VRF 0 (2000:172:16:205::11, ff05::10)
RPF nbr: ::
RPF idx: None
SPT bit: 1
Upstream State: JOINED
 Local:
 Joined:
     port1
 Asserted:
 Outgoing:
     port1

VRF 0 (*, ff05::11)
RP: 2000:172:16:200::1
RPF nbr: ::
RPF idx: None
Upstream State: JOINED
 Local:
 Joined:
     port1
 Asserted:
FCR:
```

- FGT2

```
# get router info6 multicast pim sparse-mode bsr-router
PIMv2 Bootstrap information for VRF 0
This system is the Bootstrap Router (BSR)
  BSR address: 2000:172:16:200::4
  Uptime:      00:55:37, BSR Priority: 0, Hash mask length: 126
  Next bootstrap message in 00:00:28
  Role: Candidate BSR
  State: Elected BSR

  Candidate RP: 2000:172:16:200::4(port1)
    Advertisement interval 60 seconds
    Next Cand_RP_advertisement in 00:00:25


# get router info6 multicast pim sparse-mode rp mapping
PIM Group-to-RP Mappings for VRF 0
```

```
This system is the Bootstrap Router (v2)
Group(s): ff05::10/128
  RP: 2000:172:16:200::4
    Info source: 2000:172:16:200::4, via bootstrap, priority 192
         Uptime: 00:55:51, expires: 00:01:39
  RP: 2000:172:16:205::1
    Info source: 2000:172:16:205::1, via bootstrap, priority 200
         Uptime: 00:57:36, expires: 00:01:54
Group(s): ff05::11/128
  RP: 2000:172:16:200::1
    Info source: 2000:172:16:200::1, via bootstrap, priority 192
         Uptime: 00:57:49, expires: 00:01:41
```

- FGT3

```
# get router info6 multicast pim sparse-mode bsr-router
PIMv2 Bootstrap information for VRF 0
  BSR address: 2000:172:16:200::4
  Uptime:      02:04:31, BSR Priority: 0, Hash mask length: 126
  Expires:     00:01:53
  Role: Non-candidate BSR
  State: Accept Preferred

# get router info6 multicast pim sparse-mode rp mapping
PIM Group-to-RP Mappings for VRF 0
Group(s): ff05::10/128
  RP: 2000:172:16:200::4
    Info source: 2000:172:16:200::4, via bootstrap, priority 192
         Uptime: 00:56:21, expires: 00:02:09
  RP: 2000:172:16:205::1
    Info source: 2000:172:16:200::4, via bootstrap, priority 200
         Uptime: 00:57:28, expires: 00:02:09
Group(s): ff05::11/128
  RP: 2000:172:16:200::1
    Info source: 2000:172:16:200::4, via bootstrap, priority 192
         Uptime: 00:57:28, expires: 00:02:09
```

BGP neighbor naming support

BGP neighbors can now be assigned a name that appears alongside their IP address in the GUI and logs. This improves readability and simplifies peer identification in large or multi-tenant environments.

The `config neighbor` command includes a new option:

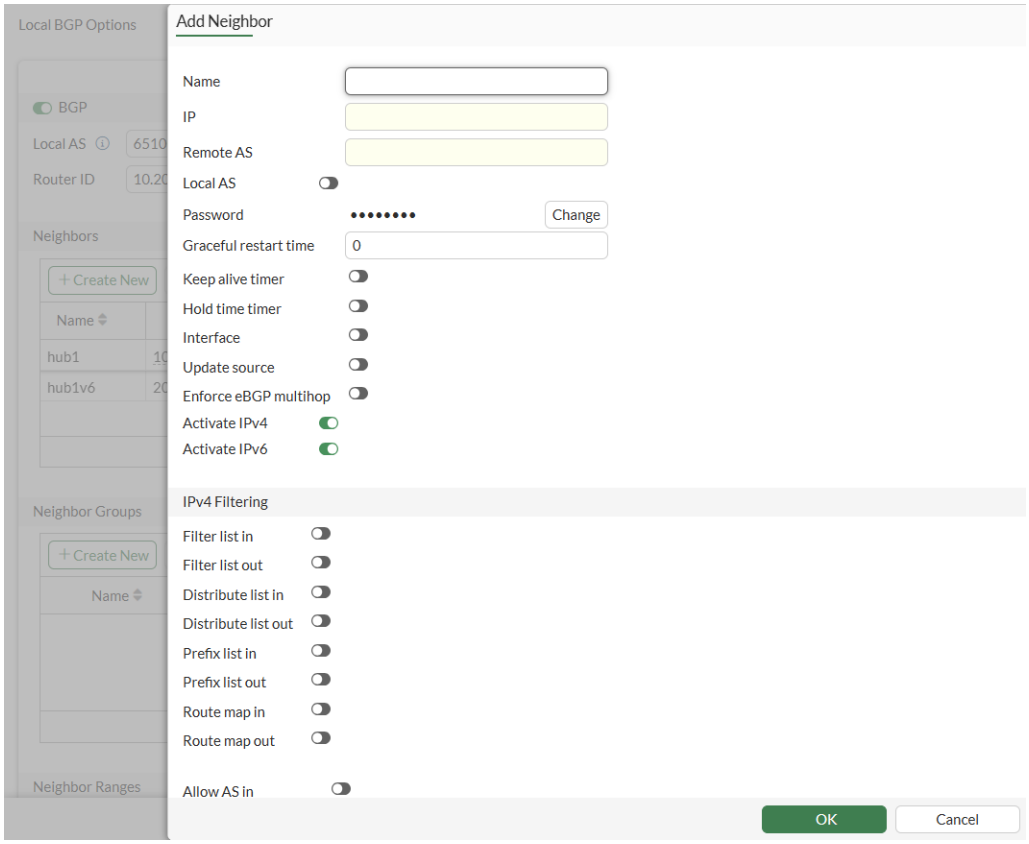
```
config neighbor
    edit <neighbor ID>
        set name <string>
        ...
    next
end
```

Option	Description
`name <string>`	Name of BGP neighbor.

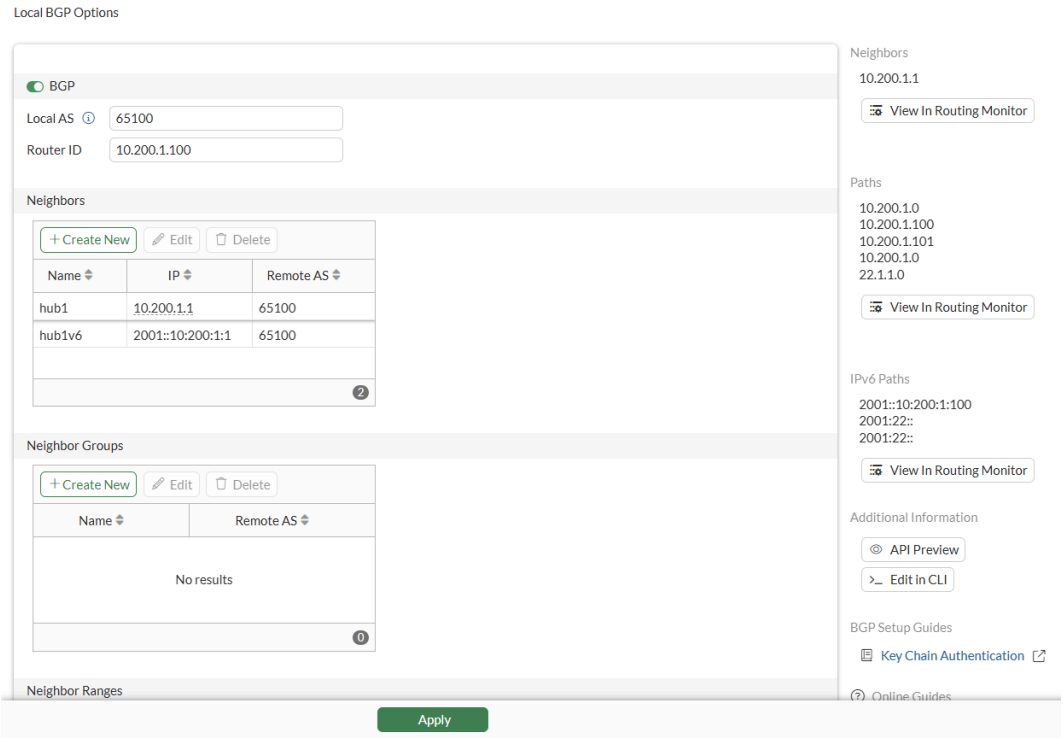
The GUI includes a new *Name* field.

To create a BGP neighbor with a name in the GUI:

1. Go to *Network* > *BGP*.
2. Under *Neighbors*, click *Create New*.
   The new *Name* field is available.



3. Enter a name for the BGP neighbor, set the remaining options as desired, and click *OK*.
   The name of the BGP neighbor (*hub1* and *hub1v6*) and its IP address is displayed in the GUI.



To create a BGP neighbor in the CLI:

```
config neighbor
    edit "10.200.1.1"
        set name "hub1"
        ...
    next
    edit "2001::10:200:1:1"
        set name "hub1v6"
        ...
    next
end
```

BGP neighbor names are included in some commands, such as:

```
# diagnose ip router command show-vrf vd2 show run | grep "router bgp" -A50

... ...
 bgp ignore-unknown-optional-capability
 neighbor 10.200.1.1 name "hub1"
 neighbor 10.200.1.1 remote-as 65100
 neighbor 10.200.1.1 update-source vd2-lo1
 neighbor 10.200.1.1 advertisement-interval 1
 neighbor 10.200.1.1 timers connect 10
 neighbor 10.200.1.1 capability dynamic
 neighbor 10.200.1.1 soft-reconfiguration inbound
 no neighbor 10.200.1.1 additional-path
 neighbor 10.200.1.1 capability graceful-restart
```

```
 neighbor 10.200.1.1 route-map tagvd2 in
 neighbor 2001::10:200:1:1 name "hub1v6"
 neighbor 2001::10:200:1:1 remote-as 65100
 neighbor 2001::10:200:1:1 update-source vd2-lo1
 neighbor 2001::10:200:1:1 advertisement-interval 1
 neighbor 2001::10:200:1:1 timers connect 10
 neighbor 2001::10:200:1:1 capability dynamic
 no neighbor 2001::10:200:1:1 activate
 no neighbor 2001::10:200:1:1 additional-path
```

Log files also include the names of BGP neighbors, for example:

```
date=2025-08-14 time=11:20:27 eventtime=1755195627191151900 tz="-0700" logid="0103020300"
type="event" subtype="router" level="warning" vd="vd2" logdesc="BGP neighbor status changed"
msg="BGP: %BGP-5-ADJCHANGE: VRF 0 neighbor 10.200.1.1(\"hub1\") Down BGP Notification FSM-ERR"
date=2025-08-14 time=11:20:27 eventtime=1755195626192462532 tz="-0700" logid="0103020300"
type="event" subtype="router" level="warning" vd="vd2" logdesc="BGP neighbor status changed"
msg="BGP: %BGP-5-ADJCHANGE: VRF 0 neighbor 10.200.1.1(\"hub1\") Up "
date=2025-08-14 time=11:20:22 eventtime=1755195621411138773 tz="-0700" logid="0103020300"
type="event" subtype="router" level="warning" vd="vd2" logdesc="BGP neighbor status changed"
msg="BGP: %BGP-5-ADJCHANGE: VRF 0 neighbor 10.200.1.1(\"hub1\") Down BGP Notification FSM-ERR"
date=2025-08-14 time=11:20:22 eventtime=1755195621411132437 tz="-0700" logid="0103020304"
type="event" subtype="router" level="warning" vd="vd2" logdesc="Routing log warning" msg="BGP:
%BGP-3-NOTIFICATION: received from 10.200.1.1 6/2 (Cease/Administratively Shutdown.) 0 data-bytes
[]"
```

Increased AS path item for BGP route maps

The maximum item for `set-aspath` in BGP route maps has been increased from 79 to 255 characters. This allows support for longer AS path prepending, enabling advanced self-healing configurations in large-scale networks.

The following example demonstrates how `set-aspath` allows repeated, multiple entries up to 255:

```
config router route-map
    edit "22"
        config rule
            edit 1
                set set-aspath 4294967295 4294967295 4294967295 4294967295 4294967295 4294967295
4294967295 4267295 4294967295 4294967295 4294967295 4294967295
                unset set-ip-prefsrc
            next
        end
    next
end
```

Multiple entries are also supported here:

```
get router info bgp network
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
VRF 0 BGP table version is 17, local router ID is 10.200.1.1
   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.200.1.0/24    0.0.0.0                           100  32768        0 i <0/1>
*>i10.200.1.100/32  10.200.1.100    0              100      0        0 4294967295 4294967295
4294967295 4294967295 4294967295 4294967295 4294967295 4267295 4294967295 4294967295 4294967295
4294967295 i <0/1>
*>i10.200.1.101/32  10.200.1.101    0              100      0        0 i <0/1>

Total number of prefixes 3
```

IPv6 PCP support for DNAT46 in NAT64 deployments

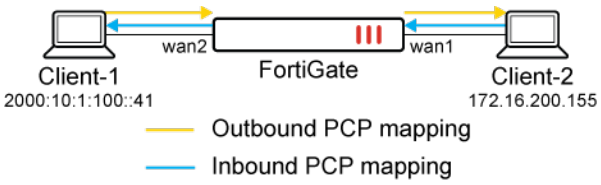
FortiOS now supports IPv6 PCP (Port Control Protocol) for DNAT46, enabling inbound PCP MAP requests in NAT64 deployments. This enhancement allows FortiGate to dynamically manage IPv6-to-IPv4 mappings through PCP, improving compatibility with CLAT-based CPEs.

The `config pools` command has new options:

```
config system pcp-server
    config pools
        edit <name>
            set client6-prefix <IPv6 client prefixes>
            set nat46 {enable | disable}
        next
    end
end
```

Option	Description
`client6-prefix <IPv6 client prefixes>`	Prefixes from which PCP requests are accepted.
`nat46 {enable | disable}`	Enable/disable NAT46.

Example



This example demonstrates how PCP IPv6-to-IPv4 mappings work with NAT64/DNAT46.

Client-1 has an IPv6 IP address, and Client-2 has an IPv4 address.

On FortiGate, a PCP server is configured with a PCP pool that has NAT46 enabled. The PCP pool is applied to a NAT46 firewall policy with `pcp-inbound` enabled and the pool specified.

Client-1 sends an IPv6 PCP request to FortiGate to request a PCP inbound mapping. FortiGate uses the inbound PCP mapping to DNAT translate the IPv4 IP address from Client-2 to the IPv6 IP address for Client-1.

To configure the FortiGate as a PCP server:

1. Configure the PCP server settings:

   The pool is named `35366_pool46_inbound`. NAT46 is enabled and an IPv6 prefix is defined. One IP/host subnet is used for both client and external IP.

```
config system pcp-server
   set status enable
   config pool
        edit "35366_pool46_inbound"
            set client-subnet "10.1.100.41/32"
            set client6-prefix "2000:10:1:100::41/128"
            set ext-intf "port1"
            set extip 172.16.200.121
            set extport 50000-51000
            set intl-intf "port9"
            set nat46 enable
        next
   end
end
```

2. Configure the firewall policy:

   In the firewall policy, `nat46` is enabled, `pcp-inbound` is enabled, and the PCP pool name (`35366_pool46_inbound`) is specified.

   The external interface (`ext-intf`) is port1, and the internal interface (`intl-intf`) is port9.

```
config firewall policy
    edit 46
        set name "pcp-46-inbound"
        set uuid de6c3632-95a6-51f0-0124-aa2705db9569
        set srcintf "port1"
        set dstintf "port9"
        set action accept
        set nat46 enable
        set custom-tags "test_tag_pcp_test"
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set pcp-inbound enable
        set pcp-poolname "35366_pool46_inbound"
        set ippool enable
        set poolname6 "pcp_ippool6_1"
```

```
    next
end
```

To verify the configuration:

1. Generate a PCP map request from Client-1 (IPv6 2000:10:1:100::41) to the FortiGate:

```
root@pc41:~# pcp -i [2000:10:1:100::41]:80 -s [2000:10:1:100::5]
```

2. On the FortiGate, verify the PCP inbound mappings list:

```
# diagnose firewall pcp-mapping list inbound
PCP inbound mappings (vdom=root):
pool:2 nonce:19c410963f68158746b735ba protocol:6 duration:809 lifetime:900 expiry:90
intl:2000:10:1:100::41:80 ext:172.16.200.121:50000
```

3. From Client-2 (IPv4 172.16.200.155), send traffic to pass through FortiGate and access the external interface (`extip`) of Client-1 (IPv4 172.16.200.121:50000).

   The traffic passes by the policy (policy id `46`), and the destination IP:port is DNAT translated from `172.16.200.231:50000` to `2000:10:1:100::41:80`, which follows the inbound pcp-mapping.

   Policy 46 only forwards traffic that matches the inbound PCP mapping (`pcp-mapping`). Other traffic is dropped.

```
# diagnose sniffer packet any 'tcp and port 80 or port 50000' 4
interfaces=[any]
filters=[tcp and port 80 or port 50000]
4.645467 port1 in 172.16.200.155.56926 -> 172.16.200.121.50000: syn 3388939279
4.645517 naf.root out 172.16.200.155.56926 -> 172.16.200.121.50000: syn 3388939279
4.645521 naf.root in 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: syn 3388939279
4.645557 port9 out 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: syn 3388939279
4.645866 port9 in 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: syn 3670124269 ack
3388939280  [flowlabel 0x60fc6]
4.645882 naf.root out 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: syn 3670124269 ack
3388939280  [flowlabel 0x60fc6]
4.645884 naf.root in 172.16.200.121.50000 -> 172.16.200.155.56926: syn 3670124269 ack
3388939280
4.645906 port1 out 172.16.200.121.50000 -> 172.16.200.155.56926: syn 3670124269 ack 3388939280
4.646006 port1 in 172.16.200.155.56926 -> 172.16.200.121.50000: ack 3670124270
4.646013 naf.root out 172.16.200.155.56926 -> 172.16.200.121.50000: ack 3670124270
4.646014 naf.root in 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: ack 3670124270
4.646020 port9 out 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: ack 3670124270
4.646138 port1 in 172.16.200.155.56926 -> 172.16.200.121.50000: psh 3388939280 ack 3670124270
4.646141 naf.root out 172.16.200.155.56926 -> 172.16.200.121.50000: psh 3388939280 ack
3670124270
4.646142 naf.root in 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: psh 3388939280 ack
3670124270
4.646144 port9 out 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: psh 3388939280 ack
3670124270
4.646227 port9 in 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: ack 3388939364  [flowlabel
0x60fc6]
4.646232 naf.root out 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: ack 3388939364
[flowlabel 0x60fc6]
```

```
4.646233 naf.root in 172.16.200.121.50000 -> 172.16.200.155.56926: ack 3388939364
4.646236 port1 out 172.16.200.121.50000 -> 172.16.200.155.56926: ack 3388939364
4.646587 port9 in 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: psh 3670124270 ack
3388939364  [flowlabel 0x60fc6]
4.646590 naf.root out 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: psh 3670124270 ack
3388939364  [flowlabel 0x60fc6]
4.646591 naf.root in 172.16.200.121.50000 -> 172.16.200.155.56926: psh 3670124270 ack
3388939364
4.646593 port1 out 172.16.200.121.50000 -> 172.16.200.155.56926: psh 3670124270 ack 3388939364
4.646668 port1 in 172.16.200.155.56926 -> 172.16.200.121.50000: ack 3670124531
4.646671 naf.root out 172.16.200.155.56926 -> 172.16.200.121.50000: ack 3670124531
4.646672 naf.root in 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: ack 3670124531
4.646675 port9 out 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: ack 3670124531
4.648833 port1 in 172.16.200.155.56926 -> 172.16.200.121.50000: fin 3388939364 ack 3670124531
4.648836 naf.root out 172.16.200.155.56926 -> 172.16.200.121.50000: fin 3388939364 ack
3670124531
4.648837 naf.root in 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: fin 3388939364 ack
3670124531
4.648839 port9 out 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: fin 3388939364 ack
3670124531
4.649053 port9 in 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: fin 3670124531 ack
3388939365  [flowlabel 0x60fc6]
4.649057 naf.root out 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: fin 3670124531 ack
3388939365  [flowlabel 0x60fc6]
4.649058 naf.root in 172.16.200.121.50000 -> 172.16.200.155.56926: fin 3670124531 ack
3388939365
4.649061 port1 out 172.16.200.121.50000 -> 172.16.200.155.56926: fin 3670124531 ack 3388939365
4.649138 port1 in 172.16.200.155.56926 -> 172.16.200.121.50000: ack 3670124532
4.649142 naf.root out 172.16.200.155.56926 -> 172.16.200.121.50000: ack 3670124532
4.649143 naf.root in 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: ack 3670124532
4.649146 port9 out 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: ack 3670124532
^C

40 packets received by filter
0 packets dropped by kernel
```

Customizable CoS marking for locally generated ARP packets

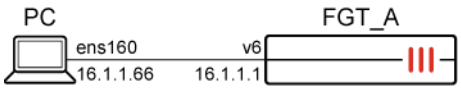
FortiGate supports 802.1p class of service (CoS) markings (cos0-cos7) for locally generated ARP packets, allowing customers to align ARP traffic with network QoS policies and constraints.

To set the CoS in VLAN tag for outgoing ARP packets:

```
config system interface
     edit <interface>
        set arp-egress-cos {cos0 | cos1 | cos2 | cos3 | cos4 | cos5 | cos6 | cos7}
```

```
    next
end
```

Example



A FortiGate and PC are both configured to be in VLAN66. The FortiGate VLAN interface is configured to send ARP packets with a CoS marking of 7, denoting the Network Control priority level.

To test the CoS markings in locally generated ARP packets:

1. Configure the interface on the FortiGate:

```
config system interface
     edit "v6"
        set vdom "root"
        set ip 16.1.1.1 255.255.255.0
        set allowaccess ping
        set arp-egress-cos cos7
        set snmp-index 40
        set ip-managed-by-fortiipam disable
        set interface "port6"
        set vlanid 66
    next
end
```

2. Configure the PC network:

```
~$ sudo cat /etc/netplan/01-network-manager-all.yaml
...
  vlans:
    vlan66:
      id: 66
      link: ens160
      addresses:
        - 16.1.1.66/24
```

3. Ping the FortiGate from the PC:

```
fosqa@fosqa-Cos:~$ ping 16.1.1.1 -c 2
PING 16.1.1.1 (16.1.1.1) 56(84) bytes of data.
64 bytes from 16.1.1.1: icmp_seq=1 ttl=255 time=0.193 ms
64 bytes from 16.1.1.1: icmp_seq=2 ttl=255 time=0.117 ms

--- 16.1.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1038ms
rtt min/avg/max/mdev = 0.117/0.155/0.193/0.038 ms
```

4. Do a packet capture on the PC:

An ARP request sent from the FortiGate sets its priority to `Network Control (7)` inside the VLAN header:

```
fosqa@fosqa-Cos:~$ sudo tshark -i ens160 -Y "vlan" -V
Running as user "root" and group "root". This could be dangerous.
Capturing on 'ens160'
...
Frame 2: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface ens160, id 0
    Section number: 1
    Interface id: 0 (ens160)
        Interface name: ens160
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 15, 2025 16:53:41.041847718 PDT
    UTC Arrival Time: Sep 15, 2025 23:53:41.041847718 UTC
    Epoch Arrival Time: 1757980421.041847718
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.000111729 seconds]
    [Time delta from previous displayed frame: 0.000111729 seconds]
    [Time since reference or first frame: 0.000111729 seconds]
    Frame Number: 2
    Frame Length: 64 bytes (512 bits)
    Capture Length: 64 bytes (512 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:vlan:ethertype:arp]
Ethernet II, Src: Fortinet_15:08:99 (94:f3:92:15:08:99), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
    Destination: Broadcast (ff:ff:ff:ff:ff:ff)
        Address: Broadcast (ff:ff:ff:ff:ff:ff)
        .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the
factory default)
        .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
    Source: Fortinet_15:08:99 (94:f3:92:15:08:99)
        Address: Fortinet_15:08:99 (94:f3:92:15:08:99)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: 802.1Q Virtual LAN (0x8100)
802.1Q Virtual LAN, PRI: 7, DEI: 0, ID: 66
    111. .... .... .... = Priority: Network Control (7)
    ...0 .... .... .... = DEI: Ineligible
    .... 0000 0100 0010 = ID: 66
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000
    Trailer: 00000000
Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Fortinet_15:08:99 (94:f3:92:15:08:99)
    Sender IP address: 16.1.1.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 16.1.1.66
```

An ARP reply sent by the FortiGate in response to the PC's ARP also sets its priority to `Network Control (7)`:

```
Frame 8: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface ens160, id 0
    Section number: 1
    Interface id: 0 (ens160)
        Interface name: ens160
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 15, 2025 16:53:46.239793531 PDT
    UTC Arrival Time: Sep 15, 2025 23:53:46.239793531 UTC
    Epoch Arrival Time: 1757980426.239793531
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.000082934 seconds]
    [Time delta from previous displayed frame: 0.000082934 seconds]
    [Time since reference or first frame: 5.198057542 seconds]
    Frame Number: 8
    Frame Length: 64 bytes (512 bits)
    Capture Length: 64 bytes (512 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:vlan:ethertype:arp]
Ethernet II, Src: Fortinet_15:08:99 (94:f3:92:15:08:99), Dst: VMware_7c:6c:08
(00:0c:29:7c:6c:08)
    Destination: VMware_7c:6c:08 (00:0c:29:7c:6c:08)
        Address: VMware_7c:6c:08 (00:0c:29:7c:6c:08)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: Fortinet_15:08:99 (94:f3:92:15:08:99)
        Address: Fortinet_15:08:99 (94:f3:92:15:08:99)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: 802.1Q Virtual LAN (0x8100)
802.1Q Virtual LAN, PRI: 7, DEI: 0, ID: 66
    111. .... .... .... = Priority: Network Control (7)
    ...0 .... .... .... = DEI: Ineligible
    .... 0000 0100 0010 = ID: 66
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000
    Trailer: 00000000
Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: Fortinet_15:08:99 (94:f3:92:15:08:99)
    Sender IP address: 16.1.1.1
    Target MAC address: VMware_7c:6c:08 (00:0c:29:7c:6c:08)
    Target IP address: 16.1.1.66
```

Firewall address support for IPAM rules

IPAM traditionally manages IP assignments directly on FortiGate interfaces by automatically configuring its IP, subnet, and DHCP based on network sizes. In this enhancement, IPAM can now manage IP pools for a Firewall Address object. IPAM type firewall addresses in turn can be configured on VPN gateways for automatic IP range allocation to VPN clients. This allows centralized IP management and traffic control across distributed subnets, improving scalability and operational efficiency.

FortiGate also supports per-device subnet allocation for Fabric object firewall addresses that use IPAM. A new option, `ipam-allocate-unique`, is introduced for firewall address objects configured with type `ipam` and `fabric-object enable`. When enabled, each FortiGate in the Security Fabric receives a unique subnet allocation from IPAM instead of inheriting the same subnet through CSF synchronization.

This enhancement allows greater flexibility in multi-device deployments where shared IPAM addresses across all FortiGates are not desirable.

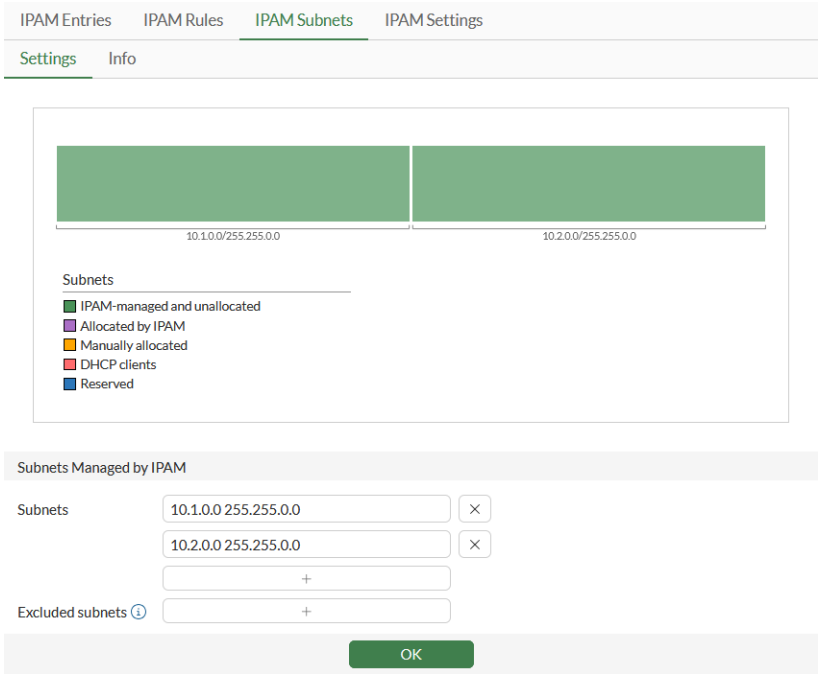
```
config firewall address
    edit "ipam"
        set type ipam
        set fabric-object enable
        set subnet 192.168.0.0 255.255.255.0
        set ipam-allocate-unique enable
    next
end
```

Example

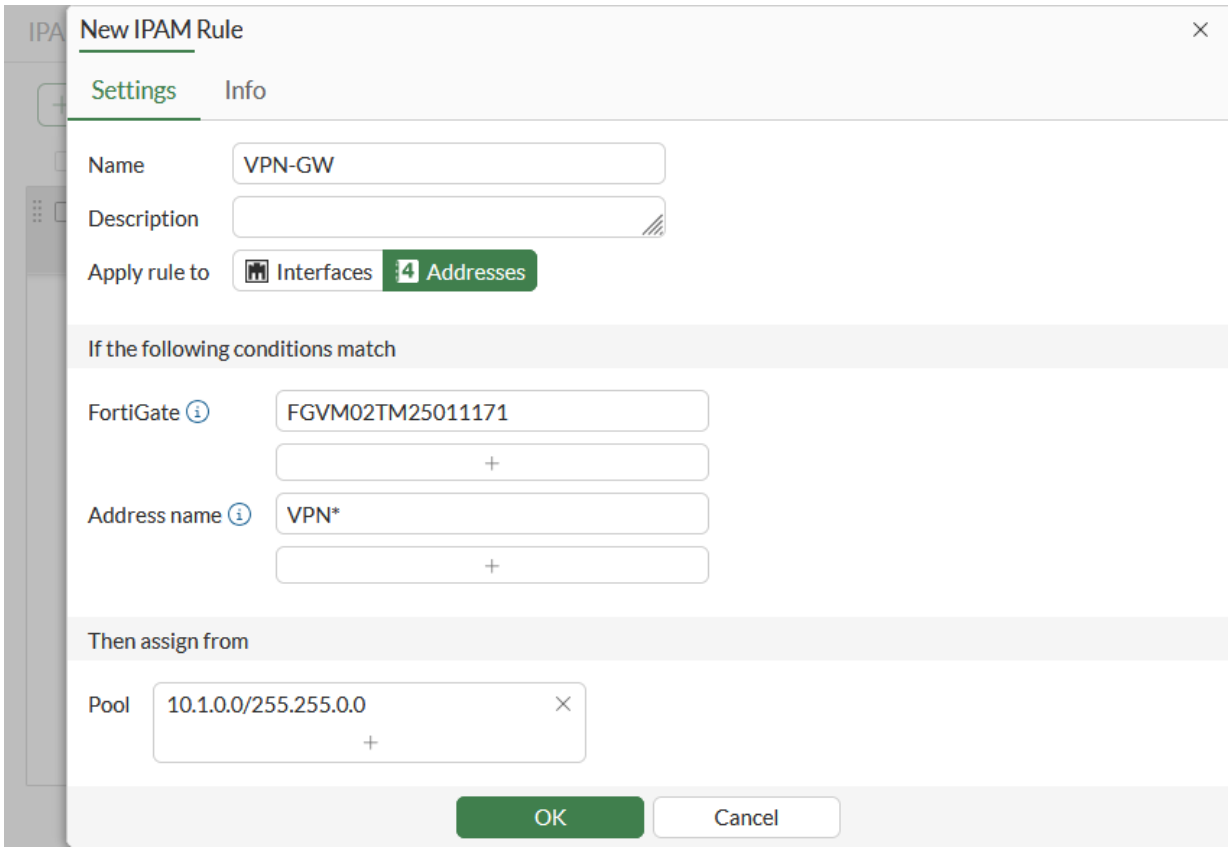
In this example, we have 2 IP range allocation on IPAM. One range will be dedicated to address objects with a name starting with VPN. The other range will be assigned to other address objects.

To configure an IPAM firewall address in the GUI:

1. Enable IPAM and create 2 IPAM subnets:
   a. Go to *Network* > *IPAM* > *IPAM Settings*.
   b. Set *Status* to *Enable* then click *OK*.
   c. Go to *Network* > *IPAM* > *IPAM Subnets*.
   d. Create 2 subnets *10.1.0.0 255.255.0.0* and *10.2.0.0 255.255.0.0*.
   e. Click *OK*.

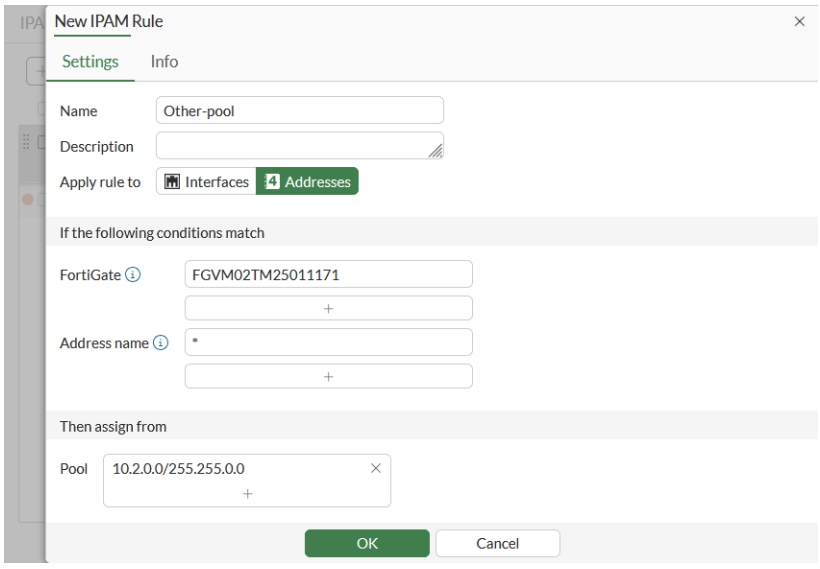


2. Create an IPAM rule for VPN firewall addresses:
   a. Go to *Network* > *IPAM* > *IPAM Rules* and click *Create new*.
   b. Enter a name for the rule.
   c. Set *Apply rule to* to *Addresses*.
   d. Set *Address Name* to the wildcard VPN*.
   e. For the *Pool*, select the first subnet pool.



f. Click *OK*.

3. Create another IPAM rule for other firewall addresses, when *Address name* matches *, and assign the second address pool, *10.2.0.0/255.255.0.0*.



4. Create a firewall address with the IPAM type that starts with *VPN*:

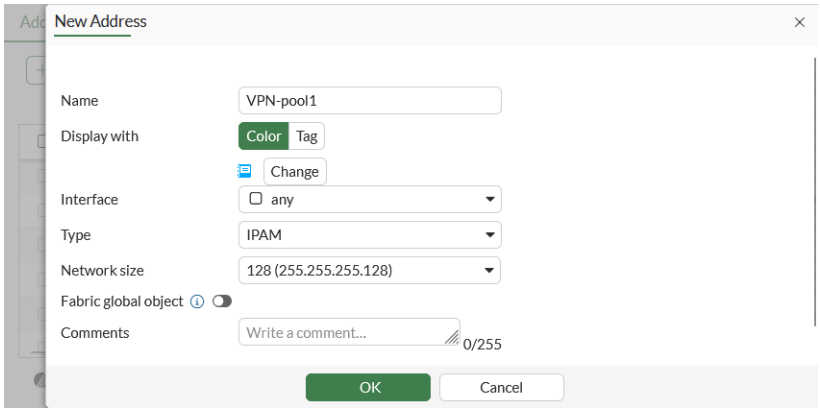
a. Go to *Policy & Objects* > *Addresses* and click *Create new*.

b. Enter a name for the address, such as *VPN-pool1*.

c. Set *Type* to *IPAM*.

d. Choose a *Network size*.

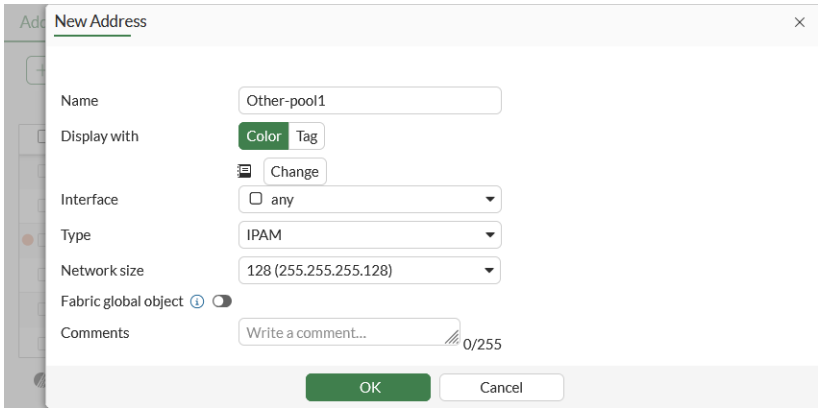
e. Configure the remaining settings as needed.



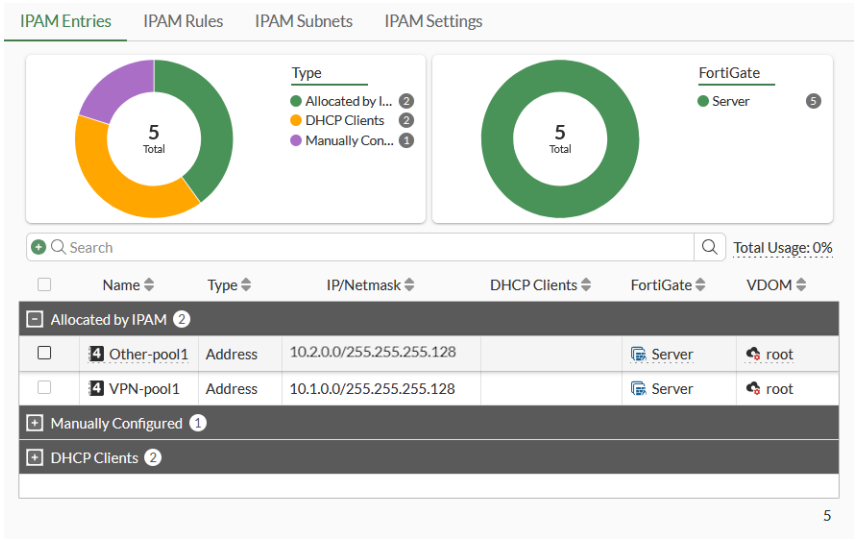
f. Click *OK*.

IPAM will start to allocate an address in the range 10.1.0.0/16.

5. Create a firewall address with the IPAM type that starts with another prefix.



6. Go to *Network* > *IPAM* > *IPAM Entries*. As the address is managed by IPAM, it is listed on the *IPAM Entries* tab.



7. Check the IPAM assignments from the CLI as well.

To configure an IPAM firewall address in the CLI:

1. Create an IPAM rule for firewall addresses:

```
config system ipam
    set status enable
    set manage-lan-addresses disable
    set manage-lan-extension-addresses disable
    set manage-ssid-addresses disable
    config pools
        edit "1"
            set subnet 10.1.0.0 255.255.0.0
        next
        edit "2"
            set subnet 10.2.0.0 255.255.0.0
        next
    end
    config rules
        edit "VPN-GW"
            set item-type address
            set item-name "VPN*"
            set device "FGVM02TM25XXXXXX"
            set vdom "*"
            set pool "1"
        next
        edit "Other-pool"
            set item-type address
            set item-name "*"
            set device "FGVM02TM25XXXXXX"
            set vdom "*"
            set pool "2"
        next
    end
end
```

2. Create a firewall address with the IPAM type:

```
config firewall address
    edit "VPN-pool1"
        set type ipam
        set color 17
        set managed-subnetwork-size 128
    next
    edit "Other-pool1"
        set type ipam
        set managed-subnetwork-size 128
    next
end
```

3. Check that the IPAM type firewall address is managed by the IPAM sever:

```
# diagnose sys ipam list entries
Entries: (sn, vdom, item-type, item, subnet/mask, conflict)
```

```
IPAM Entries:
FGVM02TM25011171 root firewall.address Other-pool1 10.2.0.0/25
FGVM02TM25011171 root firewall.address VPN-pool1 10.1.0.0/25
```

Graceful BGP shutdown

This information is also available in the FortiOS 8.0 Administration Guide:

- Graceful BGP shutdown

In this enhancement, FortiGate adds support for graceful BGP shutdown (RFC 8326) to allow controlled and orderly shutdown of BGP sessions. This ensures peers have time to adjust to the change and lower the local preferences of the routes associated with the graceful shutdown community and prefix.

CLI syntax

The following commands can be used to configure graceful BGP shutdown for the initiator and receiver. These settings will persist after a reboot.

```
config router bgp
    config neighbor
        edit <IP address>
            set shutdown {enable | disable | graceful | graceful-soft}
            set graceful-shutdown-community <community>
            set graceful-shutdown-local-preference <local preference>
            set graceful-shutdown-delay <delay>
        next
    end
end
```

Command	Description
`shutdown {enable | disable | graceful | graceful-soft}`	• `graceful`: on the graceful BGP shutdown initiator, put the device into graceful shutdown mode. BGP sessions with peers will be closed once delay is reached. • `graceful-soft`: on the graceful BGP shutdown initiator, put the device into graceful-soft shutdown mode. BGP sessions will not be closed. Instead, after the delay is reached, the local-preference in peers will be reverted to original values. This setting should only be applied to the initiator.
`graceful-shutdown-community <community>`	As the initiator, specify the community in which to advertise the graceful shutdown. As the receiver, apply the `graceful-shutdown-community` to the graceful shutdown initiator's routes and perform action on the local preference of those routes.

Command	Description
	This setting is required for both the initiator and receiver.
`graceful-shutdown-local-preference <local preference>`	The receiver lowers the local preference of routes tagged by the `graceful-shutdown-community` to the defined <local preference>.
`graceful-shutdown-delay <delay>`	After the <delay> period in seconds, initiator performs the actions for either `graceful` shutdown or `graceful-soft` shutdown. If set to `0`, then FortiGate will stay in shutdown mode until manually canceled or stopped.

If administrators want to apply a one-time graceful BGP shutdown that does not persist through a reboot, use the following commands:

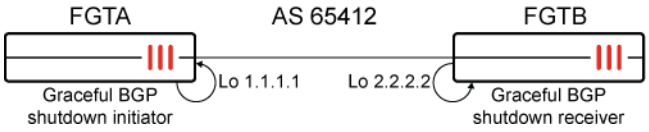
Command	Description
`execute router clear bgp all gshut [soft | cancel] [delay <60-36000>]`	On the initiator, initiate graceful shutdown on all routes. • `soft`: optionally perform a graceful-soft shutdown. • `cancel`: cancel a previously initiated shutdown. • `delay`: specify the delay timer in seconds for the shutdown. If no delay is specified, the timer is indefinite and shutdown does not occur.
`execute router clear bgp ip <IP/IPv6 address> gshut [soft | cancel] [delay <60-36000>]`	On the initiator, initiate graceful shutdown on specific routes based on the IP <IP/IPv6 address>.
`execute router clear bgp as <AS number> gshut [soft | cancel] [delay <60-36000>]`	On the initiator, initiate graceful shutdown on specific routes based on the AS <AS number>.
`execute router clear bgp group <group name> gshut [soft | cancel] [delay <60-36000>]`	On the initiator, initiate graceful shutdown on specific routes based on the group <group name>.

In addition, the default community `gshut` is added to:

- router.community-list.rule.match
- router.route-map.rule.set-community

Examples

In the following examples, two FortiGates have established BGP adjacencies.



FGTA needs to be shut down for maintenance. To pre-emptively inform its peer (FGTB), FGTA uses graceful BGP shutdown to announce the shutdown and apply the gshut community to its routes. This allows FGTB to set all routes learned from FGTA to community gshut and lower the local-preference for those routes.

Example 1: Apply graceful BGP shutdown using persistent configs

On FGTB, the graceful BGP shutdown receiver, apply the following BGP configuration:

```
config router bgp
    set as 65412
    set router-id 2.2.2.2
    set network-import-check disable
    config neighbor
        edit "1.1.1.1"
            set advertisement-interval 5
            set activate6 disable
            set capability-graceful-restart enable
            set ebgp-enforce-multihop enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65412
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
            set graceful-shutdown-community "gshut"
            set graceful-shutdown-local-preference 27
        next
    end
end
```

On FGTA, the graceful BGP shutdown initiator, apply the following BGP configuration:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set graceful-restart enable
    config neighbor
        edit "2.2.2.2"
            set advertisement-interval 5
            set activate6 disable
            set capability-graceful-restart enable
            set ebgp-enforce-multihop enable
            set next-hop-self enable
            set shutdown graceful
            set soft-reconfiguration enable
            set remote-as 65412
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
            set graceful-shutdown-community "gshut"
            set graceful-shutdown-local-preference 221
            set graceful-shutdown-delay 60
        next
    end
end
```

Once graceful shutdown mode is enabled on FGTA, it will notify FGTB of the shutdown by advertising the gshut community.

On FGTA:

```
# get router info bgp summary

VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 5
1 BGP AS-PATH entries
1 BGP community entries

Neighbor V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2  4      65412     444     464        5    0    0 00:00:54        3 (In gshut)

Total number of neighbors 1
```

FGTB's learned routes will change community and apply the configured local-preference of 27:

```
# get router info bgp network 172.16.207.2/32
VRF 0 BGP routing table entry for 172.16.207.2/32
Paths: (1 available, best #1, table Default-IP-Routing-Table, graceful shutdown)
  Not advertised to any peer
  Original VRF 0
  Local
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin incomplete distance 200 metric 0, localpref 27, valid, internal, best
      Community: gshut
      Advertised Path ID: 1
      Last update: Fri Sep 26 15:20:23 2025
```

After the configured `graceful-shutdown-delay` expires, FGTA will close the BGP session with FGTB.

On FGTA:

```
# get router info bgp summary

VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 6
1 BGP AS-PATH entries
1 BGP community entries

Neighbor V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2  4      65412     445     467        0    0    0    never Idle        (In gshut)
```

As FGTB in this example does not have any other paths, all learned routes to 172.16.207.2/32 from FGTA have disappeared.

On FGTB:

```
# get router info bgp network 172.16.207.2/32
% Network not in table
```

Example 2: Apply one-time graceful BGP shutdown using execute commands

On FGTA, use similar BGP settings as in Example 1:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set graceful-restart enable
    config neighbor
        edit "2.2.2.2"
            set graceful-shutdown-community "gshut"
            set graceful-shutdown-local-preference 221
            set graceful-shutdown-delay 0
        next
    end
end
```

`graceful-shutdown-delay` does not need to be configured.

On FGTB, use the same BGP settings as in Example 1:

```
config router bgp
    set as 65412
    set router-id 2.2.2.2
    set network-import-check disable
    config neighbor
        edit "1.1.1.1"
            set graceful-shutdown-community "gshut"
            set graceful-shutdown-local-preference 27
        next
    end
end
```

Initiate the one-time graceful shutdown from FGTA with a delay of 60 seconds:

```
# execute router clear bgp ip 2.2.2.2 gshut delay 60
```

Once graceful shutdown mode is enabled on FGTA, it will notify FGTB of the shutdown by advertising the gshut community.

On FGTA:

```
# get router info bgp summary

VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 5
1 BGP AS-PATH entries
1 BGP community entries

Neighbor V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2  4      65412     444     464        5    0    0 00:00:52        3 (In gshut)

Total number of neighbors 1
```

FGTB's learned routes will change community and apply the configured local-preference of 27 as in Example 1. After 60 seconds, BGP session will be terminated and routes from FGTA will be removed from FGTB.

Example 3: Apply graceful-soft BGP shutdown using persistent configs

Instead of completely shutting down BGP on the initiator, apply graceful-soft so that after the delay timeout, BGP reverts to normal behavior. BGP graceful shutdown receiver reverts the local-preferences of its routes.

On FGTB, the graceful BGP shutdown receiver, apply the same BGP configurations as Example 1.

On FGTA, the graceful-soft BGP shutdown initiator, apply the following BGP configurations:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set bestpath-cmp-routerid enable
    set ibgp-multipath enable
    set cluster-id 1.1.1.1
    set graceful-restart enable
    config neighbor
        edit "2.2.2.2"
            set advertisement-interval 5
            set activate6 disable
            set capability-graceful-restart enable
            set ebgp-enforce-multihop enable
            set next-hop-self enable
            set shutdown graceful-soft
            set soft-reconfiguration enable
            set remote-as 65412
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
            set graceful-shutdown-community "gshut"
            set graceful-shutdown-local-preference 221
            set graceful-shutdown-delay 300
        next
    end
end
```

Once graceful shutdown mode is enabled on FGTA, it will notify FGTB of the shutdown by advertising the gshut community:

```
# get router info bgp summary
VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 12
1 BGP AS-PATH entries
1 BGP community entries
Neighbor V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2  4      65412     599     643       12    0    0 00:00:58        3 (In gshut)
Total number of neighbors 1
```

```
# get router info bgp neighbors | grep shut
 In graceful shut down , will go back to normal in 221 seconds
```

FGTB's learned routes will change community and apply the configured local-preference of 27:

```
# get router info bgp network 172.16.207.2/32
VRF 0 BGP routing table entry for 172.16.207.2/32
Paths: (1 available, best #1, table Default-IP-Routing-Table, graceful shutdown)
  Not advertised to any peer
  Original VRF 0
  Local
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin incomplete distance 200 metric 0, localpref 27, valid, internal, best
      Community: gshut
      Advertised Path ID: 1
      Last update: Fri Sep 27 08:15:23 2025
```

After the delay time, both FortiGates will go back to normal mode:

On FGTA:

```
# get router info bgp summary
VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 12
1 BGP AS-PATH entries
1 BGP community entries
Neighbor V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2  4      65412     612     657       12    0    0 00:06:24        3
Total number of neighbors 1
```

```
# get router info bgp neighbors | grep shut
```

On FGTB:

```
# get router info bgp network 172.16.207.2/32
VRF 0 BGP routing table entry for 172.16.207.2/32
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin incomplete distance 200 metric 0, localpref 100, valid, internal, best
      Advertised Path ID: 1
      Last update: Sat Sep 27 08:17:37 2025
```

Example 4: Apply one-time graceful-soft BGP shutdown using execute command

Instead of using persistent configs like in Example 3, use the following execute command to perform a one-tine graceful-soft BGP shutdown:

```
# execute router clear bgp ip 2.2.2.2 gshut soft delay 100
```

After run this command on FGTA, FGTA will go into graceful shutdown mode, and FGTB will set `community` and `local_pref` value for all routes learned from FGTA.

After the delay time of 100 seconds, FGTA will go back to normal mode, and peer will restore the `community` and `local_pref` value for all routes learned from FGTA.

Example 5: Apply graceful BGP shutdown with indefinite delay using persistent configs on IPv6 routes

In this example, we apply graceful BGP shutdown mode indefinitely. IPv6 routes on the graceful BGP shutdown receiver will keep routes in the `graceful-shutdown-local-preference` indefinitely while in this mode.

On FGTB, apply the following configuration:

```
config router bgp
    set as 65412
    set router-id 2.2.2.2
    set network-import-check disable
    config neighbor
        edit "2000::1:1:1:1"
            set activate disable
            set soft-reconfiguration6 enable
            set remote-as 65412
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
            set graceful-shutdown-community "gshut"
            set graceful-shutdown-local-preference 276
        next
    end
end
```

On FGTA, apply the following BGP configurations:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set bestpath-cmp-routerid enable
    set ibgp-multipath enable
    set cluster-id 1.1.1.1
    set graceful-restart enable
    config neighbor
        edit "2000::2:2:2:2"
            set advertisement-interval 5
            set activate disable
            set capability-graceful-restart6 enable
            set shutdown graceful
            set soft-reconfiguration6 enable
            set remote-as 65412
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
            set graceful-shutdown-community "gshut"
            set graceful-shutdown-local-preference 2216
            set graceful-shutdown-delay 0
```

```
        next
    end
end
```

When `set shutdown graceful` is configured, FGTA is in graceful shutdown mode indefinitely:

On FGTA:

```
# get router info6 bgp summary
VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 10
1 BGP AS-PATH entries
1 BGP community entries
Neighbor      V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2000::2:2:2:2 4      65412    3235    3243       10    0    0 17:00:28        1 (In gshut)
Total number of neighbors 1
```

On FGTB:

```
# get router info6 bgp network 2000:172:16:211::/64
VRF 0 BGP routing table entry for 2000:172:16:211::/64
Paths: (1 available, best #1, table Default-IP-Routing-Table, graceful shutdown)
  Not advertised to any peer
  Original VRF 0
  Local
    2000::1:1:1:1 (metric 1000) from 2000::1:1:1:1 (1.1.1.1)
      Origin incomplete distance 200 metric 0, localpref 276, valid, internal, best
      Community: gshut
      Advertised Path ID: 1
      Last update: Fri Sep 26 15:41:04 2025
```

TCP congestion control enhancement with BBR

FortiGate integrates bottleneck bandwidth and round-trip propagation time (BBR), a model-based TCP congestion control algorithm developed by Google, offering significantly improved data transmission speeds under congested network conditions compared to the traditional CUBIC method.

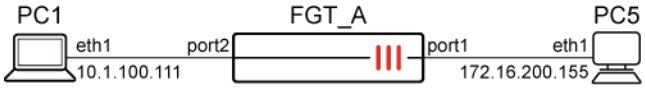
To select the TCP congestion control algorithm:

```
config system global
    set tcp-congestion-control {bbr | cubic}
end
```

Option	Description
`cubic`	FortiGate unit employs a cubic function for TCP congestion control (default).
`bbr`	FortiGate unit employs a bottleneck bandwidth and round-trip propagation time (BBR) for TCP congestion control.

Example

In this example, a file is downloaded on PC1 from PC5 using both algorithms under the same network conditions, and the data transmission speed is compared.



To compare the congestion control algorithms:

1. On the FortiGate, configure a proxy-mode firewall policy:

```
conf firewall policy
    edit 100
        set srcintf port2
        set dstintf port1
        set srcaddr all
        set dstaddr all
        set service ALL
        set schedule always
        set action accept
        set inspection-mode proxy
        set utm-status enable
        set av-profile "default"
        set ssl-ssh-profile certificate-inspection
    next
end
```

2. Set the TCP congestion control algorithm to CUBIC:

```
config system global
    set tcp-congestion-control cubic
end
```

3. Download a 100MB file from PC5 to PC1 with a 50ms delay and 1% packet loss rate on the network, and check the data transmission speed:

```
root@pc1:~# wget -O /dev/null http://172.16.200.155/big100MB.html
--2025-09-25 09:40:11--  http://172.16.200.155/big100MB.html
Connecting to 172.16.200.155:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 101869444 (97M) [text/html]
Saving to: ‘/dev/null’

 2% [==>                                                                 ]
2,733,506    570KB/s  eta 2m 38s
 2% [==>                                                                 ]
2,832,922    326KB/s  eta 3m 8s
 2% [==>                                                                 ]
2,903,874    217KB/s  eta 3m 8s
```

```
 3% [==>                                                                      ]
3,172,250   95.0KB/s  eta 4m 32s ^C
```

4. On the FortiGate, change the TCP congestion algorithm to BBR:

```
config system global
    set tcp-congestion-control bbr
end
```

5. Download the 100MB file again from PC5 to PC1 with a 50ms delay and 1% packet loss rate on the network, and check the data transmission speed:

```
root@pc1:~# wget -O /dev/null http://172.16.200.155/big100MB.html
--2025-09-25 09:44:42--  http://172.16.200.155/big100MB.html
Connecting to 172.16.200.155:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 101869444 (97M) [text/html]
Saving to: ‘/dev/null’

15% [==>                                                                      ]
16,216,786  3.82MB/s  eta 30s
32% [====================>                                                    ]
33,615,954  6.71MB/s  eta 17s
51% [================================================>                        ]
52,550,002  9.29MB/s  eta 9s
61% [=========================================================>               ]
62,858,314  10.3MB/s  eta 6s
73% [=================================================================>       ]
75,079,434  10.7MB/s  eta 4s
84% [=============================================================================> ]
85,609,290  10.1MB/s  eta 2s
```

6. Note that, under the same network conditions, the data transmission speed when using BBR is significantly better than when using CUBIC.

Session helper statistics

FortiOS now extends session helper statistics to include FTP, TFTP, RTSP, and PPTP protocols, enhancing visibility and troubleshooting for these traffic types through the CLI.

To view session help diagnostics:

```
# diagnose sys session help stats {tftp | ftp | rtsp | pptp}
```

For example:

```
# diagnose sys session help stats pptp
Calls Established: 2
Mismatched PNS Call ID: 0
GRE Session Created: 2
```

```
GRE Session Freed: 0
No Matching GRE Sesion: 0
Call ID Mismatch On Call Reply: 0
```

```
# diagnose sys session help stats ftp
PORT Requests: 5
EPRT Requests: 0
PASV Requests: 3
PASV Replies: 0
EPSV Requests: 0
EPSV Replies: 3
RETR Requests: 2
STOR Requests: 0
```

```
# diagnose sys session help stats tftp
TFTP Client sessions Created: 10
```

```
# diagnose sys session help stats rtsp
Streams Created: 18694060
Streams Freed: 19228166
Stream Creation Failures: 519922
Stream Client Ports Allocated: 0
Stream Client Ports Freed: 19228166
Unknown Stream Client Port: 0
Data Sessions Created: 18694050
Data Sessions Freed: 19228166
Data Session Creation Failures: 519932
```

Enhanced EVPN support for VXLAN Anycast gateway and type-5 routing

This enhancement adds support for VXLAN Anycast gateway, EVPN IRB (Ethernet VPN Integrated Routing and Bridging), and type-5 EVPN routes, enabling seamless L2/L3 service integration and scalable IP prefix advertisement across the VXLAN fabric. It simplifies mobility and routing in modern data center deployments. For more background on EVPN, see VXLAN with MP-BGP EVPN.

The `config router bgp` command has new options:

```
config router bgp
    config redistribute {connected | static | isis | ospf | rip}
        set status-evpn {enable | disable}
        set route-map-evpn <string>
    next
    config neighbor|neighbor-group
        edit <string>
            set adv-evpn-route {type2 | type5 | local}
    next
end
```

Option	Description
`config redistribute {connected | static | isis | ospf | rip}`	Types of routes to be redistributed.
`status-evpn {enable | disable}`	Enable/disable EVPN redistribution status.
`route-map-evpn <string>`	Route map name for EVPN redistribution.
`adv-evpn-route {type2 | type5 | local}`	Types of EVPN routes that can be advertised to this neighbor as IPv4 routes: • `type2`: Type 2 EVPN routes. • `type5`: Type 5 EVPN routes. • `local`: Local EVPN routes.

The `config system evpn` command has new options:

```
config system evpn
    edit <id>
      set type {macvrf | ipvrf}
      set adv-default-gw {enable | disable}
      set virtual-mac-vrid <integer>
      set l3-instance <integer>
      set *interface <string>
      set distribute-local-route {enable | disable}
    next
end
```

Option	Description
`type {macvrf | ipvrf}`	EVPN instance type: • `macvrf`: MAC VRF. • `ipvrf`: L3 VRF.
`adv-default-gw {enable | disable}`	Enable/disable advertisement of default gateway.
`virtual-mac-vrid <integer>`	VRID used to generate virtual MAC address (1 to 255).
`l3-instance <integer>`	L3VRF EVPN instance ID (0 to 4294967295).
`*interface <string>`	Outgoing interface for all VxLANs in this L3VRF EVPN instance. Available when `type` is set to `ipvrf`.
`distribute-local-route {enable | disable}`	Enable/disable distribution of local EVPN routes as IPv4 routes to BGP.

A new `get` command is available for `default-gateway`:

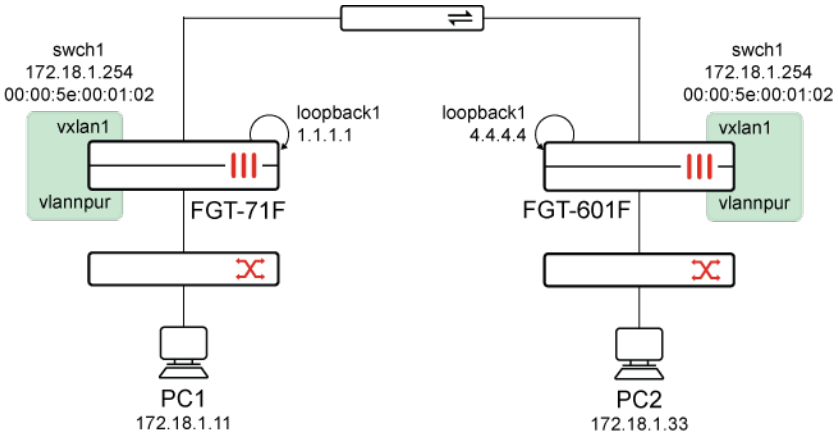
```
get l2vpn evpn default-gateway
```

Example

This section includes the following examples:

- Anycast gateway example on page 77
- IRB example on page 82
- Type-5 route example on page 87

Anycast gateway example



In a Security Fabric configured with EVPN, VXLAN, and an Anycast gateway (swch1), all leaf switches that connect endpoints in the same VLAN (or subnet) share the same virtual default gateway IP address. The leaf switches also share the same virtual MAC address for that gateway.

Endpoints (such as servers, VMs, containers) always see the same gateway regardless of which leaf they connect to. If a VM or host moves from Leaf1 to Leaf2, its default gateway stays the same and doesn't require any reconfiguration (for example re-ARP or a new IP address), which provides seamless mobility.

This example describes how to configure FortiGate-71F and FortiGate-601F in the topology. It also describes how to verify the configuration.

To configure FortiGate-71F:

1. Configure EVPN:

```
config system evpn
    edit 100
        set rd "100:100"
        set import-rt "1:1"
        set export-rt "1:1"
        set ip-local-learning enable
        set adv-default-gw enable
        set virtual-mac-vrid 2
    next
end
```

2. Configure VXLAN:

```
config system vxlan
    edit "vxlan1"
        set interface "loopback1"
        set vni 1000
        set evpn-id 100
        set learn-from-traffic enable
    next
end
```

3. Configure a system switch interface:

```
config system switch-interface
    edit "swch1"
        set vdom "root"
        set member "vxlan1" "vlannpur"
    next
end
```

4. Configure a system interface:

```
config system interface
    edit "swch1"
        set vdom "root"
        set ip 172.18.1.254 255.255.255.0
        set allowaccess ping
        set type switch
        set snmp-index 26
    next
end
```

5. Configure BGP:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set ibgp-multipath enable
    set network-import-check disable
    set recursive-next-hop enable
    set graceful-restart enable
    config neighbor
        edit "172.18.5.123"
            set ebgp-enforce-multihop enable
            set next-hop-self enable
            set next-hop-self-vpnv4 enable
            set soft-reconfiguration enable
            set soft-reconfiguration-evpn enable
            set remote-as 65412
            set graceful-shutdown-community "22:55"
        next
    end
    config network
```

```
        edit 1
            set prefix 1.1.1.1 255.255.255.255
        next
        edit 4
            set prefix 172.28.6.0 255.255.255.0
        next
    end
    config redistribute "connected"
    end
    config redistribute "rip"
    end
    config redistribute "ospf"
    end
    config redistribute "static"
    end
    config redistribute "isis"
    end
    config redistribute6 "connected"
    end
    config redistribute6 "rip"
    end
    config redistribute6 "ospf"
    end
    config redistribute6 "static"
    end
    config redistribute6 "isis"
    end
end
```

To configure FortiGate-601F:

1. Configure EVPN:

```
config system evpn
    edit 100
        set rd "100:100"
        set import-rt "1:1"
        set export-rt "1:1"
        set ip-local-learning enable
        set adv-default-gw enable
        set virtual-mac-vrid 2
    next
end
```

2. Configure VXLAN:

```
config system vxlan
    edit "vxlan1"
        set interface "loopback1"
        set vni 1000
```

```
        set evpn-id 100
        set learn-from-traffic enable
    next
end
```

3. Configure a system switch interface:

```
config system switch-interface
    edit "swch1"
        set vdom "root"
        set member "vxlan1" "vlannpur"
    next
end
```

4. Configure a system interface:

```
config system interface
    edit "swch1"
        set vdom "root"
        set ip 172.18.1.254 255.255.255.0
        set allowaccess ping
        set type switch
        set snmp-index 50
    next
end
```

5. Configure BGP:

```
config router bgp
    set as 65412
    set router-id 4.4.4.4
    set ibgp-multipath enable
    set recursive-next-hop enable
    config neighbor
        edit "172.18.5.121"
            set next-hop-self enable
            set next-hop-self-vpnv4 enable
            set soft-reconfiguration enable
            set soft-reconfiguration-evpn enable
            set remote-as 65412
        next
    end
    config network
        edit 1
            set prefix 4.4.4.4 255.255.255.255
        next
    end
    config redistribute "connected"
    end
    config redistribute "rip"
```

```
    end
    config redistribute "ospf"
    end
    config redistribute "static"
    end
    config redistribute "isis"
    end
    config redistribute6 "connected"
    end
    config redistribute6 "rip"
    end
    config redistribute6 "ospf"
    end
    config redistribute6 "static"
    end
```

To verify:

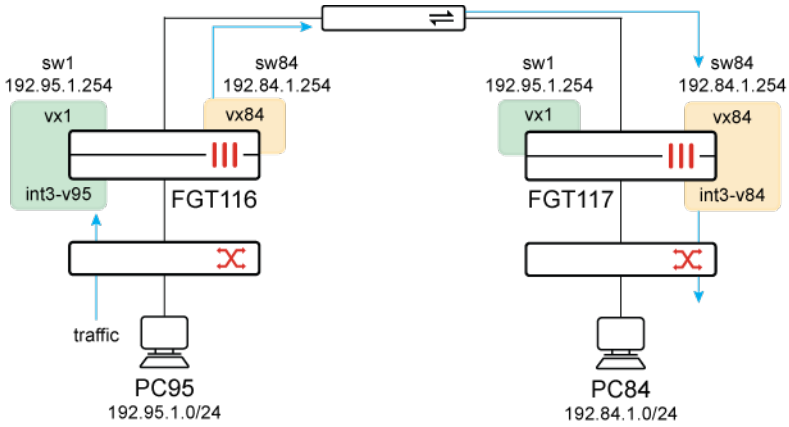
1. From FortiGate-601F, ping 172.18.1.11:

```
# execute ping 172.18.1.11
PING 172.18.1.11 (172.18.1.11): 56 data bytes
64 bytes from 172.18.1.11: icmp_seq=0 ttl=255 time=0.4 ms
64 bytes from 172.18.1.11: icmp_seq=1 ttl=255 time=0.3 ms
^C
--- 172.18.1.11 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.3/0.3/0.4 ms

FortiGate-601F (vdom1) (Interim)# diagnose ip arp list | grep 254
index=51 ifname=vlannpu1 172.18.1.254 00:00:5e:00:01:02 state=00000004 use=9094399
confirm=9100246 update=9094246 ref=0

FortiGate-601F (vdom1) (Interim)#
```

2. From FortiGate-71F, diagnose the IP ARP list:

```
# diagnose ip arp list
index=27 ifname=vlannpu1 172.18.1.33 86:39:8f:13:f7:01 state=00000002 use=2374 confirm=2374
update=1946 ref=3
index=27 ifname=vlannpu1 172.18.1.254 00:00:5e:00:01:02 state=00000002 use=435 confirm=435
update=435 ref=2
version count: 2
```

IRB example



IRB (Integrated Routing and Bridging) in EVPN means that FortiGate can both bridge (L2) and route (L3) traffic inside the Security Fabric configured with VXLAN/EVPN. EVPN IRB allows L2 services for intra-subnet communication and L3 services for inter-subnet communications.

This example uses the following VLAN, subnets, and VNI names:

PC	VLAN	Subnet	VNI
PC95	vlan95	192.95.1.0/24	vni 141
PC94	vlan84	192.84.1.0/24	vni 84

This example describes how to configure FGT116 and FGT117 in the topology to allow inter-subnet routing between vlan95 and vlan84 over the VXLANs between the two FortiGate VTEPs. This allows PC95 to reach PC84 using EVPN over VXLAN.

To configure FGT116:

1. Configure EVPN:

```
config system evpn
    edit 1
        set import-rt "1:1"
        set export-rt "1:1"
        set ip-local-learning enable
    next
    edit 84
        set import-rt "84:84"
        set export-rt "84:84"
        set ip-local-learning enable
    next
end
```

2. Configure VXLAN interfaces:

```
config system vxlan
    edit "vx1"
```

```
        set interface "loopback1"
        set vni 141
        set evpn-id 1
        set learn-from-traffic enable
    next
     edit "vx84"
        set interface "loopback1"
        set vni 84
        set evpn-id 84
        set learn-from-traffic enable
    next
end
```

3. Configure a system interface:

```
config system interface
    edit "int3-v95"
        set vdom "root"
        set interface "internal3"
        set vlanid 95
    next
end
```

4. Configure system switch interfaces and select the VXLAN interface:

```
config system switch-interface
    edit "sw1"
        set vdom "root"
        set member "int3-v95" "vx1"
        set intra-switch-policy explicit
    next
      edit "sw84"
        set vdom "root"
        set member  "vx84"
        set intra-switch-policy explicit
    next
end
```

5. Configure system interfaces:

```
config system interface
    edit "sw1"
        set vdom "root"
        set ip 192.95.1.254 255.255.255.0
        set allowaccess ping ssh telnet
    next
    edit "sw84"
        set vdom "root"
        set ip 192.84.1.254 255.255.255.0
        set allowaccess ping ssh telnet
    next
end
```

6. Configure BGP:

```
config router bgp
    set as 65010
    set router-id 10.255.255.116
    config neighbor
        edit "10.1.2.117"
            set next-hop-self enable
            set soft-reconfiguration-evpn enable
            set remote-as 65010
            set connect-timer 1
        next
    end
    config network
        edit 1
            set prefix 10.255.255.116 255.255.255.255
        next
    end
end
```

To configure FGT117:

1. Configure EVPN:

```
config system evpn
    edit 1
        set import-rt "1:1"
        set export-rt "1:1"
        set ip-local-learning enable
    next
    edit 84
        set import-rt "84:84"
        set export-rt "84:84"
        set ip-local-learning enable
    next
end
```

2. Configure VXLAN interfaces:

```
config system vxlan
    edit "vx1"
        set interface "loopback1"
        set vni 141
        set evpn-id 1
        set learn-from-traffic enable
    next
     edit "vx84"
        set interface "loopback1"
        set vni 84
        set evpn-id 84
        set learn-from-traffic enable
```

```
   next
end
```

3. Configure a system switch interface and select the VXLAN interfaces:

```
config system switch-interface
    edit "sw1"
        set vdom "root"
        set member "vx1"
        set intra-switch-policy explicit
    next
      edit "sw84"
        set vdom "root"
        set member "int3-v84"  "vx84"
        set intra-switch-policy explicit
    next
end
```

4. Configure system interfaces:

```
config system interface
    edit "sw1"
        set vdom "root"set
        set ip 192.95.1.254 255.255.255.0
        set allowaccess ping ssh telnet
    next
    edit "sw84"
        set vdom "root"
        set ip 192.84.1.254 255.255.255.0
        set allowaccess ping ssh telnet
    next
end
```

5. Configure BGP:

```
config router bgp
    set as 65010
    set router-id 10.255.255.117
    config neighbor
        edit "10.1.2.116"
            set next-hop-self enable
            set soft-reconfiguration-evpn enable
            set remote-as 65010
            set connect-timer 1
        next
    end
    config network
        edit 1
            set prefix 10.255.255.117 255.255.255.255
        next
```

```
    end
end
```

To verify:

1. On FGT116, view the routing table. The FortiGate learned both the subnet of vlan95 and vlan84:

```
# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       V - BGP VPNv4, E - BGP EVPN
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 172.18.18.1, wan1, [1/0]
C       10.1.2.0/24 is directly connected, wan2
C       10.255.255.116/32 is directly connected, loopback1
B       10.255.255.117/32 [200/0] via 10.1.2.117 (recursive is directly connected, wan2),
20:28:32, [1/0]
C       172.18.18.0/24 is directly connected, wan1
C       192.84.1.0/24 is directly connected, sw84
C       192.95.1.0/24 is directly connected, sw1
C       192.96.0.0/16 is directly connected, int3-v96
```

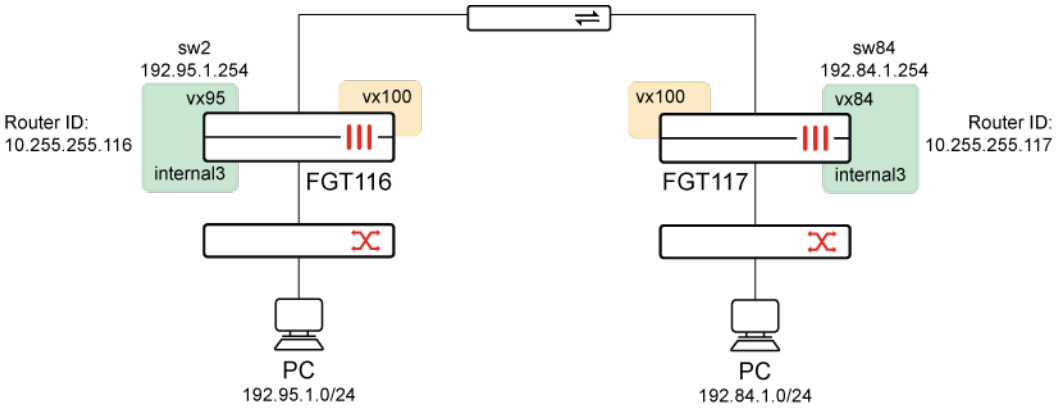
2. Look at the L2 EVPN table for VNI 84, FGT116 has learned the MAC address for PC84.
   The MAC address for PC84 is 00:0c:29:da:ca:07 84.

```
# get l2vpn evpn table 84

EVPN instance 84
 Broadcast domain VNI 84 TAGID 0

  EVPN MAC table:
  MAC               VNI      Remote Addr     Binded Address
  00:0c:29:da:ca:07 84       10.255.255.117  -
  00:11:92:d4:5e:8f 84       10.255.255.117  -
  04:d5:90:b1:9d:e8 84       10.255.255.117  -
  6a:a5:45:75:d5:65 84       10.255.255.117  -

  EVPN Local MAC table:
  "Inactive" means this MAC/IP pair will not be sent to peer.
  Flag code: S - Static, F - FDB, L - Switch local MAC, G - Default Gateway.
  Trailing * means HA
  MAC               Flag Status   Binded Address
  02:4c:63:08:3d:39 LG   Inactive 192.84.1.254
                    L    Active   -

  EVPN Local IP table:
  Address         MAC
```

```
  192.84.1.254    02:4c:63:08:3d:39

  EVPN PEER table:
  VNI      Remote Addr     Binded Address
  84       10.255.255.117  10.255.255.117
```

Type-5 route example

Instead of advertising MAC/IP bindings used in type-2 routes, type-5 EVPN routes advertise IP prefixes (subnets), allowing EVPN to carry L3 VPN routes across the Security Fabric configured with VXLAN to provide more than only L2 reachability.



This example uses the following VLAN and VNI names and IDs:

PC and L3 instance	VLAN	VNI
PC95	vlan95	vni 95
PC94	vlan84	vni 84
L3 instance	-	vni 100

This example describes how to configure FGT116 and FGT117 in the topology. It also describes how to verify the configuration.

To configure FGT116:

1. Configure EVPN:

```
config system evpn
    edit 95
        set import-rt "95:95"
        set export-rt "95:95"
        set l3-instance 100
    next
    edit 100
        set type ipvrf
        set rd "100:100"
        set import-rt "100:100"
```

```
        set export-rt "100:100"
        set interface "loopback1"
    next
end
```

2. Configure VXLAN interfaces:

```
config system vxlan
    edit "vx95"
        set vni 95
        set evpn-id 95
        set learn-from-traffic enable
    next
    edit "vx100"
        set vni 100
        set evpn-id 100
    next
end
```

3. Configure the system switch interface:

```
config system switch-interface
    edit "sw2"
        set vdom "root"
        set member "vx95" "internal3"
        set intra-switch-policy explicit
    next
end
```

4. Configure the system interface:

```
config system interface
    edit "sw2"
        set vdom "root"
        set ip 192.95.1.254 255.255.0.0
        set allowaccess ping
        set type switch
    next
end
```

5. Configure BGP:

```
config router bgp
    set as 65010
    set router-id 10.255.255.116
    config neighbor
        edit "10.1.2.117"
            set next-hop-self enable
            set soft-reconfiguration-evpn enable
            set remote-as 65010
            set connect-timer 1
        next
```

```
    end
    config network
        edit 1
            set prefix 10.255.255.116 255.255.255.255
        next
    end
    config redistribute "connected"
        set status evpn enable
    end
end
```

To configure FGT117:

1. Configure EVPN:

```
config system evpn
    edit 100
        set type ipvrf
        set rd "100:100"
        set import-rt "100:100"
        set export-rt "100:100"
        set interface "loopback1"
    next
    edit 84
        set import-rt "84:84"
        set export-rt "84:84"
        set l3-instance 100
    next
end
```

2. Configure VXLAN interfaces:

```
config system vxlan
    edit "vx84"
        set vni 84
        set evpn-id 84
        set learn-from-traffic enable
    next
    edit "vx100"
        set vni 100
        set evpn-id 100
        set learn-from-traffic enable
    next
end
```

3. Configure the system switch interface:

```
config system switch-interface
    edit "sw84"
        set vdom "root"
        set member "vx84" "internal3"
        set intra-switch-policy explicit
```

```
    next
end
```

4. Configure the system interface:

```
config system interface
    edit "sw84"
        set vdom "root"
        set ip 192.84.1.254 255.255.0.0
        set allowaccess ping
        set type switch
    next
end
```

To verify:

1. On FGT117, get router info for the BGP EVPN network:
   Type-5 routes are in bold.

```
FGT117# get router info bgp evpn network
<string>    L2VPN EVPN prefix

FGT117# get router info bgp evpn network
   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (Default for VRF 0)
*> [5][0][16][192.84.0.0]/72
                    10.255.255.117      0             100  32768        0 ? <-/1>
*> [5][0][16][192.95.0.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*> [5][0][16][192.96.0.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*  [5][0][16][192.96.0.0]/72
                    10.255.255.117      0             100  32768        0 ? <-/->
*> [5][0][24][10.1.2.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*  [5][0][24][10.1.2.0]/72
                    10.255.255.117      0             100  32768        0 ? <-/->
*> [5][0][24][172.18.18.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*  [5][0][24][172.18.18.0]/72
                    10.255.255.117      0             100  32768        0 ? <-/->
*> [5][0][32][10.255.255.116]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*> [5][0][32][10.255.255.117]/72
                    10.255.255.117      0             100  32768        0 ? <-/1>

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (received from VRF 0)
*>i[5][0][16][192.95.0.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*>i[5][0][16][192.96.0.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
```

```
*>i[5][0][24][10.1.2.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*>i[5][0][24][172.18.18.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*>i[5][0][32][10.255.255.116]/72
                    10.255.255.116      0             100      0        0 ? <-/1>

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
Route Distinguisher: 0.0.0.0:84 (Default for VRF 0)
*> [2][0][48][00:0c:29:da:ca:07][0]/72
                    10.255.255.117      0             100  32768        0 i <-/1>
*> [2][0][48][00:11:92:d4:5e:8f][0]/72
                    10.255.255.117      0             100  32768        0 i <-/1>
*> [2][0][48][04:d5:90:b1:9d:e8][0]/72
                    10.255.255.117      0             100  32768        0 i <-/1>
*> [2][0][48][0e:61:aa:b2:bb:c3][0]/72
                    10.255.255.117      0             100  32768        0 i <-/1>
*> [3][0][32][10.255.255.117]/80
                    10.255.255.117      0             100  32768        0 i <-/1>
```

2. Get router info for the BGP EVPN network:

```
FGT117# get router info bgp evpn network [5][0][16][192.95.0.0]/72
Route Distinguisher: 100:100 (Default for VRF 0)
BGP routing table entry for [5][0][16][192.95.0.0]/72
  Original VRF 0
  Local
    10.255.255.116 from 10.1.2.116 (10.255.255.116)
      Origin incomplete distance 200 metric 0, localpref 100, valid, internal, evpn, best
      Extended Community: RT:100:100  RMAC:d2:22:98:41:e5:ba
      RD [100:100]
      LABEL1 100
      Last update: Fri Apr 11 14:08:56 2025

Route Distinguisher: 100:100 (received from VRF 0)
BGP routing table entry for [5][0][16][192.95.0.0]/72
  Original VRF 0
  Local
    10.255.255.116 from 10.1.2.116 (10.255.255.116)
      Origin incomplete distance 200 metric 0, localpref 100, valid, internal, evpn, best
      Extended Community: RT:100:100  RMAC:d2:22:98:41:e5:ba
      RD [100:100]
      LABEL1 100
      Last update: Fri Apr 11 14:08:56 2025
```

Enhanced PIM support for IPv6 across all VRFs

PIM now supports all VRFs (up to 511) and is aware of IPv6 multicast routing and forwarding over a single overlay, enhancing network scalability and flexibility compared to the previous VRF 0-only support. See Multicast routing and PIM support for more information.

A new `config router multicast6-flow` command is available:

```
config router multicast6-flow
    edit <ID>
        config flows
            edit <ID>
                set group-addr <IPv6 address>
            next
        end
    next
end
```

Option	Description
`router multicast6-flow`	Configure IPv6 multicast-flow.
`Edit <ID>`	Multicast6 flow ID.
`flows`	IPv6 multicast-flow entries.
`Edit <ID>`	Flow ID.
`group-addr <IPv6 address>`	Multicast group IP address.

The `config router multicast6` command has new options to configure PIM6 RP for PIM6 interface:

```
config router multicast6
    config interface
        edit <name>
            set rp-candidate {enable | disable}
            set static-group <string>
            set rp-candidate-group <string>
            set rp-candidate-priority <integer>
            set rp-candidate-interval <integer>
        next
    end
end
```

Option	Description
`interface`	Protocol Independent Multicast (PIM) interfaces.
`Edit <name>`	Interface name.
`rp-candidate {enable | disable}`	Enable/disable compete to become RP in elections.
`static-group <string>`	Statically set IPv6 multicast groups to forward out.
`rp-candidate-group <string>`	Multicast groups managed by this RP. Enter IPv6 access list name.
`rp-candidate-priority <integer>`	Router's priority as RP (0 to 255, default = 192).
`rp-candidate-interval <integer>`	RP candidate advertisement interval (1 to 16383 sec, default = 60).

The `config router multicast6` command has new options:

```
config router multicast6
    config pim-sm-global
        set bsr-candidate {enable | disable}
        set bsr-interface <string>
        set bsr-priority <integer>
        set bsr-hash <integer>
        set bsr-allow-quick-refresh {enable | disable}
        set cisco-crp-prefix {enable | disable}
        set cisco-ignore-rp-set-priority {enable | disable}
        set spt-threshold {enable | disable}
        set spt-threshold-group
        config rp-address
            edit <id>
                set ip6-address <IPv6 addr>
                set group <string>
            next
        end
    end
end
```

Option	Description
`pim-sm-global`	PIM sparse-mode global settings.
`bsr-candidate {enable | disable}`	Enable/disable allowing this router to become a bootstrap router (BSR).
`bsr-interface <string>`	Interface to advertise as candidate BSR.
`bsr-priority <integer>`	BSR priority (0 - 255, default = 0).
`bsr-hash <integer>`	BSR hash length (0 - 32, default = 10).
`bsr-allow-quick-refresh {enable | disable}`	Enable/disable accept BSR quick refresh packets from neighbors.
`cisco-crp-prefix {enable | disable}`	Enable/disable making candidate RP compatible with old Cisco IOS. • `enable`: Do not allow sending group prefix of zero. • `disable`: Allow sending group prefix of zero.
`cisco-ignore-rp-set-priority {enable | disable}`	Use only hash for RP selection (compatibility with old Cisco IOS): • `enable`: Ignore RP-SET priority value. • `disable`: Do not ignore RP-SET priority value.
`spt-threshold {enable | disable}`	Enable/disable switching to source specific trees.
`spt-threshould-group <string>`	Groups allowed to switch to source tree. Enter the IPv6 access list name.
`rp-address`	Statically configure RP addresses.
`Edit <id>`	VRF ID.

Option	Description
ip6-address <IPv6 address>	RP router IPv6 address.
group <string>	Groups to use this RP. Enter the IPv6 access list name.

The `config router multicast6` command has new options:

```
config router multicast6
    config pim-sm-global-vrf
        edit <id>
            set bsr-candidate {enable | disable}
            set bsr-interface <string>
            set bsr-priority <integer>
            set bsr-hash <integer>
            set bsr-allow-quick-refresh {enable | disable}
            set cisco-crp-prefix {enable | disable}
            config rp-address
                edit <id>
                    set ip6-address <IPv6 addr>
                    set group <string>
                next
            end
        next
    end
end
```

Option	Description
pim-sm-global-vrf	Per-VRF PIM sparse-mode global settings.
edit <VRF ID>	VRF ID.
bsr-candidate {enable | disable}	Enable/disable allowing this router to become a bootstrap router (BSR).
bsr-interface <string>	Interface to advertise as candidate BSR.
bsr-priority <integer>	BSR priority (0 - 255, default = 0).
bsr-hash <integer>	BSR hash length (0 - 128, default = 126).
bsr-allow-quick-refresh {enable | disable}	Enable/disable accept BSR quick refresh packets from neighbors.
cisco-crp-prefix {enable | disable}	Enable/disable making candidate RP compatible with old Cisco IOS.
rp-address	Statically configured RP addresses.
edit <id>	ID of the entry.
ip6-address <IPv6 addr>	ip6 xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx.
group <string>	IPv6 access list name.

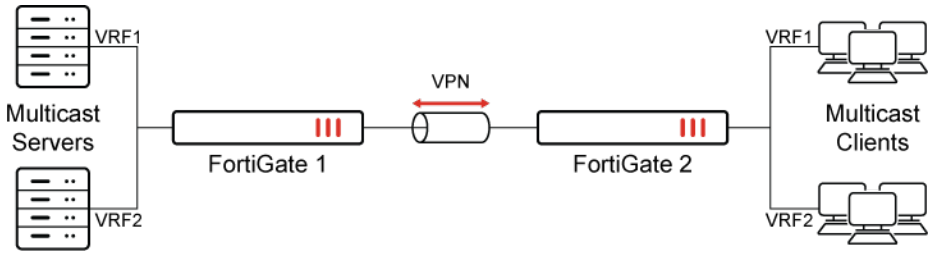
The following commands are available:

```
execute mrouter6 clear mld interface
execute mrouter6 clear mld vrf
```

The following commands now support VRF:

```
execute mrouter6 clear mld group
execute mrouter6 clear sparse-routes
get router info6 multicast mld groups
get router info6 multicast mld groups-detail
get router info6 multicast pim sparse-mode bsr-router
get router info6 multicast pim sparse-mode rp mapping
get router info6 multicast pim sparse-mode rp-hash
get router info6 multicast pim sparse-mode nexthop
get router info6 multicast pim sparse-mode mroute
```

Example



In this example, the multicast server:

- Sends out multicast traffic `ff0e::10 from 2000:172:16:205::11` in VRF1.
- Sends out multicast traffic `ff0e::20 from 2000:172:16:205::22` in VRF2.

To configure IPv6 multicast-flow:

```
config router multicast6-flow
    edit "flow6a"
        config flows
            edit 1
                set group-addr ff0e::10
            next
        end
    next
    edit "flow6b"
        config flows
            edit 1
                set group-addr ff0e::20
            next
        end
    next
end
```

To configure PIM-SM global settings for each VRF:

```
config router multicast6
    set multicast-routing enable
    config interface
        edit "vd3-1"
        next
        edit "vd3-vlan33"
        next
        edit "vd3-vlan331"
        next
    end
    config pim-sm-global-vrf
        edit 2
            config rp-address
                edit 1
                    set ip6-address 2011::12
                next
            end
        next
        edit 1
            config rp-address
                edit 1
                    set ip6-address 2011::11
                next
            end
        next
    end
end
```

For brevity, only the commands relevant to this enhancement are included.

To verify VRF in IPv6 multicast routing:

1. Review the sniffer information:
   - The VRF1 client can receive ff0e::10 and cannot receive ff0e::20:

     ```
     0.800190 npu0_vlink0 out 2000:172:16:205::11.33372 -> ff0e::10.5001: udp 46 [flowlabel
     0xa620f]
     0.800197 vd33-vlan33 in 2000:172:16:205::11.33372 -> ff0e::10.5001: udp 46 [flowlabel
     0xa620f]
     ```

   - The VRF2 client can receive ff0e::20 and cannot receive ff0e::10:

     ```
     0.666623 vd3-vlan331 out 2000:172:16:205::22.33372 -> ff0e::20.5001: udp 46 [flowlabel
     0xa620f
     0.666630 vd3-vlan331 out 2000:172:16:205::22.33372 -> ff0e::20.5001: udp 46 [flowlabel
     0xa620f
     ```

2. Review the group information:

   ```
   # get router info6 multicast mld groups
   MLD Connected Group Membership
   ```

```
VRF Group Address                          Interface           Uptime   Expires  Last
Reporter
1   ff0e::10                               vd3-vlan33          00:21:05 now(static) ::
2   ff0e::20                               vd3-vlan331         00:21:05 now(static) ::
```

GPON support for FortiGates with SFP/SFP+ ports

This information is also available in the FortiOS 8.0 Administration Guide:

- GPON support for FortiGates with SFP/SFP+ ports

Support for Gigabit Passive Optical Network (GPON) transceivers has been added to FortiGates with SFP/SFP+ ports, enabling the firewall to operate directly as an ONU (Optical Network Unit) and allowing deployment in fiber-based access networks. This allows customers transitioning from legacy DSL infrastructure to adopt widely used GPON technology, improving interoperability and easing migration.

The following diagnose commands are added to the CLI:

Command	Description
diagnose gpon config loid [interface] [loid] [password]	Configure Logical Identifier (LOID) username and password.
diagnose gpon sysinfo [interface]	Show transceiver system information.
diagnose gpon state [interface]	Show SFP GPON connection state.
diagnose gpon traffic-stats [interface]	Show SFP GPON traffic statistics.
diagnose gpon reboot [interface]	Reboot transceiver.

To detect a GPON transceiver:

```
# get system interface transceiver sfp1
Interface sfp1 - SFP/SFP+/SFP28, 1000BASE-LX
  Diagnostics  :            temperature, voltage, Tx bias, Rx power (avg), Tx power
  Vendor Name  :            FORTINET
  Part No.     :            WST-SFP-GPAU2MIF
  Serial No.   :            TK5ANK0006
  Measurement  Unit         Value        High Alarm   High Warning Low Warning  Low Alarm
  ------------ ------------ ------------ ------------ ------------ ------------ ------------

  Temperature  (Celsius)     45.8         88.0         85.0        -40.0        -45.0
  Voltage      (Volts)       3.34         3.60         3.47         3.00         2.90
  Tx Bias      (mA)         14.49        60.00        55.00         0.00         0.00
  Rx Power Avg (dBm)         -9.2         -7.0         -8.0        -28.5        -29.6
```

```
Tx Power     (dBm)          1.3          6.5          5.5         -0.5         -1.0
++ : high alarm, + : high warning, - : low warning, -- : low alarm, ? : suspect.
```

To check the GPON system information:

```
# diagnose gpon sysinfo sfp1
Mac Address: 80:A5:79:50:11:1D
Model Name: WST-SFP-GPAU2MIF
Vendor: FORTINET
Hardware Version: V1.0
Software Version: V1.2.4
Gpon SN: TK5ANK0006
```

To check the GPON state:

```
# diagnose gpon state sfp1
State codes: O1 - Initial, O2 - Standby, O3 - Serial-Number, O4 - Ranging
             O5 - Operation, O6 - POPUP, O7 - Emergency-Stop

los = 0
state = O5
fec = 1
```

To check GPON traffic statistics:

```
# diagnose gpon traffic-stats sfp1
dsDropEvents = 0
usDropEvents = 0
dsOcts = 14395972
usOcts = 34376
dsPkts = 224922
usPkts = 669
dsBcastPkts = 0
usBcastPkts = 0
dsMcastPkts = 0
usMcastPkts = 0
dsCrcErrPkts = 0
usCrcErrPkts = 0
dsUnderSzPkts = 0
usUnderSzPkts = 0
dsOverSzPkts = 0
usOverSzPkts = 0
dsFragments = 0
usFragments = 0
dsJabbers = 0
usJabbers = 0
dsPkts_l64 = 0
dsPkts_64_127 = 224920
dsPkts_128_255 = 0
dsPkts_256_511 = 2
```

```
dsPkts_512_1023 = 0
dsPkts_1024_1518 = 0
usPkts_l64 = 0
usPkts_64_127 = 0
usPkts_128_255 = 0
usPkts_256_511 = 2
usPkts_512_1023 = 0
usPkts_1024_1518 = 0
dsDiscards = 0
usDiscards = 0
dsErr = 0
usErr = 0
```

802.1X, MAB, dynamic VLAN, and address object support for software switch

802.1X, MAC authentication bypass (MAB), and dynamic VLAN support is added to the FortiGate software switch, enabling authenticated access control and intra-switch policies to regulate traffic.

With 802.1X and MAB authentication and a firewall address object, the group attribute returned from a RADIUS server can be used as the source address in firewall policies, enabling micro-segmentation for different assets within the same VLAN.

CLI syntax

A new firewall address `sub-type` of `8021x` is available for IPv4 and IPv6 addresses:

```
config firewall address
    edit <name>
        set sub-type 8021x
    next
end
```

```
config firewall address6
    edit <name>
        set sub-type 8021x
    next
end
```

Option	Description
`sub-type 8021x`	Sub-type of address: • `8021x`: 802.1X address. The address name should match the `Fortinet-Group-Name` AVP returned by the RADIUS server.

Under `config system interface`, the `security-mode` option can now be set to `802.1X` when `type` is `switch`:

```
config system interface
    edit <name>
        set vdom "root"
        set type switch
        set security-mode 802.1X
    next
end
```

Option	Description
`security-mode 802.1X`	Security mode for the interface: • `802.1X`: 802.1X port-based authentication. Available when `type` is `switch`, and switch interface members are all physical interfaces.

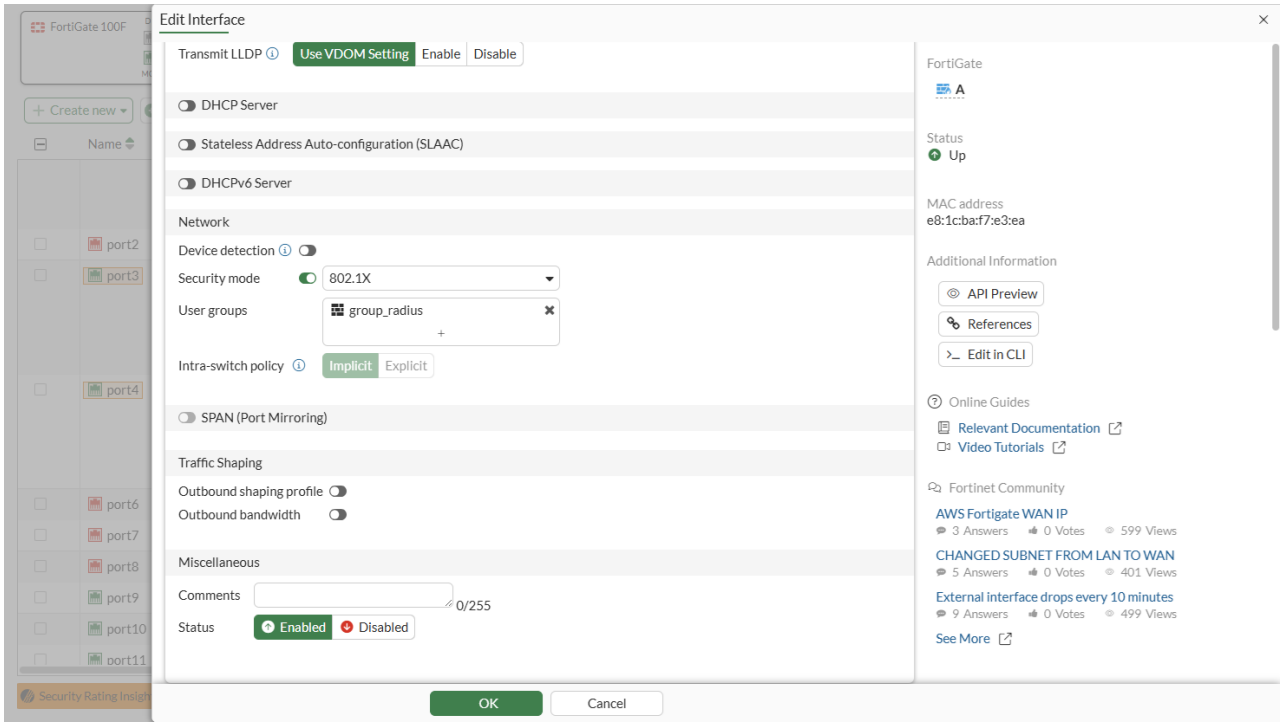
New diagnose commands are available for 802.1X firewall addresses:

```
# diagnose sys 802-1x-addr
# diagnose sys 802-1x-addr6
```

GUI

To enable 802.1x for a software switch in the GUI:

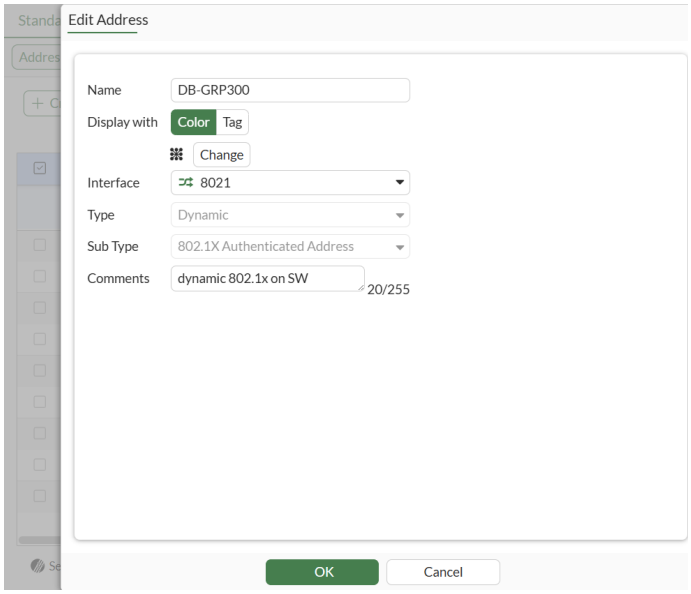
1. Go to *Network* > *Interfaces*, and create a software switch.
2. Ensure *Security mode* is set to *802.1x* on the software switch and a RADIUS user group is selected.



3. Click *OK*.

To create an address object with 802.1x authentication in the GUI:

1. Go to *Policy & Objects* > *Addresses*.
2. Click *Create New*.
3. Set *Name*.
   The name must match the `Fortinet-Group-Name` AVP returned by the RADIUS server. This example uses the name DB-GRP300.
4. Set *Interface* to the software switch.
5. Set *Type* to *Dynamic*.
6. Set *Sub Type* to *802.1X Authenticated Address*.



7. Click *OK*.

Example

This section includes the following examples:

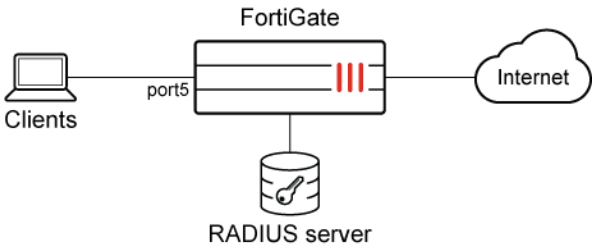
- Configuring a software switch with 802.1x on page 102
- Configuring a software switch with 802.1x MAB on page 105
- Configuring a software switch with 802.1x dynamic VLAN on page 106
- Configuring address objects with 802.1x on page 108

All the examples use 802.1x authentication with a RADIUS server.

Prerequisites:

1. Configure a RADIUS server (see RADIUS servers).
2. Define user groups to use the remote RADIUS server and for 802.1X authentication (see User definition, groups, and settings).
3. Enable 802.1X authentication on the client devices.

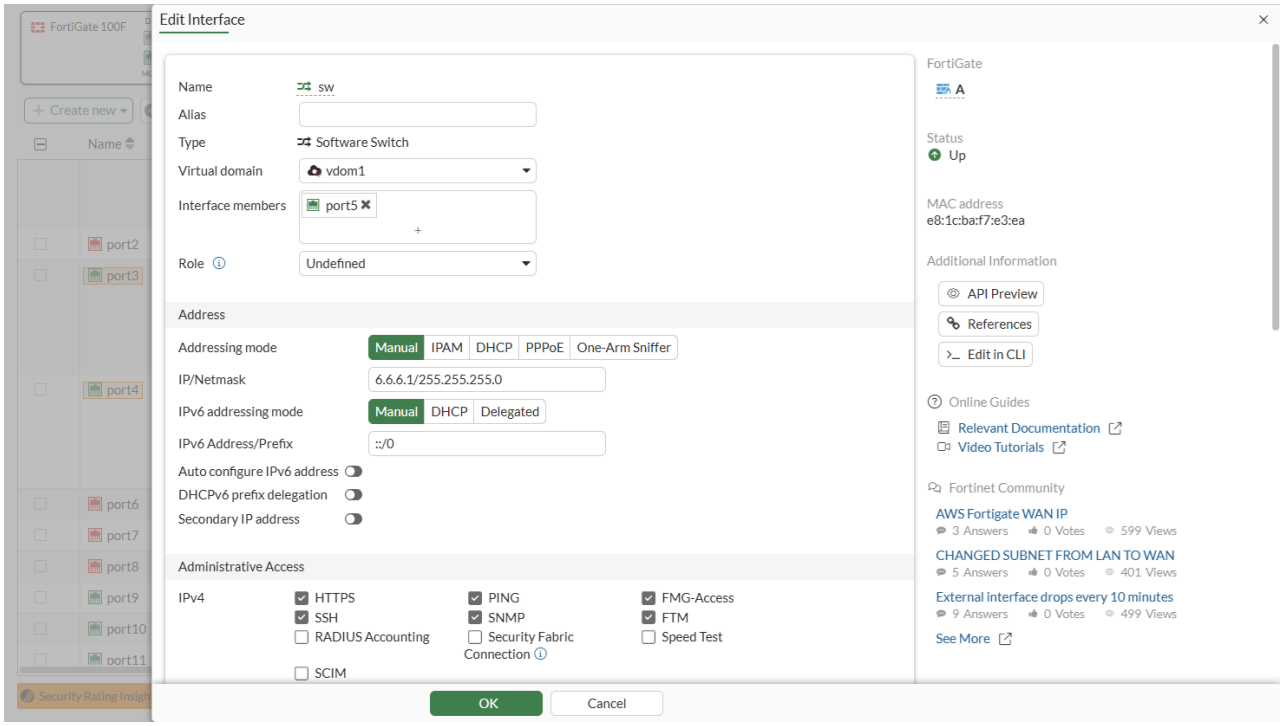
Configuring a software switch with 802.1x



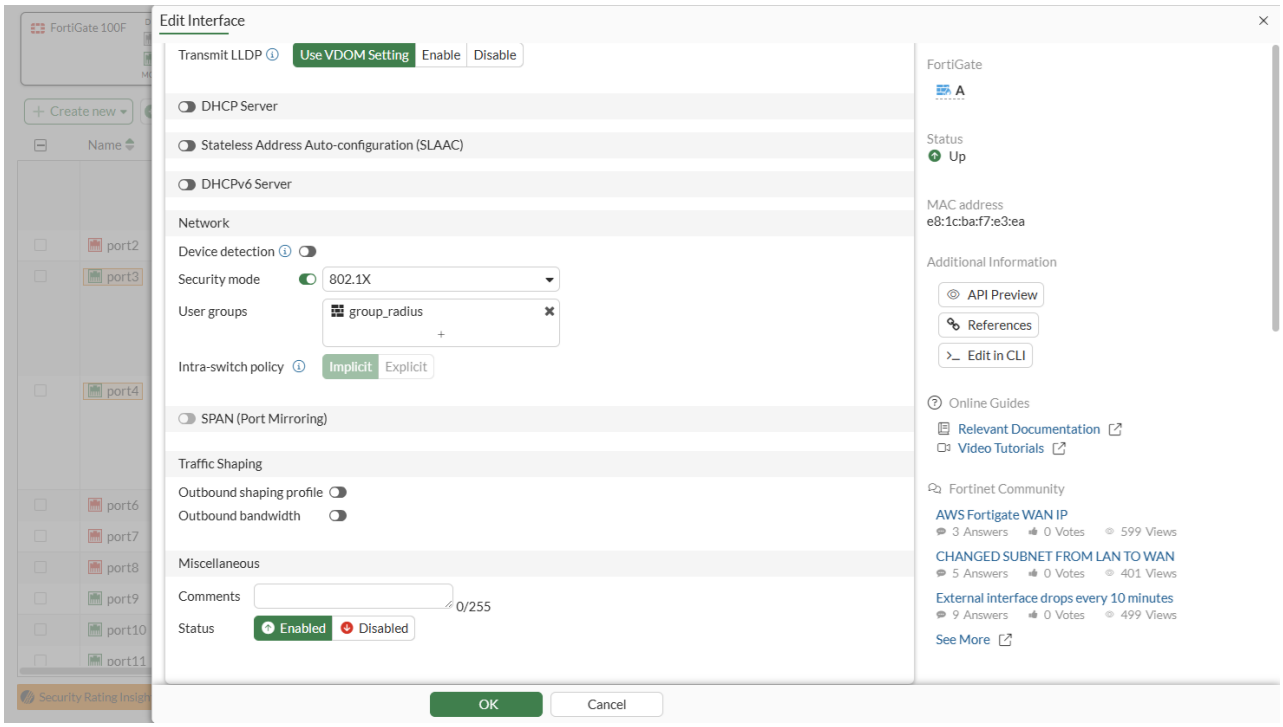
This section describes how to configure a software switch to use 802.1x authentication with a RADIUS user group. Clients use 802.1x authentication to reach the software switch.

To configure a software switch with 802.1X in the GUI:

1. Go to *Network* > *Interfaces*, and create a software switch.



2. Ensure *Security mode* is set to *802.1x* on the software switch and a RADIUS user group is selected.



To configure a software switch with 802.1X in the CLI:

1. Create a software switch interface:

```
config system switch-interface
    edit "sw"
        set vdom "vdom1"
        set member "port5"
        set intra-switch-policy explicit
    next
end
```

2. Enable 802.1x on the software switch:

```
config system interface
    edit "sw"
        set vdom "vdom1"
        set ip 6.6.6.1 255.255.255.0
        set allowaccess ping https ssh snmp fgfm ftm
        set type switch
        set security-mode 802.1X
        set security-groups "group_radius"
        set snmp-index 39
    next
end
```

To test the software switch with 802.1X:

1. Check 802.1x status on the FortiGate. 802.1x is enabled but not authorized.

   This command verifies the operational state of the bridge at the kernel level for the software switch. It confirms that the 802.1X handshake has transitioned the member ports from an `unauthorized` state to an `authorized` state, physically allowing Ethernet frames to pass through the switch fabric.

   ```
   # diagnose netlink brctl name port sw
   show bridge sw data port.
   port5 8021x_enable  peer_dev=0
   ```

2. On the client device, verify that traffic cannot reach software switch:

   ```
   fosqa@switch_pc01:~$ ping 6.6.6.1
   PING 6.6.6.1 (6.6.6.1): 56 data bytes
   ^C--- 6.6.6.1 ping statistics ---
   2 packets transmitted, 0 packets received, 100% packet loss
   ```

3. Perform 802.1x authentication. Authentication is successful.

   ```
   eth2: CTRL-EVENT-CONNECTED - Connection to 01:80:c2:00:00:03 completed [id=0 id_str=]
   EAPOL: SUPP_PAE entering state AUTHENTICATED
   EAPOL: Supplicant port status: Authorized
   EAPOL: SUPP_BE entering state RECEIVE
   EAPOL: SUPP_BE entering state SUCCESS
   EAPOL: SUPP_BE entering state IDLE
   EAPOL authentication completed - result=SUCCESS
   ```

4. On the client device, verify that traffic can reach the software switch:

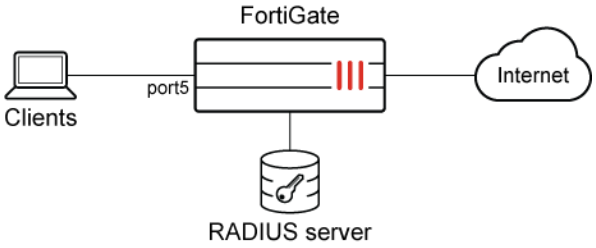
   ```
   fosqa@switch_pc01:~$ ping 6.6.6.1 -c 3
   PING 6.6.6.1 (6.6.6.1): 56 data bytes
   64 bytes from 6.6.6.1: icmp_seq=0 ttl=255 time=0.271 ms
   64 bytes from 6.6.6.1: icmp_seq=1 ttl=255 time=0.214 ms
   64 bytes from 6.6.6.1: icmp_seq=2 ttl=255 time=0.209 ms
   --- 6.6.6.1 ping statistics ---
   3 packets transmitted, 3 packets received, 0% packet loss
   round-trip min/avg/max/stddev = 0.209/0.231/0.271/0.028 ms
   ```

5. On the FortiGate, check the 802.1x status. The port is authorized.

   This command verifies the operational state of the bridge at the kernel level for the software switch. It confirms that the 802.1X handshake has transitioned the member ports from an `unauthorized` state to an `authorized` state, physically allowing Ethernet frames to pass through the switch fabric.

   ```
   # diagnose netlink brctl name port sw
   show bridge sw data port.
   port5 8021x_enable 8021x_authorized  peer_dev=0
   ```

Configuring a software switch with 802.1x MAB



This section describes how to configure the following items:

- Software switch with one port member and `intra-switch-policy` set to `explicit`
- Software switch with 802.1x and MAB enabled and a RADIUS user group selected

Clients use 802.1x and MAB authentication to reach the software switch.

To configure and test a software switch with 802.1X MAB:

1. Create a software switch:

```
config system switch-interface
    edit "sw"
        set vdom "vdom1"
        set member "port5"
        set intra-switch-policy explicit
    next
end
```

2. Enable 802.1x MAB on the software switch:

```
config system interface
    edit "sw"
        set vdom "vdom1"
        set ip 6.6.6.1 255.255.255.0
        set allowaccess ping https ssh snmp fgfm ftm
        set type switch
        set security-mode 802.1X
        set security-mac-auth-bypass enable
        set security-groups "group_radius"
        set snmp-index 39
    next
end
```

3. On client device, verify traffic can reach software switch, which indicates MAB is successful:

```
fosqa@switch_pc01:~$ ping 6.6.6.1 -c 3
PING 6.6.6.1 (6.6.6.1): 56 data bytes
64 bytes from 6.6.6.1: icmp_seq=0 ttl=255 time=0.268 ms
64 bytes from 6.6.6.1: icmp_seq=1 ttl=255 time=0.216 ms
64 bytes from 6.6.6.1: icmp_seq=2 ttl=255 time=0.221 ms
--- 6.6.6.1 ping statistics ---
```

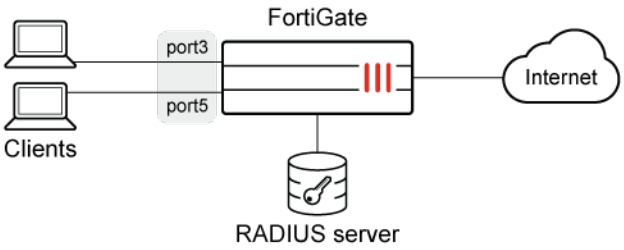
```
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.216/0.235/0.268/0.023 ms
```

4. On FortiGate, check 802.1x status. Port is authorized.

   This command verifies the operational state of the bridge at the kernel level for the software switch. It confirms that the 802.1X handshake has transitioned the member ports from an `unauthorized` state to an `authorized` state, physically allowing Ethernet frames to pass through the switch fabric.

```
# diagnose netlink brctl name port sw
show bridge sw data port.
port5 8021x_enable 8021x_mab  peer_dev=0
```

Configuring a software switch with 802.1x dynamic VLAN



This section describes how to configure two software switches in a dynamic VLAN. 802.1x is enabled on both software switches, and a RADIUS user group is selected.

Clients use 802.1x authentication to reach the software switch.

To configure and test a software switch with 802.1X dynamic VLAN:

1. Create a software switch:

```
config system switch-interface
    edit "untrust"
        set vdom "vdom1"
        set member "port3" "port5"
    next
    edit "vlan0200"
        set vdom "vdom1"
        set intra-switch-policy explicit
    next
end
```

2. Enable 802.1x dynamic VLAN mode on the software switch:

```
config system interface
    edit "untrust"
        set vdom "vdom1"
        set ip 6.6.6.101 255.255.255.0
        set allowaccess ping https ssh
        set type switch
        set security-mode 802.1X
```

```
        set security-8021x-mode dynamic-vlan
        set security-groups "group_radius"
        set snmp-index 39
    next
        edit "vlan0200"
        set vdom "vdom1"
        set ip 6.6.6.1 255.255.255.0
        set allowaccess ping https ssh
        set type switch
        set security-mode 802.1X
        set security-8021x-mode slave
        set security-8021x-master "untrust"
        set security-8021x-dynamic-vlan-id 200
        set snmp-index 40
    next
end
```

3. On the FortiGate, check the 802.1x status. It is enabled but not authorized.

   This command verifies the operational state of the bridge at the kernel level for the software switch. It confirms that the 802.1X handshake has transitioned the member ports from an `unauthorized` state to an `authorized` state, physically allowing Ethernet frames to pass through the switch fabric.

```
# diagnose netlink brctl name port untrust
show bridge untrust data port.
port5 8021x_enable  peer_dev=0
port3 8021x_enable  peer_dev=0
```

```
# diagnose netlink brctl name port vlan0200
show bridge vlan0200 data port.
```

4. On the client device, verify that traffic cannot reach the software switch:

```
fosqa@switch_pc01:~$ ping 6.6.6.1 -c 3
PING 6.6.6.1 (6.6.6.1): 56 data bytes
--- 6.6.6.1 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss
```

```
fosqa@switch_pc01:~$ ping 6.6.6.101 -c 3
PING 6.6.6.101 (6.6.6.101): 56 data bytes
--- 6.6.6.101 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss
```

5. Perform 802.1x authentication. Authentication is successful.

```
eth2: CTRL-EVENT-CONNECTED - Connection to 01:80:c2:00:00:03 completed [id=0 id_str=]
EAPOL: SUPP_PAE entering state AUTHENTICATED
EAPOL: Supplicant port status: Authorized
EAPOL: SUPP_BE entering state RECEIVE
EAPOL: SUPP_BE entering state SUCCESS
EAPOL: SUPP_BE entering state IDLE
EAPOL authentication completed - result=SUCCESS
```

6. On the client device, verify that traffic can reach the software switch:

```
fosqa@switch_pc01:~$ ping 6.6.6.1 -c 3
PING 6.6.6.1 (6.6.6.1): 56 data bytes
64 bytes from 6.6.6.1: icmp_seq=0 ttl=255 time=0.468 ms
64 bytes from 6.6.6.1: icmp_seq=1 ttl=255 time=0.211 ms
64 bytes from 6.6.6.1: icmp_seq=2 ttl=255 time=0.184 ms
--- 6.6.6.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
```

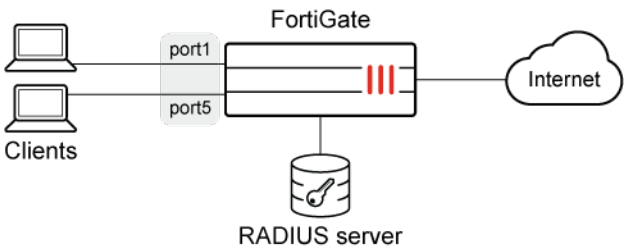
7. On the FortiGate, check the 802.1x status. The port is authorized.

   This command verifies the operational state of the bridge at the kernel level for the software switch. It confirms that the 802.1X handshake has transitioned the member ports from an `unauthorized` state to an `authorized` state, physically allowing Ethernet frames to pass through the switch fabric.

```
# diagnose netlink brctl name port untrust
show bridge untrust data port.
port3 8021x_enable  peer_dev=0
```

```
# diagnose netlink brctl name port vlan0200
show bridge vlan0200 data port.
port5 8021x_enable 8021x_authorized  peer_dev=0
```

Configuring address objects with 802.1x



This section describes how to configure a dynamic VLAN to use 802.1x authentication and a firewall address object assigned to a firewall policy to authorize users by MAC address.

It also describes how to configure a default VLAN to achieve the same results.

This section covers how to:

- Create a software switch interface.
- Configure a RADIUS server and user group.
- On the software switch, enable 802.1x, MAB, dynamic VLAN, and select the RADIUS user group.
- Create a firewall address for the software switch with `type` set to `dynamic` and `sub-type` set to `802.1x Authenticated Address`.

  > The name of the address object must match the `Fortinet-Group-Name` AVP returned by the RADIUS server.

- Use the firewall address as the `srcaddr` in a firewall policy.

With this configuration, users are authorized by 802.1x and MAC address.

A second example is provided to describe how to configure a default VLAN for the same results.

To configure dynamic VLAN mode in the CLI:

1. Create software switch switches.
   In this example, the following software switches are created: 8021, slave300, slave200.

```
config system switch-interface
    edit "8021"
        set vdom "root"
        set member "port1" "port5"
    next
    edit "slave300"
        set vdom "root"
    next
edit "slave200"
        set vdom "root"
    next
end
```

2. Configure a RADIUS server and user group:
   Server is set to the IP address for the RADIUS server.

```
config user radius
    edit "170lab"
        set server "172.16.116.185"
        set secret 123456
    next
end
config user group
    edit "group-rad"
        set member "170lab"
    next
end
```

3. Configure the software switches.
   The key settings for the master software switch are to enable 802.1X security mode, MAB mode, and dynamic-VLAN. On the slave software switches, enable 802.1X security mode.

```
 config system interface
    edit "8021"
        set vdom "root"
        set type switch
        set security-mode 802.1X
        set security-mac-auth-bypass enable
        set security-8021x-mode dynamic-vlan
        set security-groups "rad"
        set snmp-index 15
    next
edit "slave300"
        set vdom "root"
        set type switch
```

```
        set security-mode 802.1X
        set security-8021x-mode slave
        set security-8021x-master "8021"
        set security-8021x-dynamic-vlan-id 300
        set snmp-index 19
    next
edit "slave200"
        set vdom "root"
        set ip 10.100.100.1 255.255.255.0
        set allowaccess ping https ssh snmp http telnet
        set type switch
        set security-mode 802.1X
        set security-8021x-mode slave
        set security-8021x-master "8021"
        set security-8021x-dynamic-vlan-id 200
        set snmp-index 19
    next
end
```

4. Create a firewall address object for the software switch.
   The address object name must match the `Fortinet-Group-Name` AVP returned by the RADIUS server. This example uses `DB-GRP300`.

```
config firewall address
    edit "DB-GRP300"
        set type dynamic
        set sub-type 8021x
    next
edit "ubuntu-desk"
        set uuid f36bf56e-046d-51f1-8fb8-1cb1e9d30081
        set type dynamic
        set sub-type 8021x
    next
end
```

5. Use the firewall address in a firewall policy.
   The firewall address `DB-GRP300` is used as the `srcaddr`.

```
config firewall policy
    edit 2
        set name "802.1x-policy"
        set uuid b81c4a8a-04f5-51f1-f68b-4ea3ecc40fb6
        set srcintf "slave200"
        set dstintf "any"
        set action accept
        set srcaddr "DB-GRP300"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

To test dynamic VLAN mode:

1. Connect a client and wait for authorization to complete.

   This command verifies the authenticated session state of a client. It confirms the FortiGate has successfully validated the client's MAC address through RADIUS and has mapped that device to a specific FortiGate User Group, granting it entry into the firewall's session table.

   ```
   # diagnose firewall auth mac list

   0c:37:96:c3:66:ca, 0C-37-96-C3-66-CA
           type: fw, id: 0, duration: 109, idled: 109
           flag(10): radius
           server: 170
           packets: in 0 out 0, bytes: in 0 out 0
           group_id: 2
           group_name: rad

   48:3a:02:cd:f0:0e, 48-3A-02-CD-F0-0E
           type: fw, id: 0, duration: 127, idled: 127
           flag(10): radius
           server: 170
           packets: in 0 out 0, bytes: in 0 out 0
           group_id: 2
           group_name: rad

   ----- 2 listed, 0 filtered ------
   ```

   This command verifies the operational state of the bridge at the kernel level for the software switch. It confirms that the 802.1X handshake has transitioned the member ports from an `unauthorized` state to an `authorized` state, physically allowing Ethernet frames to pass through the switch fabric

   ```
   # diagnose netlink brctl name port slave200
   show bridge slave200 data port.
   port5 8021x_enable 8021x_mab 8021x_authorized  peer_dev=0
   ```

   ```
   # diagnose netlink brctl name port slave300
   show bridge slave300 data port.
   port1 8021x_enable 8021x_mab 8021x_authorized  peer_dev=0
   ```

2. Dump firewall address.

   This command verifies the population of the dynamic firewall address objects created by the administrator. It confirms that the FortiGate has successfully matched the RADIUS attributes to your manually defined objects and has dynamically bound the client's MAC address and assigned VLAN to those objects for policy enforcement.

   ```
   # diagnose sys 802-1x-addr dump
   name: DB-GRP300
   addr-8021x (1):
           interface: port1
                   MAC: 48:3a:02:cd:f0:0e
                   Acct-User: 48-3A-02-CD-F0-0E
                   IP: 0.0.0.0
   ```

```
                VLAN ID: 300
name: ubuntu-desk
addr-8021x (1):
        interface: port5
                MAC: 0c:37:96:c3:66:ca
                Acct-User: 0C-37-96-C3-66-CA
                IP: 0.0.0.0
                VLAN ID: 200
total 2
```

3. From the client, send traffic to the software switch.
   On the client side, view the authorized MAC address:

```
PC3:/$ ifconfig

usbeth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.100.100.7  netmask 255.255.255.0  broadcast 10.100.100.255
        ether 0c:37:96:c3:66:ca  txqueuelen 1000  (Ethernet)
        RX packets 11662798  bytes 1495115443 (1.4 GB)
        RX errors 0  dropped 4  overruns 0  frame 0
        TX packets 911354  bytes 118307806 (118.3 MB)
        TX errors 0  dropped 1117 overruns 0  carrier 0  collisions 0
```

   Before authorization:

```
-PC3:/$ ping 10.100.100.1
From 10.100.100.7 icmp_seq=41734 Destination Host Unreachable
From 10.100.100.7 icmp_seq=41735 Destination Host Unreachable
From 10.100.100.7 icmp_seq=41736 Destination Host Unreachable
```

   After authorization:

```
-PC3:/$ ping 10.100.100.1
PING 10.100.100.1 (10.100.100.1) 56(84) bytes of data.
64 bytes from 10.100.100.1: icmp_seq=1 ttl=255 time=0.367 ms
64 bytes from 10.100.100.1: icmp_seq=2 ttl=255 time=0.352 ms
64 bytes from 10.100.100.1: icmp_seq=3 ttl=255 time=0.334 ms
64 bytes from 10.100.100.1: icmp_seq=4 ttl=255 time=0.276 ms
^C
--- 10.100.100.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3054ms
rtt min/avg/max/mdev = 0.276/0.332/0.367/0.034 ms
```

4. After successful authentication, confirm the firewall policy allows traffic through FortiGate.

```
# diagnose firewall iprope list 100004

policy index=2 uuid_idx=667 action=accept
flag (8010008): redir master pol_stats
flag2 (4000): resolve_sso
flag3 (100000a0): link-local best-route no-vwp
schedule(always)
```

```
cos_fwd=255  cos_rev=255
group=00100004 av=00004e20 au=00000000 split=00000000
host=0 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 27 -> zone(1): 0
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=551,
source MAC address(2):
        48:3a:02:cd:f0:0e-48:3a:02:cd:f0:0e, uuid_idx=666
        0c:37:96:c3:66:ca-0c:37:96:c3:66:ca, uuid_idx=666
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto

3.on Client, ping any address pass through FGT could seccess
PC3:/$ ping 192.168.1.254
PING 192.168.1.254 (192.168.1.254) 56(84) bytes of data.
64 bytes from 192.168.1.254: icmp_seq=1 ttl=255 time=0.401 ms
64 bytes from 192.168.1.254: icmp_seq=2 ttl=255 time=0.351 ms
64 bytes from 192.168.1.254: icmp_seq=3 ttl=255 time=0.291 ms
^C
--- 192.168.1.254 ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 2086ms
rtt min/avg/max/mdev = 0.291/0.347/0.401/0.044 ms
```

To configure the default mode in the CLI:

1. Create a software switch.
2. Edit the software switch, add members, and set `intra-switch-policy` to `explicit`.

```
config system switch-interface
    edit "8021"
        set vdom "root"
        set member "port1" "port5"
        set intra-switch-policy explicit
    next
end
```

3. On the software switch, enable 802.1X security mode and MAB mode.

```
config system interface
      edit "8021"
        set vdom "root"
        set ip 10.100.100.1 255.255.255.0
        set allowaccess ping https ssh snmp http
        set type switch
        set security-mode 802.1X
        set security-mac-auth-bypass enable
        set security-groups "rad"
        set snmp-index 15
    next
end
```

4. Create a firewall address.

The address object name must match the `Fortinet-Group-Name` AVP returned by the RADIUS server. This example uses `DB-GRP300`.

```
config firewall address
    edit "DB-GRP300"
        set type dynamic
        set sub-type 8021x
    next
edit "ubuntu-desk"
        set uuid f36bf56e-046d-51f1-8fb8-1cb1e9d30081
        set type dynamic
        set sub-type 8021x
    next
end
```

5. Use the firewall address in a firewall policy.
   The firewall address `DB-GRP300` is used as the `srcaddr`.

```
config firewall policy
    edit 2
        set name "802.1x-policy"
        set uuid b81c4a8a-04f5-51f1-f68b-4ea3ecc40fb6
        set srcintf "slave200"
        set dstintf "any"
        set action accept
        set srcaddr "DB-GRP300"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

To test the default mode:

1. Connect the client and wait for authorization to complete.
   This command verifies the authenticated session state of a client. It confirms the FortiGate has successfully validated the client's MAC address through RADIUS and has mapped that device to a specific FortiGate User Group, granting it entry into the firewall's session table.

```
# diagnose firewall auth mac list
0c:37:96:c3:66:ca, 0C-37-96-C3-66-CA
        type: fw, id: 0, duration: 115, idled: 115
        flag(10): radius
        server: 170
        packets: in 0 out 0, bytes: in 0 out 0
        group_id: 2
        group_name: rad
48:3a:02:cd:f0:0e, 48-3A-02-CD-F0-0E
        type: fw, id: 0, duration: 141, idled: 141
        flag(10): radius
        server: 170
```

```
        packets: in 0 out 0, bytes: in 0 out 0
        group_id: 2
        group_name: rad
----- 2 listed, 0 filtered ------
```

This command verifies the operational state of the bridge at the kernel level for the software switch. It confirms that the 802.1X handshake has transitioned the member ports from an `unauthorized` state to an `authorized` state, physically allowing Ethernet frames to pass through the switch fabric.

```
# diagnose netlink brctl name port 8021
show bridge 8021 data port.
port5 8021x_enable 8021x_mab 8021x_authorized  peer_dev=0
port1 8021x_enable 8021x_mab 8021x_authorized  peer_dev=0
```

2. Dump firewall address.

   This command verifies the population of the dynamic firewall address objects created by the administrator. It confirms that the FortiGate has successfully matched the RADIUS attributes to your manually defined objects and has dynamically bound the client's MAC address and assigned VLAN to those objects for policy enforcement.

```
# diagnose sys 802-1x-addr dump
name: DB-GRP300
addr-8021x (1):
        interface: port1
                MAC: 48:3a:02:cd:f0:0e
                Acct-User: 48-3A-02-CD-F0-0E
                IP: 0.0.0.0
                VLAN ID: 300
name: ubuntu-desk
addr-8021x (1):
        interface: port5
                MAC: 0c:37:96:c3:66:ca
                Acct-User: 0C-37-96-C3-66-CA
                IP: 0.0.0.0
                VLAN ID: 200
total 2
```

3. From the client, send traffic to the software switch.

   The client before authorization:

```
-PC3:/$ ping 10.100.100.1
From 10.100.100.7 icmp_seq=41734 Destination Host Unreachable
From 10.100.100.7 icmp_seq=41735 Destination Host Unreachable
From 10.100.100.7 icmp_seq=41736 Destination Host Unreachable
```

   The client after authorization:

```
-PC3:/$ ping 10.100.100.1
PING 10.100.100.1 (10.100.100.1) 56(84) bytes of data.
64 bytes from 10.100.100.1: icmp_seq=1 ttl=255 time=0.367 ms
64 bytes from 10.100.100.1: icmp_seq=2 ttl=255 time=0.352 ms
64 bytes from 10.100.100.1: icmp_seq=3 ttl=255 time=0.334 ms
```

```
64 bytes from 10.100.100.1: icmp_seq=4 ttl=255 time=0.276 ms
^C
--- 10.100.100.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3054ms
rtt min/avg/max/mdev = 0.276/0.332/0.367/0.034 ms
```

4. After successful authentication, confirm the firewall policy allows traffic through FortiGate.

```
# diagnose firewall iprope list 100004

policy index=2 uuid_idx=667 action=accept
flag (8010008): redir master pol_stats
flag2 (4000): resolve_sso
flag3 (100000a0): link-local best-route no-vwp
schedule(always)
cos_fwd=255  cos_rev=255
group=00100004 av=00004e20 au=00000000 split=00000000
host=0 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 27 -> zone(1): 0
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=551,
source MAC address(2):
        48:3a:02:cd:f0:0e-48:3a:02:cd:f0:0e, uuid_idx=666
        0c:37:96:c3:66:ca-0c:37:96:c3:66:ca, uuid_idx=666
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto

3.on Client, ping any address pass through FGT could seccess
PC3:/$ ping 192.168.1.254
PING 192.168.1.254 (192.168.1.254) 56(84) bytes of data.
64 bytes from 192.168.1.254: icmp_seq=1 ttl=255 time=0.401 ms
64 bytes from 192.168.1.254: icmp_seq=2 ttl=255 time=0.351 ms
64 bytes from 192.168.1.254: icmp_seq=3 ttl=255 time=0.291 ms
^C
--- 192.168.1.254 ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 2086ms
rtt min/avg/max/mdev = 0.291/0.347/0.401/0.044 ms
```

Dynamic BGP learning of ISDB reputation IPs

Dynamic BGP-based learning for Botnet and Spam ISDB categories is added, allowing FortiGate to automatically receive and advertise these IPs for more responsive, threat-aware routing.

```
config router bgp
    config network
        edit <id>
            set internet-service-name {Botnet-C&C.Server | Spam-Spamming.Server}
        next
```

```
    end
end
```

To test using the Botnet-C&C.Server internet service:

1. Configure BGP:

```
config router bgp
    set as 65412
    set router-id 2.2.2.2
    set network-import-check disable
    config neighbor
        edit "3.3.3.3"
            set advertisement-interval 5
            set capability-graceful-restart enable
            set ebgp-enforce-multihop enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65412
            set route-map-out "as-prepend"
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
        next
    end
    config network
        edit 1
            set prefix 172.28.1.0 255.255.255.0
        next
        edit 2
            set prefix 172.28.2.0 255.255.255.0
        next
        edit 3
            set prefix 172.25.1.0 255.255.255.0
        next
        edit 4
            set internet-service-name "Botnet-C&C.Server"
        next
    end
    config network6
        edit 1
            set prefix6 2000:172:27:1::/64
        next
    end
end
```

2. Check the internet service ID:

```
# diagnose internet-service id 3080383
...
222.158.197.138-222.158.197.138 country(392) region(65535) city(65535) blocklist(0x1)
reputation(1), popularity(1) domain(0) botnet(7630624) proto(6) port(80)
```

```
222.165.194.68-222.165.194.68 country(360) region(65535) city(65535) blocklist(0x1) reputation
(1), popularity(1) domain(0) botnet(7630624) proto(6) port(50486)
222.165.205.154-222.165.205.154 country(360) region(65535) city(65535) blocklist(0x1)
reputation(1), popularity(1) domain(0) botnet(7630624) proto(6) port(8089)
222.173.92.154-222.173.92.154 country(156) region(65535) city(65535) blocklist(0x1) reputation
(1), popularity(1) domain(0) botnet(7630624) proto(6) port(32075)
222.191.243.187-222.191.243.187 country(156) region(65535) city(65535) blocklist(0x1)
reputation(1), popularity(1) domain(0) botnet(7630624) proto(6) port(45730)
222.211.72.29-222.211.72.29 country(156) region(65535) city(65535) blocklist(0x1) reputation
(1), popularity(1) domain(0) botnet(7630020) proto(6) port(80)
222.217.68.17-222.217.68.17 country(156) region(65535) city(65535) blocklist(0x1) reputation
(1), popularity(1) domain(0) botnet(7630624) proto(6) port(35165)
222.252.23.5-222.252.23.5 country(704) region(65535) city(65535) blocklist(0x1) reputation(1),
popularity(1) domain(0) botnet(7630624) proto(6) port(8080)
223.100.166.3-223.100.166.3 country(156) region(65535) city(65535) blocklist(0x1) reputation
(1), popularity(1) domain(0) botnet(7630624) proto(6) port(49247)
223.165.243.209-223.165.243.209 country(410) region(65535) city(65535) blocklist(0x1)
reputation(1), popularity(1) domain(0) botnet(7630624) proto(6) port(47205)
```

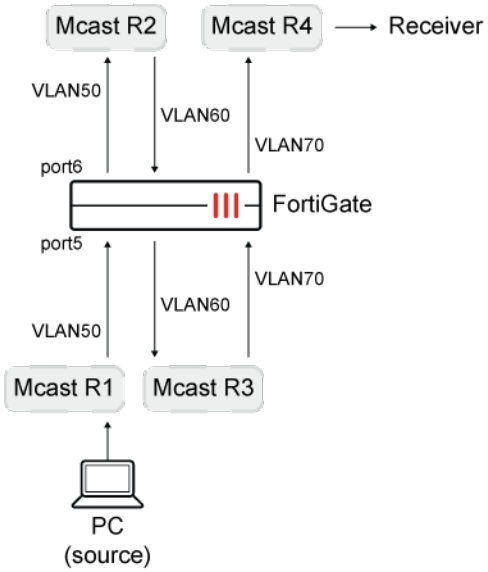
3. On the BGP peer, check the networks/subnets of this internet service:

```
# get router info routing-table bgp
...
B       221.194.47.218/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       221.229.204.124/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.129.33.141/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.129.35.9/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.158.197.138/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.165.194.68/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.165.205.154/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.173.92.154/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.191.243.187/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.211.72.29/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.217.68.17/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.252.23.5/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       223.100.166.3/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       223.165.243.209/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:14,
[1/0]
```

Enhanced multicast session key for virtual wire pair

FortiOS now adds ingress interface and VLAN ID as additional session keys for multicast traffic in virtual wire pair (VWP) configurations with wildcard-vlan enabled. This enhancement enables the FortiGate to create distinct multicast sessions for traffic returning through the same VWP on different VLANs, improving accuracy and ensuring optimal session handling.

Example



In this example, a VWP with wildcard-vlan enabled is created. A firewall policy and multicast policy are created. Multicast traffic goes through FortiGate multiple times in different directions, and distinct multicast sessions are created for traffic returning through the same VWP on different VLANs.

To configure:

1. Configure a virtual wire pair:
   The VWP contains members a and b, and *wildcard-vlan* is enabled.

```
config system virtual-wire-pair
    edit "vwp"
        set member "a" "b"
        set wildcard-vlan enable
    next
end
```

2. Configure a firewall policy:
   The firewall policy specifies VWP members a and b as source and destination interfaces.

```
config firewall policy
    edit 1
```

```
        set uuid dfb4861a-f6ff-51f0-888b-682470ce488d
        set srcintf "a" "b"
        set dstintf "b" "a"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

3. Configure a multicast policy:

```
config firewall multicast-policy
    edit 1
        set uuid 9d6c29a4-f720-51f0-f60b-07a30ec0aaac
        set srcintf "any"
        set dstintf "any"
        set srcaddr "all"
        set dstaddr "all"
        set logtraffic all
    next
end
```

4. After multicast traffic flows, view the different multicast sessions in traffic logs:

```
1: date=2026-01-21 time=15:38:42 eventtime=1769038717948452847 tz="-0800" logid="0002000012"
type="traffic" subtype="multicast" level="notice" vd="root" srcip=10.1.1.1 identifier=13
srcintf="b" srcintfrole="undefined" dstip=225.1.1.1 dstintf="a" dstintfrole="undefined"
vlan=60 srccountry="Reserved" dstcountry="Reserved" sessionid=4089 proto=1 action="accept"
policyid=1 policytype="multicast-policy" poluuid="9d6c29a4-f720-51f0-f60b-07a30ec0aaac"
service="PING" trandisp="noop" appcat="unscanned" duration=181 sentbyte=168 rcvdbyte=0
sentpkt=2 rcvdpkt=0

2: date=2026-01-21 time=15:38:42 eventtime=1769038717948449960 tz="-0800" logid="0002000012"
type="traffic" subtype="multicast" level="notice" vd="root" srcip=10.1.1.1 identifier=13
srcintf="a" srcintfrole="undefined" dstip=225.1.1.1 dstintf="b" dstintfrole="undefined"
vlan=50 srccountry="Reserved" dstcountry="Reserved" sessionid=4088 proto=1 action="accept"
policyid=1 policytype="multicast-policy" poluuid="9d6c29a4-f720-51f0-f60b-07a30ec0aaac"
service="PING" trandisp="noop" appcat="unscanned" duration=181 sentbyte=168 rcvdbyte=0
sentpkt=2 rcvdpkt=0

3: date=2026-01-21 time=15:38:42 eventtime=1769038717940450742 tz="-0800" logid="0002000012"
type="traffic" subtype="multicast" level="notice" vd="root" srcip=10.1.1.1 identifier=13
srcintf="a" srcintfrole="undefined" dstip=225.1.1.1 dstintf="b" dstintfrole="undefined"
vlan=70 srccountry="Reserved" dstcountry="Reserved" sessionid=4090 proto=1 action="accept"
policyid=1 policytype="multicast-policy" poluuid="9d6c29a4-f720-51f0-f60b-07a30ec0aaac"
service="PING" trandisp="noop" appcat="unscanned" duration=181 sentbyte=168 rcvdbyte=0
sentpkt=2 rcvdpkt=0
```

5. View the different multicast sessions using the `diagnose sys mcast-session list` command:

```
# diagnose sys mcast-session list

session info: id=4264 vf=0 vrf=0 proto=1 10.1.1.1.16->225.1.1.1.8
used=2 path=1 duration=167 expire=13 indev=11 vlanid=50 pkts=2 bytes=168
state=0000001a: tp npu-cap offloaded
session-npu-info: ipid/vlifid=9/9 vlanid/vtag_in=50/50 in_npuid=1 tae_index=0 qid=1 fwd_
map=0x00000000
path: log offloaded policy=1, outdev=12, tos=0xff
act-npu-info:  ipid/vlifid=9/9 vlanid/vtag_in=50/50 in_npu_id=1, out_npuid=1 epid=10 fwd=0

session info: id=4265 vf=0 vrf=0 proto=1 10.1.1.1.16->225.1.1.1.8
used=2 path=1 duration=167 expire=13 indev=12 vlanid=60 pkts=2 bytes=168
state=0000001a: tp npu-cap offloaded
session-npu-info: ipid/vlifid=10/10 vlanid/vtag_in=60/60 in_npuid=1 tae_index=0 qid=2 fwd_
map=0x00000000
path: log offloaded policy=1, outdev=11, tos=0xff
act-npu-info:  ipid/vlifid=10/10 vlanid/vtag_in=60/60 in_npu_id=1, out_npuid=1 epid=9 fwd=0

session info: id=4266 vf=0 vrf=0 proto=1 10.1.1.1.16->225.1.1.1.8
used=2 path=1 duration=167 expire=13 indev=11 vlanid=70 pkts=2 bytes=168
state=0000001a: tp npu-cap offloaded
session-npu-info: ipid/vlifid=9/9 vlanid/vtag_in=70/70 in_npuid=1 tae_index=0 qid=7 fwd_
map=0x00000000
path: log offloaded policy=1, outdev=12, tos=0xff
act-npu-info:  ipid/vlifid=9/9 vlanid/vtag_in=70/70 in_npu_id=1, out_npuid=1 epid=10 fwd=0
Total 3 sessions
```

Next-hop self-support for VPNv4 and VPNv6 route reflectors

FortiGate now supports setting the BGP next-hop attribute for VPNv4 and VPNv6 routes on route reflectors.

The `next-hop-self-rr-vpnv4` and `next-hop-self-rr-vpnv6` commands allow route reflectors to set the NEXT_HOP attribute to the local interface address when advertising VPNv4 or VPNv6 routes. This enhancement extends the existing `next-hop-self-rr behavior` that previously applied only to IPv4 and IPv6 unicast routes.

The feature can be configured at both the BGP neighbor and neighbor group levels, providing flexibility for large-scale MPLS VPN and route reflector deployments.

```
config router bgp
    config neighbor
        edit <ip>
            set next-hop-self-rr-vpnv4 {enable | disable}
            set next-hop-self-rr-vpnv6 {enable | disable}
        next
    end
    config neighbor-group
        edit <name>
            set next-hop-self-rr-vpnv4 {enable | disable}
            set next-hop-self-rr-vpnv6 {enable | disable}
```

```
        next
    end
end
```

Command	Description
next-hop-self-rr-vpnv4 {enable | disable}	Enable/disable setting of the nexthop's address to interface's address for route-reflector VPNv4 routes.
next-hop-self-rr-vpnv6 {enable | disable}	Enable/disable setting of the nexthop's address to interface's address for route-reflector VPNv6 routes.

IPv6

This section includes information about IPv6 related new features:

- IPv6 Link-local configuration enhancement on page 122
- IPv6 proxy address and address group object support on page 123

IPv6 Link-local configuration enhancement

You can now configure `ip6-link-local` directly in an interface's IPv6 settings. When a manual link-local address is applied, the system automatically removes the default auto-generated IPv6 link-local address, preventing multiple link-local addresses from co-existing.

```
config system interface
    edit <port>
        config ipv6
            set ip6-link-local <IPv6 prefix>
        end
    next
end
```

Command	Description
`ip6-link-local <IPv6 prefix>`	IPv6 link-local address of interface.

1. Set the IPv6 link-local address of port1 and then check the address list:

```
config system interface
    edit port1
        config ipv6
            set ip6-address 2000:172:16:200::1/64
            set ip6-allowaccess ping
            set ip6-link-local fe80::1111/64
        end
```

```
    next
end
```

```
# diagnose ipv6 address list | grep port1
dev=7 devname=port1 flag=P scope=0 prefix=64 addr=2000:172:16:200::1 preferred=4294967295
valid=4294967295 cstamp=10042 tstamp=10042
dev=7 devname=port1 flag=P scope=253 prefix=64 addr=fe80::1111 preferred=4294967295
valid=4294967295 cstamp=260034 tstamp=260034
```

2. Unset the IPv6 link-local address and then check the address list:

```
config system interface
    edit port1
        config ipv6
            unset ip6-link-local
        end
    next
end
```

```
# diagnose ipv6 address list | grep port1
dev=7 devname=port1 flag=P scope=0 prefix=64 addr=2000:172:16:200::1 preferred=4294967295
valid=4294967295 cstamp=10042 tstamp=10042
dev=7 devname=port1 flag=TP scope=253 prefix=64 addr=fe80::96f3:92ff:fe15:1ecc
preferred=4294967295 valid=4294967295 cstamp=270370 tstamp=270370
```

IPv6 proxy address and address group object support

FortiGate now supports IPv6 proxy address and proxy address group objects, enabling proxy-based policy matching for IPv6 traffic.

IPv6 proxy address and address group objects can be configured using the CLI and used as IPv6 address objects in proxy policies and authentication rules. They provide the same matching capabilities as IPv4 proxy address and address group objects, except that the host definitions use IPv6 firewall address objects.

IPv6 proxy address and address group objects can be applied to both the `srcaddr6` and `dstaddr6` options, allowing granular IPv6 traffic matching based on URL, header, method, user agent, category, SaaS application, and other proxy criteria.

IPv6 proxy address and address group objects:

```
config firewall proxy-address6
```

```
config firewall proxy-addrgrp6
```

IPv6 proxy objects in proxy policies:

```
config firewall proxy-policy
    set srcaddr6 {proxy-address6 | proxy-addrgrp6}
```

```
    set dstaddr6 {proxy-address6 | proxy-addrgrp6}
end
```

IPv6 proxy objects in authentication rules:

```
config authentication rule
    set srcaddr6 {proxy-address6 | proxy-addrgrp6}
    set dstaddr6 {proxy-address6 | proxy-addrgrp6}
end
```

Example

To configure an IPv6 proxy address object and use it in a proxy policy:

1. Configure an IPv6 proxy address:

```
config firewall proxy-address6
    edit "ipv6-proxy-url"
        set type url
        set path "/example"
    next
end
```

2. Apply the IPv6 proxy address to a proxy policy:

```
config firewall proxy-policy
    edit 1
        set srcaddr6 "ipv6-proxy-url"
        set dstaddr6 "all"
    next
end
```

Explicit and transparent proxy

This section includes information about explicit and transparent proxy related new features:

- Secure explicit proxy with client certificate blocklist enforcement on page 125

Secure explicit proxy with client certificate blocklist enforcement

This information is also available in the FortiOS 8.0 Administration Guide:
- Secure explicit proxy with client certificate blocklist enforcement

Client certificate blocklist enforcement is now supported. FortiGate ensures that secure explicit proxy connections are protected against known malicious client certificates listed in the Malicious Certificate Database (MCDB), improving security posture and compliance.

Overview

Secure explicit proxy with client certificates allows FortiGate to authenticate proxy clients using X.509 certificates. When client certificate blocklist, enforcement is enabled, FortiGate checks the presented client certificate against the MCDB during the TLS handshake.

If the client certificate is listed in the MCDB:

- The TLS handshake is aborted
- The proxy connection is denied
- A log entry is generated

This prevents malicious certificates from accessing web resources through the proxy.

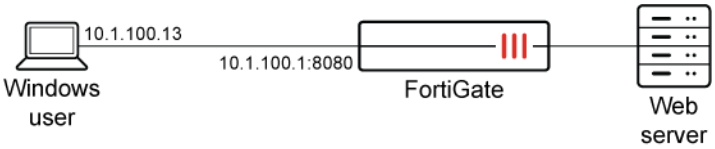
The config web-proxy explicit command includes a new option:

```
config web-proxy explicit
    set client-certificate-blocklist {enable | disable}
end
```

Option	Description
`client-certificate-blocklist {enable | disable}`	Enable/disable blocking client malicious certificates list by FortiGuard during TLS (default = enabled). • `enable`: Enable client certificate blocklist during TLS handshake. • `disable`: Disable client certificate blocklist during TLS handshake.

Example

The following example demonstrates client certificate blocklist enforcement using a secure explicit proxy.



Prerequisite:

An active FortiGuard subscription is required to use the Malicious Certificate Database (MCDB). The MCDB is used by FortiGate's SSL/TLS inspection and certificate reputation checks to detect and block SSL/TLS sessions that present malicious certificates.

To configure client certificate and blocklist enforcement with explicit proxies:

1. Prepare the certificate:
   a. Use a CA to sign the client certificate.
   b. Import the root CA certificate that signed the client certificate to FortiGate. In this scenario, the certificate is `root_ca`.
   c. Install the client certificate on an endpoint.
2. Configure the explicit web-proxy policy to request the client certificate from the endpoint.

```
config web-proxy explicit
    set status enable
    set secure-web-proxy secure
    set http-incoming-port 8080
    set secure-web-proxy-cert "proxyserver"
    set client-cert enable
    set client-certificate-blocklist enable
end
```

3. Configure verification of the client certificate with the root CA.

```
config authentication setting
    set user-cert-ca "root_ca"
end
```

4. Configure explicit proxy policy.

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set srcaddr "all"
        set dstaddr "all"
        set service "web"
        set action accept
        set schedule "always"
        set logtraffic all
        set utm-status enable
        set ssl-ssh-profile "deep-custom"
    next
end
```

Verification

On Client PC, initiate a connection with blocklisted client certificate:

```
# curl -v -k --proxy-insecure --proxy-cert /root/CA/client_blocklist.p12:123456 --proxy-cert-type
P12 -x https://10.1.100.1:8080 http://172.16.200.99
```

TLS handshake fails, connection is denied, and following log is generated on the FortiGate:

```
1: date=2026-04-02 time=17:33:31 eventtime=1775176410797092890 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.13 srcport=53684
srcintf="port2" srcintfrole="undefined" dstip=10.1.100.1 dstport=8080 dstintf="unknown-0"
dstintfrole="undefined" srccountry="Reserved" dstcountry="Reserved" sessionid=268435461 proto=6
action="deny" policyid=0 policytype="policy" service="tcp/8080" trandisp="noop" appcat="unscanned"
duration=0 sentbyte=1965 rcvdbyte=0 sentpkt=0 rcvdpkt=0 crscore=30 craction=131072 crlevel="high"
msg="Traffic denied because of client certificate blocklisted"
```

SD-WAN

This section includes information about SD-WAN related new features:

- Overlays and underlays on page 128
- Performance SLA on page 187

Overlays and underlays

This section includes information about overlay and underlay related new features:

- SD-WAN underlay bandwidth steers traffic on page 128
- ToS-based SD-WAN duplication matching on page 135
- Stop packet duplication upon reaching bandwidth utilization threshold on page 141
- Selective SD-WAN duplication on page 147
- Use secondary tunnel for FEC redundant parity packets on page 152
- ToS matching and negate options on adaptive FEC profiles on page 159
- Control SD-WAN interface usage based on monthly traffic volume (quota) on page 166
- Confine packet duplication across multiple overlays to the same spoke on page 173
- On-demand duplication at the Hub using remote health-check on page 178

SD-WAN underlay bandwidth steers traffic

A new option is available to allow load-balance service with hash-mode inbandwidth, outbandwidth, or bibandwidth to steer traffic based on shared underlay available bandwidth or individual overlay available bandwidth.

The `config service command` includes new options:

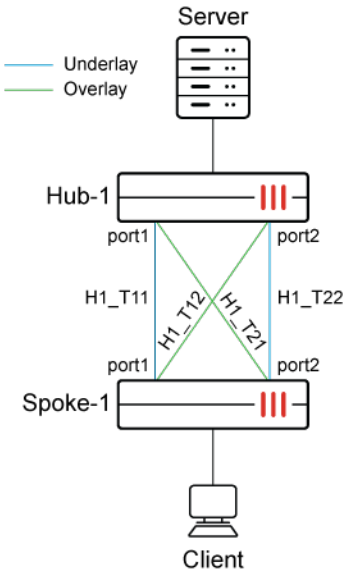
```
config system sdwan
    config service
        edit < num>
          set load-balance enable
          set hash-mode {round-robin | source-ip-based | source-dest-ip-based | inbandwidth |
outbandwidth | bibandwidth}
          set bandwidth-type {overlay* | underlay}
        next
    end
end
```

Option	Description
`hash-mode {round-robin | source-ip-based | source-dest-ip-based | inbandwidth | outbandwidth | bibandwidth}`	Hash algorithm for selected priority members for load-balance mode. Available when `load-balance` is enabled. Choose from: • `inbandwidth`: All traffic are distributed to a selected interface with most available bandwidth for incoming traffic. • `outbandwidth`: All traffic are distributed to a selected interface with most available bandwidth for outgoing traffic. • `bibandwidth`: All traffic are distributed to a selected interface with most available bandwidth for both incoming and outgoing traffic.
`bandwidth-type {overlay* | underlay}`	Overlay/underlay type of bandwidth (default = overlay).

Example

This example demonstrates how to use the underlay bandwidth to make traffic steering decisions for SD-WAN. In summary:

- Spoke-1 has four overlays: H1_T11, H1_T12, H1_T21, and H1_T22.
- H1_T11 and H1_T12 are associated with the same underlay port1, and H1_T21 and H1_T22 are associated with the same underlay port2.
- The load-balance service with `hash-mode bibandwidth` is configured on Spoke-1 to steer traffic on all four overlays.
- The overlay that is associated with the underlay with most available bandwidth is intended to carry traffic.



To configure Spoke-1:

1. Set the estimated bandwidth for upstream and downstream direction on each underlay and overlay:

```
config system interface
    edit "port1"
        ...
        set estimated-upstream-bandwidth 100000
        set estimated-downstream-bandwidth 100000
        ...
    next
    edit "port2"
        ...
        set estimated-upstream-bandwidth 50000
        set estimated-downstream-bandwidth 50000
        ...
    next
    edit "H1_T11"
        ...
        set estimated-upstream-bandwidth 90000
        set estimated-downstream-bandwidth 90000
         ...
    next
    edit "H1_T12"

        set estimated-upstream-bandwidth 90000
        set estimated-downstream-bandwidth 90000

    next
    edit "H1_T21"
        ...
        set estimated-upstream-bandwidth 40000
        set estimated-downstream-bandwidth 40000

    next
    edit "H1_T22"
        ...
        set estimated-upstream-bandwidth 40000
        set estimated-downstream-bandwidth 40000
        ...
    next
end
```

2. Configure SD-WAN with load-balancing enabled and `hash-mode` set to `bibandwidth`:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "overlay"
        next
        edit "underlay"
```

```
        next
    end
    config members
        edit 1
            set interface "port1"
            set zone "underlay"
        next
        edit 2
            set interface "port2"
            set zone "underlay"
            set gateway 172.31.3.6
        next
        edit 3
            set interface "H1_T11"
            set zone "overlay"
            set source 172.31.0.65
        next
        edit 4
            set interface "H1_T12"
            set zone "overlay"
            set source 172.31.0.65
        next
        edit 5
            set interface "H1_T21"
            set zone "overlay"
            set source 172.31.0.65
        next
        edit 6
            set interface "H1_T22"
            set zone "overlay"
            set source 172.31.0.65
        next
    end
    config health-check
        edit "HUB"
            set server "8.8.4.4"
            set update-static-route disable
            set members 3 4 5 6
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
    config service
        edit 1
            set name "1"
            set load-balance enable
            set mode sla
```

```
            set hash-mode bibandwidth
            set dst "all"
            set src "CORP_LAN"
            config sla
                edit "HUB"
                    set id 1
                next
            end
            set priority-members 3 4 5 6
        next
    end
end
```

3. By default, the service references the overlay available bandwidth:

```
Branch1_A_FGT (root) (Interim)# diagnose sys sdwan  service4

Service(1): Address Mode(IPV4) flags=0x24200 use-shortcut-sla use-shortcut load-balance
 Tie break: cfg
 Shortcut priority: 2
  Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla hash-
mode=bibandwidth)
  Members(4):
    1: Seq_num(3 H1_T11 overlay), alive, sla(0x1), gid(2), num of pass(1), bibandwidth:
179995Kbps, selected                          // 90000 + 90000 = 180000
    2: Seq_num(4 H1_T12 overlay), alive, sla(0x1), gid(2), num of pass(1), bibandwidth:
179992Kbps, selected                          // 90000 + 90000 = 180000
    3: Seq_num(5 H1_T21 overlay), alive, sla(0x1), gid(2), num of pass(1), bibandwidth:
79997Kbps, selected                           // 40000 + 40000 = 80000
    4: Seq_num(6 H1_T22 overlay), alive, sla(0x1), gid(2), num of pass(1), bibandwidth:
79997Kbps, selected                           // 40000 + 40000 = 80000
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

4. Configure underlay as `bandwidth-type` to have the service refer to underlay available bandwidth:

```
config sys sdwan
    config service
        edit 1
            set bandwidth-type underlay
        next
    end
end
```

5. Check the service.
   H1_T11/H1_12 has more available bibandwidth. H1_T11 is configured before H1_12, so H1_T11 is preferred.

```
Branch1_A_FGT (root) (Interim)# diagnose sys sdwan  service4

Service(1): Address Mode(IPV4) flags=0xa4200 use-shortcut-sla use-shortcut load-balance
```

```
 Tie break: cfg
 Shortcut priority: 2
  Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla hash-
mode=bibandwidth)
  Members(4):
    1: Seq_num(3 H1_T11 overlay), alive, sla(0x1), gid(2), num of pass(1), bibandwidth:
199979Kbps, selected         // 100000 + 100000 = 200000
    2: Seq_num(4 H1_T12 overlay), alive, sla(0x1), gid(2), num of pass(1), bibandwidth:
199979Kbps, selected         // 100000 + 100000 = 200000
    3: Seq_num(5 H1_T21 overlay), alive, sla(0x1), gid(2), num of pass(1), bibandwidth:
99990Kbps, selected           // 50000 + 50000 = 100000
    4: Seq_num(6 H1_T22 overlay), alive, sla(0x1), gid(2), num of pass(1), bibandwidth:
99990Kbps, selected           // 50000 + 50000 = 100000
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

6. Send around 83M traffic from the client to server, and check the SD-WAN status.
   83M traffic goes through H1_T11:

```
Branch1_A_FGT (root) (Interim)# diagnose sys sdwan intf-sla-log H1_T11

Timestamp: Wed Sep 24 11:34:44 2025, ifn: H1_T11, used inbandwidth: 10548bps, used
outbandwidth: 82640376bps, used bibandwidth: 82650924bps, tx bytes: 28265816236bytes, rx
bytes: 10078621bytes.
Timestamp: Wed Sep 24 11:34:54 2025, ifn: H1_T11, used inbandwidth: 7029bps, used
outbandwidth: 82321898bps, used bibandwidth: 82328927bps, tx bytes: 28367910559bytes, rx
bytes: 10078621bytes.


Branch1_A_FGT (root) (Interim)# diagnose sys sdwan intf-sla-log H1_T12

Timestamp: Wed Sep 24 11:36:35 2025, ifn: H1_T12, used inbandwidth: 2333bps, used
outbandwidth: 5032bps, used bibandwidth: 7365bps, tx bytes: 2862896972bytes, rx bytes:
6317363bytes.
Timestamp: Wed Sep 24 11:36:45 2025, ifn: H1_T12, used inbandwidth: 3464bps, used
outbandwidth: 5331bps, used bibandwidth: 8795bps, tx bytes: 2862903397bytes, rx bytes:
6324431bytes.

Branch1_A_FGT (root) (Interim)# diagnose sys sdwan intf-sla-log H1_T21

Timestamp: Wed Sep 24 11:39:05 2025, ifn: H1_T21, used inbandwidth: 2392bps, used
outbandwidth: 1468bps, used bibandwidth: 3860bps, tx bytes: 35828418453bytes, rx bytes:
4425608bytes.
Timestamp: Wed Sep 24 11:39:15 2025, ifn: H1_T21, used inbandwidth: 1580bps, used
outbandwidth: 1340bps, used bibandwidth: 2920bps, tx bytes: 35828419813bytes, rx bytes:
4425608bytes.

Branch1_A_FGT (root) (Interim)# diagnose sys sdwan intf-sla-log H1_T22

Timestamp: Wed Sep 24 11:39:35 2025, ifn: H1_T22, used inbandwidth: 1580bps, used
```

```
outbandwidth: 778bps, used bibandwidth: 2358bps, tx bytes: 54580737167bytes, rx bytes:
7183126bytes.
Timestamp: Wed Sep 24 11:39:45 2025, ifn: H1_T22, used inbandwidth: 2371bps, used
outbandwidth: 869bps, used bibandwidth: 3240bps, tx bytes: 54580738499bytes, rx bytes:
7188066bytes.
```

83M traffic goes through H1_T11's underlay:

```
Branch1_A_FGT (root) (Interim)# diagnose sys sdwan intf-sla-log port1

Timestamp: Wed Sep 24 11:40:35 2025, ifn: port1, used inbandwidth: 12103bps, used
outbandwidth: 86314630bps, used bibandwidth: 86326733bps, tx bytes: 36450897420bytes, rx
bytes: 18861999bytes.
Timestamp: Wed Sep 24 11:40:45 2025, ifn: port1, used inbandwidth: 12126bps, used
outbandwidth: 86321860bps, used bibandwidth: 86333986bps, tx bytes: 36558788532bytes, rx
bytes: 18877396bytes.

Branch1_A_FGT (root) (Interim)# diagnose sys sdwan intf-sla-log port2

Timestamp: Wed Sep 24 11:41:46 2025, ifn: port2, used inbandwidth: 4652bps, used outbandwidth:
5278bps, used bibandwidth: 9930bps, tx bytes: 94847045244bytes, rx bytes: 18330597bytes.
Timestamp: Wed Sep 24 11:41:56 2025, ifn: port2, used inbandwidth: 4728bps, used outbandwidth:
5375bps, used bibandwidth: 10103bps, tx bytes: 94847052353bytes, rx bytes: 18336819bytes.
```

For H1_T11, the underlay available bandwidth (roughly 200000 - 86333.986) is referenced. Although traffic is not on H1_T12, its underlay available bandwidth (roughly 200000 - 86333.986) is referenced as well.

```
Branch1_A_FGT (root) (Interim)# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0xa4200 use-shortcut-sla use-shortcut load-balance
 Tie break: cfg
 Shortcut priority: 2
  Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla hash-
mode=bibandwidth)
  Members(4):
    1: Seq_num(3 H1_T11 overlay), alive, sla(0x1), gid(2), num of pass(1), bibandwidth:
113673Kbps, selected
    2: Seq_num(4 H1_T12 overlay), alive, sla(0x1), gid(2), num of pass(1), bibandwidth:
113673Kbps, selected
    3: Seq_num(5 H1_T21 overlay), alive, sla(0x1), gid(2), num of pass(1), bibandwidth:
99990Kbps, selected
    4: Seq_num(6 H1_T22 overlay), alive, sla(0x1), gid(2), num of pass(1), bibandwidth:
99990Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255.
```

7. Make H1_T11 out-of-sla.
   H1_T12 will be preferred as expected because it has more available bibandwidth within all in-sla members:

```
Branch1_A_FGT (root) (Interim)# diagnose sys sdwan  service4

Service(1): Address Mode(IPV4) flags=0xa4200 use-shortcut-sla use-shortcut load-balance
 Tie break: cfg
 Shortcut priority: 2
  Gen(2), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla hash-
mode=bibandwidth)
  Members(4):
    1: Seq_num(4 H1_T12 overlay), alive, sla(0x1), gid(2), num of pass(1), bibandwidth:
113673Kbps, selected
    2: Seq_num(5 H1_T21 overlay), alive, sla(0x1), gid(2), num of pass(1), bibandwidth:
99990Kbps, selected
    3: Seq_num(6 H1_T22 overlay), alive, sla(0x1), gid(2), num of pass(1), bibandwidth:
99990Kbps, selected
    4: Seq_num(3 H1_T11 overlay), alive, sla(0x0), gid(1), num of pass(0), bibandwidth:
113673Kbps, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

ToS-based SD-WAN duplication matching

Duplication rules now support matching traffic based on ToS (Type of Service), enabling more precise control. Previously, ToS criterion was not available in the duplication configuration.

The `config duplication` command includes new options:

```
config system sdwan
    config duplication
      edit 1
        ...
        set tos <8bit hex value>
        set tos-mask <8bit hex value>
      next
    end
end
```

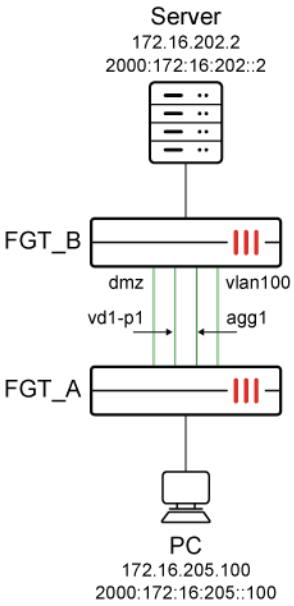
Option	Description
`tos <8bit hex value>`	Type of service bit pattern. The 8bit hex value is the bit pattern to match after applying `tos-mask`.
`tos-mask <8bit hex value>`	Type of service evaluated bits. The 8bit hex value is the bit mask of bytes to compare to ToS.

Example

This example demonstrates how to configure duplication and de-duplication rules that match traffic based on ToS 0x77.

FGT_A is configured with a zone (zone1) that allows a maximum of four zone members (dmz, vd1-p1, agg1, and vlan100) to perform duplication on packets with ToS 0x77.

FGT_B is configured with a zone (virtual-wan-link) that allows four zone members (dmz, vd1-p1, agg1, and vlan100) to perform de-duplication on packets with ToS 0x77.



Configuring duplication on FGT_A

To configure duplication on FGT_A:

1. On FGT_A, enable SD-WAN, set a duplication maximum, create a zone with four members, and create a duplication rule for ToS 0x77:

   A duplication rule is defined to only match packets with ToS 0x77.

```
config system sdwan
    set status enable
    set duplication-max-num 4
    config zone
        edit "virtual-wan-link"
        next
        edit "zone1"
        next
    end
    config members
        edit 1
            set interface "dmz"
            set zone "zone1"
```

```
            set gateway 172.16.208.2
            set gateway6 2000:172:16:208::2
        next
        edit 2
            set interface "vd1-p1"
            set zone "zone1"
        next
        edit 3
            set interface "agg1"
            set zone "zone1"
            set gateway 172.16.203.2
            set gateway6 2000:172:16:203::2
        next
        edit 4
            set interface "vlan100"
            set zone "zone1"
            set gateway 172.16.206.2
            set gateway6 2000:172:16:206::2
        next
    end
    config duplication
        edit 1
            set srcaddr "172.16.205.0"
            set dstaddr "172.16.202.0"
            set srcaddr6 "2000:172:16:205::"
            set dstaddr6 "2000:172:16:202::"
            set srcintf "port5"
            set dstintf "zzzzzzzzzzz"
            set service "PING" "PING6"
            set tos 0x77
            set tos-mask 0xff
            set packet-duplication force
        next
    end
end
```

2. Configure IPv4 and IPv6 routes on members of zone1:

```
config router static
    ...
    edit 2
        set distance 1
        set sdwan-zone "virtual-wan-link" "zone1"
    next
end

config router static6
    ...
    edit 2
        set distance 1
        set sdwan-zone "virtual-wan-link" "zone1"
```

```
    next
end
```

3. Check IPv4 and IPv6 routing on which duplication relies:

```
get router info routing-table static
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 172.16.203.2, agg1, [1/0]
                        [1/0] via 172.16.206.2, vlan100, [1/0]
                        [1/0] via 172.16.208.2, dmz, [1/0]
                        [1/0] via vd1-p1 tunnel 172.16.209.2, [1/0]

FGT_A (root) (Interim)# get router info6 routing-table static
Routing table for VRF=0
S*      ::/0 [1/0] via vd1-p1 tunnel ::172.16.209.2, 23:20:44, [1024/0]
                [1/0] via 2000:172:16:203::2, agg1, 23:20:44, [1024/0]
                [1/0] via 2000:172:16:206::2, vlan100, 23:20:44, [1024/0]
                [1/0] via 2000:172:16:208::2, dmz, 23:20:44, [1024/0]
```

4. Perform IPv4 and IPv6 ping with ToS 0x00 from PC to server to confirm duplication doesn't occur because duplication rule only matches ToS 0x77:

```
~$ ping 172.16.202.2 -Q 0x00
PING 172.16.202.2 (172.16.202.2) 56(84) bytes of data.
64 bytes from 172.16.202.2: icmp_seq=1 ttl=253 time=0.480 ms
64 bytes from 172.16.202.2: icmp_seq=2 ttl=253 time=0.307 ms
64 bytes from 172.16.202.2: icmp_seq=3 ttl=253 time=0.158 ms
64 bytes from 172.16.202.2: icmp_seq=4 ttl=253 time=0.168 ms
^C
--- 172.16.202.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3059ms
rtt min/avg/max/mdev = 0.158/0.278/0.480/0.131 ms


~$ ping 2000:172:16:202::2 -Q 0x00
PING 2000:172:16:202::2(2000:172:16:202::2) 56 data bytes
64 bytes from 2000:172:16:202::2: icmp_seq=1 ttl=62 time=0.699 ms
64 bytes from 2000:172:16:202::2: icmp_seq=2 ttl=62 time=0.416 ms
64 bytes from 2000:172:16:202::2: icmp_seq=3 ttl=62 time=0.178 ms
64 bytes from 2000:172:16:202::2: icmp_seq=4 ttl=62 time=0.179 ms
^C
--- 2000:172:16:202::2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3051ms
rtt min/avg/max/mdev = 0.178/0.368/0.699/0.214 ms
```

5. Perform IPv4 and IPv6 ping with ToS 0x77 from PC to server to confirm duplication occurs:

```
~$ ping 172.16.202.2 -Q 0x77
PING 172.16.202.2 (172.16.202.2) 56(84) bytes of data.
64 bytes from 172.16.202.2: icmp_seq=1 ttl=253 time=0.609 ms
64 bytes from 172.16.202.2: icmp_seq=1 ttl=253 time=0.609 ms (DUP!)
64 bytes from 172.16.202.2: icmp_seq=1 ttl=253 time=0.609 ms (DUP!)
```

```
64 bytes from 172.16.202.2: icmp_seq=1 ttl=253 time=0.609 ms (DUP!)
64 bytes from 172.16.202.2: icmp_seq=2 ttl=253 time=0.512 ms
64 bytes from 172.16.202.2: icmp_seq=2 ttl=253 time=0.512 ms (DUP!)
64 bytes from 172.16.202.2: icmp_seq=2 ttl=253 time=0.512 ms (DUP!)
64 bytes from 172.16.202.2: icmp_seq=2 ttl=253 time=0.512 ms (DUP!)
^C
--- 172.16.202.2 ping statistics ---
2 packets transmitted, 2 received, +6 duplicates, 0% packet loss, time 1005ms
rtt min/avg/max/mdev = 0.512/0.560/0.609/0.053 ms


~$ ping 2000:172:16:202::2 -Q 0x77
PING 2000:172:16:202::2(2000:172:16:202::2) 56 data bytes
64 bytes from 2000:172:16:202::2: icmp_seq=1 ttl=62 time=0.551 ms
64 bytes from 2000:172:16:202::2: icmp_seq=1 ttl=62 time=0.551 ms (DUP!)
64 bytes from 2000:172:16:202::2: icmp_seq=1 ttl=62 time=0.551 ms (DUP!)
64 bytes from 2000:172:16:202::2: icmp_seq=1 ttl=62 time=0.551 ms (DUP!)
64 bytes from 2000:172:16:202::2: icmp_seq=2 ttl=62 time=0.429 ms
64 bytes from 2000:172:16:202::2: icmp_seq=2 ttl=62 time=0.429 ms (DUP!)
64 bytes from 2000:172:16:202::2: icmp_seq=2 ttl=62 time=0.429 ms (DUP!)
64 bytes from 2000:172:16:202::2: icmp_seq=2 ttl=62 time=0.429 ms (DUP!)
^C
--- 2000:172:16:202::2 ping statistics ---
2 packets transmitted, 2 received, +6 duplicates, 0% packet loss, time 1010ms
rtt min/avg/max/mdev = 0.429/0.490/0.551/0.061 ms
```

Configuring de-duplication on FGT_B

To configure de-duplication on FGT_B:

1. On FGT_B, enable SD-WAN, define zone members, and define a de-duplication rule:
   A de-duplication rule is defined to only match packets with ToS 0x77.

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "dmz"
            set gateway 172.16.208.1
            set gateway6 2000:172:16:208::1
        next
        edit 2
            set interface "vd1-p1"
        next
        edit 3
            set interface "agg1"
            set gateway 172.16.203.1
```

```
            set gateway6 2000:172:16:203::1
        next
        edit 4
            set interface "vlan100"
            set gateway 172.16.206.1
            set gateway6 2000:172:16:206::1
        next
    end
    config duplication
        edit 1
            set srcaddr "172.16.205.0"
            set dstaddr "172.16.202.0"
            set srcaddr6 "2000:172:16:205::"
            set dstaddr6 "2000:172:16:202::"
            set srcintf "virtual-wan-link"
            set dstintf "port5"
            set service "PING" "PING6"
            set tos 0x77
            set tos-mask 0xff
            set packet-de-duplication enable
        next
    end
end
```

2. Perform an IPv4 and IPv6 ping with ToS 0x77 from PC to server to show that de-duplication occurs on FGT_B:

```
~$ ping 172.16.202.2 -Q 0x77
PING 172.16.202.2 (172.16.202.2) 56(84) bytes of data.
64 bytes from 172.16.202.2: icmp_seq=1 ttl=253 time=0.587 ms
64 bytes from 172.16.202.2: icmp_seq=2 ttl=253 time=0.451 ms
64 bytes from 172.16.202.2: icmp_seq=3 ttl=253 time=0.458 ms
64 bytes from 172.16.202.2: icmp_seq=4 ttl=253 time=0.428 ms
64 bytes from 172.16.202.2: icmp_seq=5 ttl=253 time=0.437 ms
^C
--- 172.16.202.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4074ms
rtt min/avg/max/mdev = 0.428/0.472/0.587/0.059 ms

~$ ping 2000:172:16:202::2 -Q 0x77
PING 2000:172:16:202::2(2000:172:16:202::2) 56 data bytes
64 bytes from 2000:172:16:202::2: icmp_seq=1 ttl=62 time=0.758 ms
64 bytes from 2000:172:16:202::2: icmp_seq=2 ttl=62 time=0.420 ms
64 bytes from 2000:172:16:202::2: icmp_seq=3 ttl=62 time=0.421 ms
64 bytes from 2000:172:16:202::2: icmp_seq=4 ttl=62 time=0.400 ms
64 bytes from 2000:172:16:202::2: icmp_seq=5 ttl=62 time=0.426 ms
^C
--- 2000:172:16:202::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4090ms
rtt min/avg/max/mdev = 0.400/0.485/0.758/0.136 ms
```

Stop packet duplication upon reaching bandwidth utilization threshold

Utilized bandwidth on overlays or their underlays can be checked before duplication. When utilized bandwidth is less than the specified threshold, duplication starts. When it exceeds the specified threshold, duplication stops.

New options are available to specify the bandwidth measurement to use for evaluation:

```
config system sdwan
    config members
        edit <id>
            set duplication-threshold-upbandwidth <integer>
            set duplication-threshold-dwbandwidth <integer>
            set duplication-threshold-bibandwidth <integer>
        next
    end
end
```

Option	Description
`duplication-threshold-upbandwidth <integer>`	Bandwidth upstream threshold value in kilobytes per second (0 - 4294967295, default = 0).
`duplication-threshold-dwbandwidth <integer>`	Bandwidth downstream threshold value in kilobytes per second (0 - 4294967295, default = 0).
`duplication-threshold-bibandwidth <integer>`	Bandwidth bistream threshold value in kilobytes per second (0 - 4294967295, default = 0).

A new option is available to specify underlay or overlay:

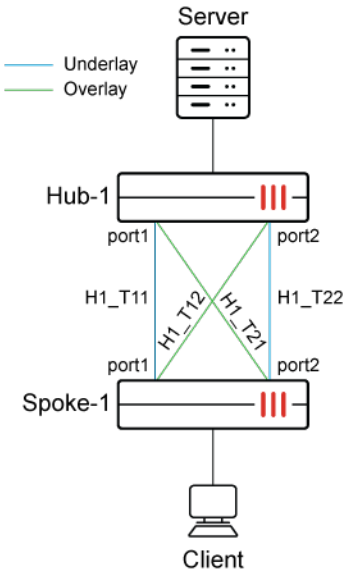
```
config system sdwan
    config members
        edit <id>
            set duplication-threshold-bandwidth {overlay* | underlay}
        next
    end
end
```

Option	Description
`duplication-threshold-bandwidth {overlay* | underlay}`	Configure duplication threshold bandwidth interface in the SD-WAN. Choose from: • `overlay`: Use overlay interface bandwidth for duplication in the SD-WAN. • `underlay`: Use underlay interface bandwidth for duplication in the SD-WAN.

Example

This example demonstrates how to configure and test a threshold for bandwidth utilization. In summary:

- Spoke-1 has four overlays: H1_T11, H1_T12, H1_T21, and H1_T22.
- H1_T11 and H1_T12 are associated with the same underlay port1, and H1_T21 and H1_T22 are associated with the same underlay port2.
- Traffic is intended to duplicate on all four overlays as long as the used bandwidth on the underlays doesn't exceed the specified threshold.
- Once the used bandwidth exceeds the specified threshold, duplication on the associated overlays stops.



To configure Spoke-1:

1. Configure Spoke-1.
   In this example, the duplication maximum is set to 4. For 4 members, the upstream bandwidth is set to 50 M, and the used bandwidth of its underlay is referenced. Duplication is also configured.

```
config system sdwan
    set status enable
    set duplication-max-num 4
    config zone
        edit "virtual-wan-link"
        next
        edit "overlay"
        next
        edit "underlay"
        next
    end
    config members
        edit 1
            set interface "port1"
            set zone "underlay"
        next
        edit 2
            set interface "port2"
            set zone "underlay"
```

```
            set gateway 172.31.3.6
        next
        edit 3
            set duplication-threshold-upbandwidth 50000
            set duplication-threshold-bandwidth underlay
            set interface "H1_T11"
            set zone "overlay"
            set source 172.31.0.65
        next
        edit 4
            set duplication-threshold-upbandwidth 50000
            set duplication-threshold-bandwidth underlay
            set interface "H1_T12"
            set zone "overlay"
            set source 172.31.0.65
        next
        edit 5
            set duplication-threshold-upbandwidth 50000
            set duplication-threshold-bandwidth underlay
            set interface "H1_T21"
            set zone "overlay"
            set source 172.31.0.65
        next
        edit 6
            set duplication-threshold-upbandwidth 50000
            set duplication-threshold-bandwidth underlay
            set interface "H1_T22"
            set zone "overlay"
            set source 172.31.0.65
        next
    end
     config health-check
        edit "HUB"
            set server "172.31.100.100"
            set update-static-route disable
            set members 3 4 5 6
            config sla
                edit 1
                next
            end
        next
    end
    config duplication
        edit 1
            set srcaddr "CORP_LAN"
            set dstaddr "CORP_LAN"
            set srcintf "lan_zone"
            set dstintf "overlay"
            set service "PING"
            set packet-duplication force
        next
```

```
    end
end
```

2. Send constant pings from the client to server to duplicate traffic on all overlays.
   The following example shows duplication on four overlays:

```
test@PC3:~$ ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2: icmp_seq=1 ttl=62 time=0.593 ms
64 bytes from 10.0.1.2: icmp_seq=1 ttl=62 time=0.628 ms (DUP!)
64 bytes from 10.0.1.2: icmp_seq=1 ttl=62 time=0.636 ms (DUP!)
64 bytes from 10.0.1.2: icmp_seq=1 ttl=62 time=0.643 ms (DUP!)
64 bytes from 10.0.1.2: icmp_seq=2 ttl=62 time=0.579 ms
64 bytes from 10.0.1.2: icmp_seq=2 ttl=62 time=0.592 ms (DUP!)
64 bytes from 10.0.1.2: icmp_seq=2 ttl=62 time=0.594 ms (DUP!)
64 bytes from 10.0.1.2: icmp_seq=2 ttl=62 time=0.595 ms (DUP!)
^C
--- 10.0.1.2 ping statistics ---
2 packets transmitted, 2 received, +6 duplicates, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.579/0.607/0.643/0.033 ms

# diagnose sniffer packet any 'host 10.0.1.2' 4
interfaces=[any]
filters=[host 10.0.1.2]
0.700706 port4 in 10.0.3.2 -> 10.0.1.2: icmp: echo request
0.700740 H1_T12 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
0.700761 H1_T21 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
0.700782 H1_T22 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
0.700796 H1_T11 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
0.701139 H1_T11 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
0.701164 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
0.701170 H1_T11 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
0.701182 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
0.701188 H1_T11 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
0.701203 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
0.701208 H1_T11 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
0.701220 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
1.700691 port4 in 10.0.3.2 -> 10.0.1.2: icmp: echo request
1.700730 H1_T12 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
1.700759 H1_T21 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
1.700777 H1_T22 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
1.700792 H1_T11 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
1.701115 H1_T11 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
1.701144 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
1.701150 H1_T11 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
1.701163 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
1.701168 H1_T11 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
1.701180 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
1.701185 H1_T11 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
1.701196 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
```

3. Generate around 80 M background traffic on underlay port2.

Duplication on H1_T21 and H1_T22, which are associated with port2, stops because 80 M background traffic on underlay port2 exceeds the threshold 50 M. Duplication occurs on H1_T11 and H1_T12:

```
# diagnose sys sdwan intf-sla-log port2

Timestamp: Tue Sep 30 09:42:18 2025, ifn: port2, used inbandwidth: 5363bps, used outbandwidth:
84142940bps, used bibandwidth: 84148303bps, tx bytes: 95718653738bytes, rx bytes:
304404953bytes.
Timestamp: Tue Sep 30 09:42:28 2025, ifn: port2, used inbandwidth: 5152bps, used outbandwidth:
84143025bps, used bibandwidth: 84148177bps, tx bytes: 95823831406bytes, rx bytes:
304410822bytes.

test@PC3:~$ ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2: icmp_seq=1 ttl=62 time=0.734 ms
64 bytes from 10.0.1.2: icmp_seq=1 ttl=62 time=0.744 ms (DUP!)
64 bytes from 10.0.1.2: icmp_seq=2 ttl=62 time=0.508 ms
64 bytes from 10.0.1.2: icmp_seq=2 ttl=62 time=0.519 ms (DUP!)
^C
--- 10.0.1.2 ping statistics ---
2 packets transmitted, 2 received, +2 duplicates, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.508/0.626/0.744/0.114 ms
test@PC3:~$

# diagnose sniffer packet any 'host 10.0.1.2 and icmp' 4
interfaces=[any]
filters=[host 10.0.1.2 and icmp]
2.631011 port4 in 10.0.3.2 -> 10.0.1.2: icmp: echo request
2.631117 H1_T12 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
2.631139 H1_T11 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
2.631587 H1_T11 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
2.631618 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
2.631624 H1_T11 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
2.631637 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
3.629973 port4 in 10.0.3.2 -> 10.0.1.2: icmp: echo request
3.630001 H1_T12 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
3.630023 H1_T11 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
3.630361 H1_T11 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
3.630387 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
3.630393 H1_T11 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
3.630406 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
```

4. Stop 80 M background traffic on port2.
   Duplication on four overlays occurs again:

```
# diagnose sys sdwan intf-sla-log port2

Timestamp: Tue Sep 30 09:54:30 2025, ifn: port2, used inbandwidth: 7239bps, used outbandwidth:
7964bps, used bibandwidth: 15203bps, tx bytes: 101144616164bytes, rx bytes: 304801192bytes.
Timestamp: Tue Sep 30 09:54:40 2025, ifn: port2, used inbandwidth: 7898bps, used outbandwidth:
7957bps, used bibandwidth: 15855bps, tx bytes: 101144625934bytes, rx bytes: 304812991bytes.

test@PC3:~$ ping 10.0.1.2
```

```
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2: icmp_seq=1 ttl=62 time=0.663 ms
64 bytes from 10.0.1.2: icmp_seq=1 ttl=62 time=0.712 ms (DUP!)
64 bytes from 10.0.1.2: icmp_seq=1 ttl=62 time=0.719 ms (DUP!)
64 bytes from 10.0.1.2: icmp_seq=1 ttl=62 time=0.722 ms (DUP!)
64 bytes from 10.0.1.2: icmp_seq=2 ttl=62 time=0.561 ms
64 bytes from 10.0.1.2: icmp_seq=2 ttl=62 time=0.611 ms (DUP!)
64 bytes from 10.0.1.2: icmp_seq=2 ttl=62 time=0.613 ms (DUP!)
64 bytes from 10.0.1.2: icmp_seq=2 ttl=62 time=0.614 ms (DUP!)
^C
--- 10.0.1.2 ping statistics ---
2 packets transmitted, 2 received, +6 duplicates, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.561/0.651/0.722/0.066 ms

# diagnose sniffer packet any 'host 10.0.1.2 and icmp' 4
interfaces=[any]
filters=[host 10.0.1.2 and icmp]
3.019665 port4 in 10.0.3.2 -> 10.0.1.2: icmp: echo request
3.019778 H1_T12 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
3.019800 H1_T21 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
3.019819 H1_T22 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
3.019833 H1_T11 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
3.020200 H1_T22 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
3.020240 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
3.020247 H1_T22 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
3.020260 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
3.020266 H1_T22 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
3.020277 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
3.020282 H1_T22 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
3.020293 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
4.018668 port4 in 10.0.3.2 -> 10.0.1.2: icmp: echo request
4.018722 H1_T12 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
4.018744 H1_T21 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
4.018759 H1_T22 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
4.018772 H1_T11 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
4.019100 H1_T22 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
4.019133 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
4.019139 H1_T22 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
4.019152 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
4.019157 H1_T22 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
4.019170 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
4.019174 H1_T22 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
4.019186 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
^C
26 packets received by filter
0 packets dropped by kernel
```

Selective SD-WAN duplication

SD-WAN duplication now supports selective member targeting, offering greater flexibility and control. Previously, duplication applied to all members in one zone indiscriminately.

The `config duplication` command includes new options:

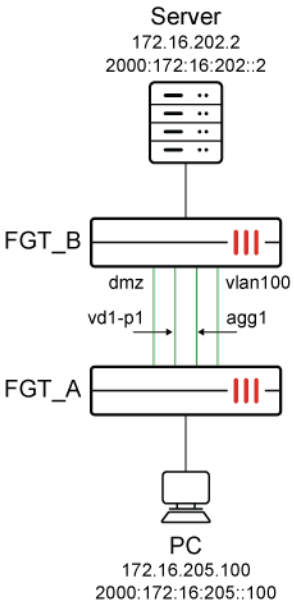
```
config system sdwan
    config duplication
      edit 1
        ...
        set members <*seq-num>
      next
    end
end
```

Option	Description
`members <*seq-num>`	Member sequence number list.

Example

This example demonstrates how to configure FGT_A of the SD-WAN topology to perform duplication on only two of four zone members.

FGT_A is configured with a zone (zone1) that allows a maximum of four zone members (dmz, vd1-p1, agg1, and vlan100) to perform duplication, and then duplication is further specified to occur only on members 1 and 2.



To configure duplication on FGT_A:

1. On FGT_A, enable SD-WAN, set a duplication maximum, create a zone with four members, set member priority, and define the duplication rule:

   In this example, member 1 is preferred for IPv4 and IPv6 traffic.

```
config system sdwan
    set status enable
    set duplication-max-num 4
    config zone
        edit "virtual-wan-link"
        next
        edit "zone1"
        next
    end
    config members
        edit 1
            set interface "dmz"
            set zone "zone1"
            set gateway 172.16.208.2
            set gateway6 2000:172:16:208::2
        next
        edit 2
            set interface "vd1-p1"
            set zone "zone1"
        next
        edit 3
            set interface "agg1"
            set zone "zone1"
            set gateway 172.16.203.2
            set gateway6 2000:172:16:203::2
        next
        edit 4
            set interface "vlan100"
            set zone "zone1"
            set gateway 172.16.206.2
            set gateway6 2000:172:16:206::2
        next
    end
    config service
        edit 1
            set name "1"
            set dst "all"
            set src "172.16.205.0"
            set priority-members 1 2 3 4
        next
        edit 2
            set name "2"
            set addr-mode ipv6
            set priority-members 1 2 3 4
            set dst6 "all"
            set src6 "2000:172:16:205::"
```

```
        next
    end
    config duplication
        edit 1
            set srcaddr "172.16.205.0"
            set dstaddr "172.16.202.0"
            set srcaddr6 "2000:172:16:205::"
            set dstaddr6 "2000:172:16:202::"
            set srcintf "port5"
            set dstintf "zone1"
            set service "PING" "PING6"
            set packet-duplication force
        next
    end
end
```

2. Configure IPv4 and IPv6 routes on members of zone1:

```
config router static
    ...
    edit 2
        set distance 1
        set sdwan-zone "virtual-wan-link" "zone1"
    next
end

config router static6
    ...
    edit 2
        set distance 1
        set sdwan-zone "virtual-wan-link" "zone1"
    next
end
```

3. Check IPv4 and IPv6 routing on which duplication relies:

```
get router info routing-table static
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 172.16.203.2, agg1, [1/0]
                        [1/0] via 172.16.206.2, vlan100, [1/0]
                        [1/0] via 172.16.208.2, dmz, [1/0]
                        [1/0] via vd1-p1 tunnel 172.16.209.2, [1/0]

FGT_A (root) (Interim)# get router info6 routing-table static
Routing table for VRF=0
S*      ::/0 [1/0] via vd1-p1 tunnel ::172.16.209.2, 23:20:44, [1024/0]
                [1/0] via 2000:172:16:203::2, agg1, 23:20:44, [1024/0]
                [1/0] via 2000:172:16:206::2, vlan100, 23:20:44, [1024/0]
                [1/0] via 2000:172:16:208::2, dmz, 23:20:44, [1024/0]
```

4. Perform IPv4 and IPv6 ping from PC to server to see that packets are copied on all members in zone1:

```
~$ ping 172.16.202.2
PING 172.16.202.2 (172.16.202.2) 56(84) bytes of data.
64 bytes from 172.16.202.2: icmp_seq=1 ttl=253 time=0.577 ms
64 bytes from 172.16.202.2: icmp_seq=1 ttl=253 time=0.577 ms (DUP!)
64 bytes from 172.16.202.2: icmp_seq=1 ttl=253 time=0.577 ms (DUP!)
64 bytes from 172.16.202.2: icmp_seq=1 ttl=253 time=0.577 ms (DUP!)
64 bytes from 172.16.202.2: icmp_seq=2 ttl=253 time=0.467 ms
64 bytes from 172.16.202.2: icmp_seq=2 ttl=253 time=0.467 ms (DUP!)
64 bytes from 172.16.202.2: icmp_seq=2 ttl=253 time=0.467 ms (DUP!)
64 bytes from 172.16.202.2: icmp_seq=2 ttl=253 time=0.467 ms (DUP!)
^C
--- 172.16.202.2 ping statistics ---
2 packets transmitted, 2 received, +6 duplicates, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.467/0.522/0.577/0.055 ms

FGT_A (root) (Interim)# diagnose sniffer packet any 'host 172.16.202.2' 4
interfaces=[any]
filters=[host 172.16.202.2]
2.515725 port5 in 172.16.205.100 -> 172.16.202.2: icmp: echo request
2.515864 vd1-p1 out 172.16.205.100 -> 172.16.202.2: icmp: echo request           // member 2
2.515903 agg1 out 172.16.205.100 -> 172.16.202.2: icmp: echo request              // member 3
2.515909 port4 out 172.16.205.100 -> 172.16.202.2: icmp: echo request
2.515918 vlan100 out 172.16.205.100 -> 172.16.202.2: icmp: echo request         // member 4
2.515922 agg1 out 172.16.205.100 -> 172.16.202.2: icmp: echo request
2.515924 port4 out 172.16.205.100 -> 172.16.202.2: icmp: echo request
2.515930 dmz out 172.16.205.100 -> 172.16.202.2: icmp: echo request               // member 1
2.516292 vlan100 in 172.16.202.2 -> 172.16.205.100: icmp: echo reply
2.516347 vlan100 in 172.16.202.2 -> 172.16.205.100: icmp: echo reply
2.516362 vlan100 in 172.16.202.2 -> 172.16.205.100: icmp: echo reply
2.516375 vlan100 in 172.16.202.2 -> 172.16.205.100: icmp: echo reply
2.516431 port5 out 172.16.202.2 -> 172.16.205.100: icmp: echo reply
2.516437 port5 out 172.16.202.2 -> 172.16.205.100: icmp: echo reply
2.516441 port5 out 172.16.202.2 -> 172.16.205.100: icmp: echo reply
2.516444 port5 out 172.16.202.2 -> 172.16.205.100: icmp: echo reply


~$ ping 2000:172:16:202::2
PING 2000:172:16:202::2(2000:172:16:202::2) 56 data bytes
64 bytes from 2000:172:16:202::2: icmp_seq=1 ttl=62 time=0.766 ms
64 bytes from 2000:172:16:202::2: icmp_seq=1 ttl=62 time=0.766 ms (DUP!)
64 bytes from 2000:172:16:202::2: icmp_seq=1 ttl=62 time=0.766 ms (DUP!)
64 bytes from 2000:172:16:202::2: icmp_seq=1 ttl=62 time=0.766 ms (DUP!)
64 bytes from 2000:172:16:202::2: icmp_seq=2 ttl=62 time=0.539 ms
64 bytes from 2000:172:16:202::2: icmp_seq=2 ttl=62 time=0.539 ms (DUP!)
64 bytes from 2000:172:16:202::2: icmp_seq=2 ttl=62 time=0.539 ms (DUP!)
64 bytes from 2000:172:16:202::2: icmp_seq=2 ttl=62 time=0.539 ms (DUP!)
^C
--- 2000:172:16:202::2 ping statistics ---
2 packets transmitted, 2 received, +6 duplicates, 0% packet loss, time 1015ms
rtt min/avg/max/mdev = 0.539/0.652/0.766/0.116 ms

FGT_A (root) (Interim)#  diagnose sniffer packet any 'host 2000:172:16:202::2' 4
```

```
interfaces=[any]
filters=[host 2000:172:16:202::2]
3.400223 port5 in 2000:172:16:205::100 -> 2000:172:16:202::2: icmp6: echo request seq 1
[flowlabel 0xf62c8]
3.400370 vd1-p1 out 2000:172:16:205::100 -> 2000:172:16:202::2: icmp6: echo request seq 1
[flowlabel 0xf62c8]                    // member 2
3.400419 agg1 out 2000:172:16:205::100 -> 2000:172:16:202::2: icmp6: echo request seq 1
[flowlabel 0xf62c8]                       // member 3
3.400424 port4 out 2000:172:16:205::100 -> 2000:172:16:202::2: icmp6: echo request seq 1
[flowlabel 0xf62c8]
3.400435 vlan100 out 2000:172:16:205::100 -> 2000:172:16:202::2: icmp6: echo request seq 1
[flowlabel 0xf62c8]                  // member 4
3.400439 agg1 out 2000:172:16:205::100 -> 2000:172:16:202::2: icmp6: echo request seq 1
[flowlabel 0xf62c8]
3.400441 port4 out 2000:172:16:205::100 -> 2000:172:16:202::2: icmp6: echo request seq 1
[flowlabel 0xf62c8]
3.400449 dmz out 2000:172:16:205::100 -> 2000:172:16:202::2: icmp6: echo request seq 1
[flowlabel 0xf62c8]                        // member 1
3.400663 vlan100 in 2000:172:16:202::2 -> 2000:172:16:205::100: icmp6: echo reply seq 1
3.400764 vlan100 in 2000:172:16:202::2 -> 2000:172:16:205::100: icmp6: echo reply seq 1
3.400778 vlan100 in 2000:172:16:202::2 -> 2000:172:16:205::100: icmp6: echo reply seq 1
3.400791 vlan100 in 2000:172:16:202::2 -> 2000:172:16:205::100: icmp6: echo reply seq 1
3.400913 port5 out 2000:172:16:202::2 -> 2000:172:16:205::100: icmp6: echo reply seq 1
3.400921 port5 out 2000:172:16:202::2 -> 2000:172:16:205::100: icmp6: echo reply seq 1
3.400926 port5 out 2000:172:16:202::2 -> 2000:172:16:205::100: icmp6: echo reply seq 1
3.400930 port5 out 2000:172:16:202::2 -> 2000:172:16:205::100: icmp6: echo reply seq 1
```

5. Edit the service rule to specify zone members 1 and 2 for duplication:

```
config sys sdwan
    config duplication
        edit 1
            set member 1 2
    end
end
```

6. Perform IPv4 and IPv6 ping from PC to server to confirm packets are copied only on zone members 1 and 2:

```
~$ ping 172.16.202.2
PING 172.16.202.2 (172.16.202.2) 56(84) bytes of data.
64 bytes from 172.16.202.2: icmp_seq=1 ttl=253 time=0.762 ms
64 bytes from 172.16.202.2: icmp_seq=1 ttl=253 time=0.762 ms (DUP!)
64 bytes from 172.16.202.2: icmp_seq=2 ttl=253 time=0.471 ms
64 bytes from 172.16.202.2: icmp_seq=2 ttl=253 time=0.471 ms (DUP!)
^C
--- 172.16.202.2 ping statistics ---
2 packets transmitted, 2 received, +2 duplicates, 0% packet loss, time 1010ms
rtt min/avg/max/mdev = 0.471/0.616/0.762/0.147 ms


FGT_A (root) (Interim)#  diagnose sniffer packet any 'host 172.16.202.2' 4
interfaces=[any]
```

```
filters=[host 172.16.202.2]
3.740162 port5 in 172.16.205.100 -> 172.16.202.2: icmp: echo request
3.740262 vd1-p1 out 172.16.205.100 -> 172.16.202.2: icmp: echo request                    //
member 2
3.740299 dmz out 172.16.205.100 -> 172.16.202.2: icmp: echo request                        //
member 1
3.740566 vd1-p1 in 172.16.202.2 -> 172.16.205.100: icmp: echo reply
3.740591 port5 out 172.16.202.2 -> 172.16.205.100: icmp: echo reply
3.740596 vd1-p1 in 172.16.202.2 -> 172.16.205.100: icmp: echo reply
3.740607 port5 out 172.16.202.2 -> 172.16.205.100: icmp: echo reply


~$ ping 2000:172:16:202::2
PING 2000:172:16:202::2(2000:172:16:202::2) 56 data bytes
64 bytes from 2000:172:16:202::2: icmp_seq=1 ttl=62 time=0.730 ms
64 bytes from 2000:172:16:202::2: icmp_seq=1 ttl=62 time=0.754 ms (DUP!)
64 bytes from 2000:172:16:202::2: icmp_seq=2 ttl=62 time=0.473 ms
64 bytes from 2000:172:16:202::2: icmp_seq=2 ttl=62 time=0.473 ms (DUP!)
^C
--- 2000:172:16:202::2 ping statistics ---
2 packets transmitted, 2 received, +2 duplicates, 0% packet loss, time 1024ms
rtt min/avg/max/mdev = 0.473/0.607/0.754/0.137 ms


FGT_A (root) (Interim)#  diagnose sniffer packet any 'host 2000:172:16:202::2' 4
interfaces=[any]
filters=[host 2000:172:16:202::2]
2.233077 port5 in 2000:172:16:205::100 -> 2000:172:16:202::2: icmp6: echo request seq 1
[flowlabel 0xf62c8]
2.233210 vd1-p1 out 2000:172:16:205::100 -> 2000:172:16:202::2: icmp6: echo request seq 1
[flowlabel 0xf62c8]                  // member 2
2.233243 dmz out 2000:172:16:205::100 -> 2000:172:16:202::2: icmp6: echo request seq 1
[flowlabel 0xf62c8]                     // member 1
2.233542 vd1-p1 in 2000:172:16:202::2 -> 2000:172:16:205::100: icmp6: echo reply seq 1
2.233572 port5 out 2000:172:16:202::2 -> 2000:172:16:205::100: icmp6: echo reply seq 1
2.233579 vd1-p1 in 2000:172:16:202::2 -> 2000:172:16:205::100: icmp6: echo reply seq 1
2.233590 port5 out 2000:172:16:202::2 -> 2000:172:16:205::100: icmp6: echo reply seq 1
```

Use secondary tunnel for FEC redundant parity packets

This information is also available in the FortiOS 8.0 Administration Guide:

- Use secondary tunnel for FEC redundant parity packets

In the original implementation of adaptive FEC, FEC parity packets are sent inside the same overlay tunnel. However, the parity packets are susceptible to being dropped at the same rate as other VPN packets.

This enhancement to adaptive FEC introduces the capability to send FEC parity packets in a secondary overlay tunnel that is associated with the same destination. The following setting can be configured on the primary tunnel that requires parity packets to be sent on a redundant tunnel:

```
config vpn ipsec phase1-interface
    edit <tunnel>
        set fec-separate-redundant-tunnel enable
    next
end
```

The redundant tunnel must have the same destination `location-id` as the original tunnel.

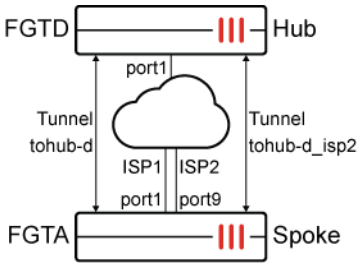
Example

In the following example, a VPN Hub (FGTD) is configured as a dialup VPN gateway. A spoke has two underlays going to different ISPs, where the primary (port1) has 10G bandwidth and the secondary (port9) has 1G bandwidth. Two tunnels are established from the Spoke to the Hub over the two underlays. The intention is to send data over the primary tunnel while using the secondary tunnel for redundancy and parity packets for the primary tunnel.

The FEC mappings are configured to send two parity packets per 8 base packets upon packet-loss threshold above 2%. Additionally, it will send three parity packets per nine base packets when bandwidth exceeds 950Mbps.

This example utilizes one Hub/destination, so the default location-id will be shared between the tunnels without any conflicts. When multiple Hubs are in use, and you need to apply the parity packet duplication on the same Hub and destination, use a different location-id for each hub.

```
config system settings
    set location-id <unique id for each hub>
end
```



To configure the hub, FGTD:

```
config vpn ipsec phase1-interface
    edit "tospokes"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dpd on-idle
        set dhgrp 20 21
        set auto-discovery-sender enable
```

```
        set fec-ingress enable
        set transport auto
        set ipv4-start-ip 100.100.100.10
        set ipv4-end-ip 100.100.100.20
        set ipv4-netmask 255.255.255.0
        set psksecret ****************
        set dpd-retryinterval 60
    next
end
```

```
config vpn ipsec phase2-interface
    edit "tospokes"
        set phase1name "tospokes"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

To configure the spoke, FGTA:

1. Configure FEC mapping to bind network SLA metrics and FEC base and redundant packets:

```
config vpn ipsec fec
    edit "m1"
        config mappings
            edit 1
                set base 8
                set redundant 2
                set packet-loss-threshold 2
            next
            edit 2
                set base 9
                set redundant 3
                set bandwidth-up-threshold 950000
            next
        end
    next
end
```

2. Configure the IPsec tunnels:

   On the primary tunnel, enable FEC enabled and apply FEC mapping. The redundant tunnel does not need FEC enabled, or the fec-separate-redundant-tunnel setting.

```
config vpn ipsec phase1-interface
    edit "tohub-d"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
```

```
chacha20poly1305-prfsha256
        set add-route disable
        set dhgrp 20 21
        set auto-discovery-receiver enable
        set auto-discovery-shortcuts dependent
        set fec-egress enable
        set fec-separate-redundant-tunnel enable
        set fec-codec rs
        set fec-health-check "1"
        set fec-mapping-profile "m1"
        set transport auto
        set remote-gw 172.16.200.4
        set psksecret ****************
    next
    edit "tohub-d_isp2"
        set interface "port9"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dhgrp 20 21
        set auto-discovery-receiver enable
        set auto-discovery-shortcuts dependent
        set transport auto
        set remote-gw 172.16.200.4
        set psksecret ****************
    next
end
```

```
config vpn ipsec phase2-interface
    edit "tohub-d"
        set phase1name "tohub-d"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
        set auto-negotiate enable
    next
    edit "tohub-d_isp2"
        set phase1name "tohub-d_isp2"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
        set auto-negotiate enable
    next
end
```

3. Configure SD-WAN:

   Assign the overlay interface as an SD-WAN member and attach it to a zone. Configure health-check on the overlay member in order to detect packet-loss and bandwidth utilization.

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port1"
            set gateway 172.16.200.3
        next
        edit 4
            set interface "tohub-d"
        next
        edit 5
            set interface "port9"
            set gateway 11.101.1.2
        next
    end
    config health-check
            edit "1"
            set server "192.168.5.44"
            set members 4
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 2
            set name "1_clone"
            set dst "all"
            set src "10.1.100.0"
            set priority-members 4 1
        next
    end
end
```

To verify the results:

1. Confirm that both tunnels are established on their respective interfaces on FGTA:

```
# diagnose vpn ike gateway list

vd: root/0
name: tohub-d_isp2
version: 2
interface: port9 15
addr: 11.101.1.1:5929 -> 172.16.200.4:4500
ext addr: 11.101.1.1 5929
tun_id: 10.0.0.2/::10.0.0.2
remote_location: 0.0.0.0
```

```
network-id: 0
transport: TCP
virtual-interface-addr: remote: 100.100.100.4
created: 407s ago
peer-id: 172.16.200.4
peer-id-auth: no
assigned IPv4 address: 100.100.100.11/255.255.255.0
nat: me
auto-discovery: 2 receiver
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 36000/36000/36000 ms
IPsec SA: created 1/1  established 1/1  time 36000/36000/36000 ms

  id/spi: 10217 77f9b317d1c985cd/cd40650eb175ec62
  direction: initiator
  status: established 407-371s ago = 36000ms
  proposal: aes128-sha256
  child: no
  SK_ei: 3fc9a5151af50e58-fec5cfed3c436fb1
  SK_er: de07bdc42f053f82-cdc945fc5a7ad5f4
  SK_ai: 0262f60ee41bafa2-2f7b9fc40f38e2b6-415b31885b81b778-b28c62a9915d2e5d
  SK_ar: 693d5e98d6695490-bc9c26b8a15e5322-0285e96df10e48cc-3535038e2c9a442d
  PPK: no
  message-id sent/recv: 2/0
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/85728
  DPD sent/recv: 00000000/00000000
  peer-id: 172.16.200.4

vd: root/0
name: tohub-d
version: 2
interface: port1 7
addr: 172.16.200.1:500 -> 172.16.200.4:500
ext addr: 172.16.200.1 500
tun_id: 172.16.200.4/::172.16.200.4
remote_location: 0.0.0.0
network-id: 0
transport: UDP
virtual-interface-addr: remote: 100.100.100.4
created: 377s ago
peer-id: 172.16.200.4
peer-id-auth: no
assigned IPv4 address: 100.100.100.10/255.255.255.0
auto-discovery: 2 receiver
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms
```

```
  id/spi: 10218 43bc6decd5bc56af/d5a405f1ff957a2f
  direction: initiator
  status: established 377-377s ago = 0ms
  proposal: aes128-sha256
  child: no
  SK_ei: 677aadd417ae719f-6cc99b2215be8711
  SK_er: f2b69ab1aecc08d6-04236a6849b56cae
  SK_ai: e9294b9b9087f973-9aebb458b27f5b13-39aab03bfee864bf-e616994e29cae238
  SK_ar: 7ef28f030c54ef5a-84c9765946e408da-d026e7294a209973-3d607b182eb86ec4
  PPK: no
  message-id sent/recv: 2/0
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/85722
  DPD sent/recv: 00000000/00000000
  peer-id: 172.16.200.4
```

2. Confirm that FEC is enabled in the egress direction for the tohub-d tunnel:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=tohub-d_isp2 ver=2 serial=5 11.101.1.1:5929->172.16.200.4:4500 nexthop=11.101.1.2 tun_
id=10.0.0.2 tun_id6=::10.0.0.2 status=up dst_mtu=1500 weight=1 country=ZZ
bound_if=15 real_if=15 lgwy=static/1 tun=intf mode=auto/1 encap=none options[0x228]=npu frag-
rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=5 ilast=31 olast=31 ad=r/2
stat: rxp=347 txp=22 rxb=458878 txb=1410
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=1802
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=tohub-d_isp2 proto=0 sa=1 ref=3 serial=1 auto-negotiate adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=1a227 type=00 soft=0 mtu=1438 expire=42519/0B replaywin=2048
       seqno=17 esn=0 replaywin_lastseq=0000015c qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42902/43200
  dec: spi=18e0b19e esp=aes key=16 c005a5c587fcd4ad85686868323f7a21
       ah=sha256 key=32 3da0b40b525f363355acb3dd445f0131c924b14bb3f704dfd8d7b205412e01d9
  enc: spi=c0e46a65 esp=aes key=16 a161fb71279cd02bed75614c16ce48d0
       ah=sha256 key=32 34977a64e77716c20d4fdfb4619f0745f8d850e448e0330a64baed64dbc24f91
  dec:pkts/bytes=347/458878, enc:pkts/bytes=22/2952
  npu_flag=00 npu_rgwy=0.0.0.0:0 npu_lgwy=0.0.0.0:0npu_selid=6
  dec_npuid=0 enc_npuid=0 dec_engid=-1 enc_engid=-1 dec_saidx=-1 enc_saidx=-1
------------------------------------------------------
name=tohub-d ver=2 serial=4 172.16.200.1:0->172.16.200.4:0 nexthop=172.16.200.4 tun_
id=172.16.200.4 tun_id6=::172.16.200.4 status=up dst_mtu=1500 weight=1 country=ZZ
bound_if=7 real_if=7 lgwy=static/1 tun=intf mode=auto/1 encap=none options[0x228]=npu frag-rfc
 run_state=0 role=primary accept_traffic=1 overlay_id=0
```

```
proxyid_num=1 child_num=0 refcnt=4 ilast=1 olast=1 ad=r/2
stat: rxp=375 txp=937 rxb=40092 txb=61856
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=16
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=1 ingress=0
proxyid=tohub-d proto=0 sa=1 ref=31 serial=1 auto-negotiate adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=1a227 type=00 soft=0 mtu=1342 expire=42509/0B replaywin=2048
       seqno=17aa esn=0 replaywin_lastseq=0000004d qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42898/43200
  dec: spi=18e0b19f esp=aes key=16 3e578ce917adbf940e9ad602baa71a92
       ah=sha256 key=32 4a23b8861918b8ad336cae572dd17f808e64ee22367a219d00249d06405736cf
  enc: spi=c0e46a64 esp=aes key=16 b5e7464c94b4847c30c804f7effa8a0d
       ah=sha256 key=32 bbde5831078147c8b3fd88d0d5112b6110cce75bdb5b23392cc3c2f90c9e4336
  dec:pkts/bytes=6/240, enc:pkts/bytes=937/125532
  npu_flag=03 npu_rgwy=172.16.200.4:0 npu_lgwy=172.16.200.1:0npu_selid=5
  dec_npuid=1 enc_npuid=1 dec_engid=-1 enc_engid=-1 dec_saidx=7399 enc_saidx=4
```

3. Confirm that the profile selected is sending three parity packets per nine base packets:

```
# diagnose vpn tunnel fec tohub-d
egress:  enabled=1
        base=9 redundant=3 codec=0 timeout=10(ms)
        encode=132 encode_timeout=129 encode_fail=0
        tx_data=166 tx_parity=396
        total_tx_data=166 total_tx_parity=396
ingress: enabled=0
        timeout=0(ms) fasm_cnt=0 fasm_full=0
        ipsec_fec_chk_fail=0 complete=0
        recover=0 recover_timeout=0 recover_fail=0
        rx_data=0 rx_parity=0
        rx=0 rx_fail=0
```

ToS matching and negate options on adaptive FEC profiles

This information is also available in the FortiOS 8.0 Administration Guide:

- ToS matching and negate options on adaptive FEC profiles

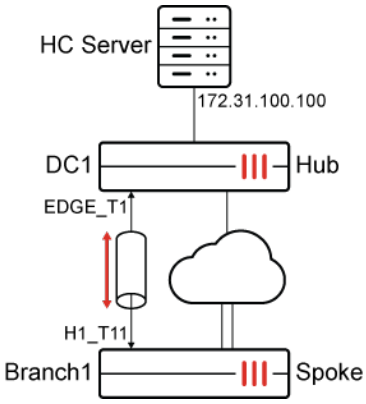
Adaptive FEC now has the ability to take the ToS/DSCP value into consideration when mapping the FEC profile. In the first round of matching, traffic will be matched against packet-loss, latency, and bandwidth. In the second round of matching, the TOS value and mask will be used. In addition, new negate options allow users to match against the negative of a threshold. For example, negate packet-loss of 5% means any packet-loss less than 5% will be matched.

For more information about how to configure ToS value and netmask and how it maps to DSCP values, please see Configuring SD-WAN rules.

New CLI:

```
config vpn ipsec fec
    edit <name>
        config mappings
            edit <id>
                set packet-loss-threshold-negate {enable | disable}
                set latency-threshold-negate {enable | disable}
                set bandwidth-up-threshold-negate {enable | disable}
                set bandwidth-down-threshold-negate {enable | disable}
                set bandwidth-bi-threshold-negate {enable | disable}
                config tos
                    edit <id>
                        set tos <value>
                        set tos-mask <value>
                        set base <value>
                        set redundant <value>
                    next
                end
            next
        end
    next
end
```

Example



In this single-hub SD-WAN deployment, the spoke has defined a performance health-check targeting the HC Server behind the Hub. Adaptive FEC is configured based on the health-checks using the following criteria:

	Health Check	Matching tos 0x01 mask 0xff	Matching tos 0x02 mask 0xff	Not matching either traffic class
Rule 1	Packet-loss is above the threshold of 5%	Apply base 5 redundant 1	Apply base 5 redundant 2	Apply default base 11 redundant 1
Rule 2	Latency is less than the threshold of 20ms	Apply base 6 redundant 1	Apply base 6 redundant 2	Apply default base 12 redundant 2

	Health Check	Matching tos 0x01 mask 0xff	Matching tos 0x02 mask 0xff	Not matching either traffic class
Rule 3	Available upstream bandwidth is more than 50Mbps	Apply base 7 redundant 1	Apply base 7 redundant 2	Apply default base 13 redundant 3
Rule 4	Packet-loss is below the threshold of 5%	Do not send any redundant packets		

The rules are examined in the order that they are configured.

To configure the spoke:

1. Configure SD-WAN:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "overlay"
        next
        edit "underlay"
        next
    end
    config members
        ...
        edit 3
            set interface "H1_T11"
            set zone "overlay"
            set source 172.31.0.65
        next
        ...
    end
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set update-static-route disable
            set members 3
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 1
            set dst "CORP_LAN"
            set priority-members 3
        next
```

```
    end
end
```

2. Configure VPN:

```
config vpn ipsec fec
    edit "fec_mapping"
        config mappings
            edit 1
                set base 11
                set redundant 1
                set packet-loss-threshold 5
                config tos
                    edit 1
                        set tos 0x01
                        set tos-mask 0xff
                        set base 5
                        set redundant 1
                    next
                    edit 2
                        set tos 0x02
                        set tos-mask 0xff
                        set base 5
                        set redundant 2
                    next
                end
            next
            edit 2
                set base 12
                set redundant 2
                set latency-threshold 20
                config tos
                    edit 1
                        set tos 0x01
                        set tos-mask 0xff
                        set base 6
                        set redundant 1
                    next
                    edit 2
                        set tos 0x02
                        set tos-mask 0xff
                        set base 6
                        set redundant 2
                    next
                end
            next
            edit 3
                set base 13
                set redundant 3
                set bandwidth-up-threshold 50000
                config tos
                    edit 1
```

```
                        set tos 0x01
                        set tos-mask 0xff
                        set base 7
                        set redundant 1
                    next
                    edit 2
                        set tos 0x02
                        set tos-mask 0xff
                        set base 7
                        set redundant 2
                    next
                end
            next
            edit 4
                set base 40
                set redundant 0
                set packet-loss-threshold 5
                set packet-loss-threshold-negate enable
            next
        end
    next
end
```

```
config vpn ipsec phase1-interface
    edit "H1_T11"
        ...
        set fec-egress enable
        set fec-codec rs
        set fec-health-check "HUB"
        set fec-mapping-profile "fec_mapping"
        ...
    next
end
```

3. Configure a firewall policy:

```
config firewall policy
    edit 1
       ...
        set fec enable
    next
end
```

To configure the hub:

```
config vpn ipsec phase1-interface
    edit "EDGE_T1"
        ...
        set fec-ingress enable
        ...
    next
end
```

To test the configuration:

1. Check SD-WAN metrics on H1_T11 and the applied FEC settings:
   In this first scenario, the first 3 rules were not matched. Therefore, the 4th rule implies that no redundant packets are sent.

```
Branch1_A_FGT (root) (Interim)# diagnose sys sdwan health-check
Health Check(HUB):
Seq(3 H1_T11): state(alive), packet-loss(0.000%), latency(30.247), jitter(0.016), mos(4.389),
custom_profile(0.000), bandwidth-up(49758), bandwidth-dw(89990), bandwidth-bi(139748), sla_
map=0x0

Branch1_A_FGT# diagnose vpn  tunnel  fec  H1_T11
egress:  enabled=1
        base=40 redundant=0 codec=0 timeout=10(ms)
        encode=0 encode_timeout=0 encode_fail=0
        tx_data=0 tx_parity=0
        total_tx_data=0 total_tx_parity=0
ingress: enabled=0
        timeout=0(ms) fasm_cnt=0 fasm_full=0
        ipsec_fec_chk_fail=0 complete=0
        recover=0 recover_timeout=0 recover_fail=0
        rx_data=0 rx_parity=0
        rx=0 rx_fail=0
```

2. Make the available upstream bandwidth more than 50M by decreasing traffic on H1_T11, then check the applied FEC settings. The third rule is matched:

```
# diagnose sys sdwan health-check
Health Check(HUB):
Seq(3 H1_T11): state(alive), packet-loss(0.000%), latency(30.234), jitter(0.022), mos(4.389),
custom_profile(0.000), bandwidth-up(63244), bandwidth-dw(89993), bandwidth-bi(153237), sla_
map=0x0

Branch1_A_FGT (root) (Interim)# diagnose vpn  tunnel  fec  H1_T11
egress:  enabled=1
        base=13 redundant=3 codec=0 timeout=10(ms)
        encode=0 encode_timeout=0 encode_fail=0
        tx_data=0 tx_parity=0
        tos/mask=0x01/0xFF:
                base=7 redundant=1
                encode=0 encode_timeout=0 encode_fail=0
                tx_data=0 tx_parity=0
        tos/mask=0x02/0xFF:
                base=7 redundant=2
                encode=0 encode_timeout=0 encode_fail=0
                tx_data=0 tx_parity=0
        total_tx_data=0 total_tx_parity=0
ingress: enabled=0
        timeout=0(ms) fasm_cnt=0 fasm_full=0
        ipsec_fec_chk_fail=0 complete=0
        recover=0 recover_timeout=0 recover_fail=0
```

```
        rx_data=0 rx_parity=0
        rx=0 rx_fail=0
```

3. Make latency on H1_T11 less than 20ms, then check the applied FEC settings again. Both the latency and upstream bandwidth are above the threshold, so because rule 2 comes first, rule 2 is applied:

```
# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(3 H1_T11): state(alive), packet-loss(0.000%), latency(0.247), jitter(0.027), mos(4.404),
custom_profile(0.000), bandwidth-up(89999), bandwidth-dw(89997), bandwidth-bi(179996), sla_
map=0x1

Branch1_A_FGT (root) (Interim)# diagnose vpn  tunnel  fec  H1_T11
egress:  enabled=1
        base=12 redundant=2 codec=0 timeout=10(ms)
        encode=0 encode_timeout=0 encode_fail=0
        tx_data=0 tx_parity=0
        tos/mask=0x01/0xFF:
                base=6 redundant=1
                encode=0 encode_timeout=0 encode_fail=0
                tx_data=0 tx_parity=0
        tos/mask=0x02/0xFF:
                base=6 redundant=2
                encode=0 encode_timeout=0 encode_fail=0
                tx_data=0 tx_parity=0
        total_tx_data=0 total_tx_parity=0
ingress: enabled=0
        timeout=0(ms) fasm_cnt=0 fasm_full=0
        ipsec_fec_chk_fail=0 complete=0
        recover=0 recover_timeout=0 recover_fail=0
        rx_data=0 rx_parity=0
        rx=0 rx_fail=0
```

4. Make packet-loss on H1_T11 more than 5%, then check the applied FEC settings again. All 3 rules match since health checks are above the threshold, so because rule 1 comes first, rule 1 is applied.

```
# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(3 H1_T11): state(alive), packet-loss(6.000%), latency(0.254), jitter(0.029), mos(4.401),
custom_profile(0.000), bandwidth-up(89999), bandwidth-dw(89997), bandwidth-bi(179996), sla_
map=0x0

Branch1_A_FGT (root) (Interim)# diagnose vpn  tunnel  fec  H1_T11
egress:  enabled=1
        base=11 redundant=1 codec=0 timeout=10(ms)
        encode=0 encode_timeout=0 encode_fail=0
        tx_data=0 tx_parity=0
        tos/mask=0x01/0xFF:
                base=5 redundant=1
                encode=0 encode_timeout=0 encode_fail=0
                tx_data=0 tx_parity=0
        tos/mask=0x02/0xFF:
```

```
                base=5 redundant=2
                encode=0 encode_timeout=0 encode_fail=0
                tx_data=0 tx_parity=0
        total_tx_data=0 total_tx_parity=0
ingress: enabled=0
        timeout=0(ms) fasm_cnt=0 fasm_full=0
        ipsec_fec_chk_fail=0 complete=0
        recover=0 recover_timeout=0 recover_fail=0
        rx_data=0 rx_parity=0
        rx=0 rx_fail=0
```

Control SD-WAN interface usage based on monthly traffic volume (quota)

Sometimes, SD-WAN underlay members may have monthly total bandwidth usage limits, which incurs overage costs with their ISPs once the limit is exceeded. To accomplish traffic steering based on monthly traffic volume, volume quota limit, billing start day, and related settings are added to SD-WAN members.

If accumulated traffic volume on one member exceeds the specified quota limit within one-month billing period, the member's cost, weight, or volume-ratio value can be automatically adjusted to force or redirect traffic to other members.

Traffic billing and steering based on monthly traffic volume supports high availability (HA).

The `config system sdwan` command has new options:

```
config system sdwan
    config members
      edit 1
        set quota-limit <integer>
        set billing-start-day <integer>
        set overage {enable | disable}
        set overage-cost <integer>
        set overage-weight <integer>
        set overage-volume-ratio <integer>
      next
    end
endd
```

Option	Description
`quota-limit <integer>`	Volume quota limit assigned to this member in gigabytes (0 - 10485760, default = 0).
`billing-start-day <integer>`	Volume billing start day when this member's volume usage will begin to calculate (1 - 31, default = 1). Available when `quota-limit` is set.

Option	Description
`overage {enable | disable}`	Enable/disable the volume overage when member's volume usage reaches quota-limit. Available when `quota-limit` is set.
`overage-cost <integer>`	Cost value for this member when its volume is over quota and overage is enabled (0 - 4294967295, default = 0). Available when `quota-limit` is set.
`overage-weight <integer>`	Weight value for this member when its volume is over quota and overage is enabled (0 - 255, default = 1). Available when `quota-limit` is set.
`overage-volume-ratio <integer>`	Volume ratio value for this member when its volume is over quota and overage is enabled (1 - 255, default = 1). Available when `quota-limit` is set.

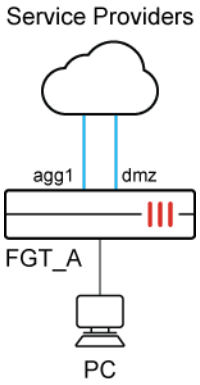
A new diagnose command is available:

```
# diagnose sys sdwan bill  member 1
```

Example

In this example,

- FGT_A has two SD-WAN members: agg1 and dmz.
- The service provider grants 2 GB bandwidth to agg1 per month. If 2 GB bandwidth on agg1 is reached within the one month billing period, traffic will switch to dmz for the rest of the billing period by increasing agg1's cost to deprioritize agg1.
- On the next billing day, 2 GB quota and cost will be reset and traffic switches back to agg1.



To configure interface usage based on monthly traffic volume:

1. Configure SD-WAN members.
   In this example:

- Traffic volume limit is 2 GB within one-month billing period (`quota-limit 2`).
- The start day is the first day in the month (`billing-start-day 1`).
- The overage function is enabled (`overage enable`).
- Cost 10 will be set when accumulated traffic volume on it exceeds 2 GB (`overage-cost 10`).
- The overage weight will be set when accumulated traffic volume on it exceeds 2 GB (`overage-weight 1`).
- The volume ratio will be set when accumulated traffic volume on it exceeds 2 GB (`overage-volume-ratio 1`).

In addition, `mode sla` is set for IPv4 and IPv6 services.

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "agg1"
            set gateway 172.16.203.2
            set gateway6 2000:172:16:203::2
            set quota-limit 2
            set billing-start-day 1
            set overage enable
            set overage-cost 10
            set overage-weight 1
            set overage-volume-ratio 1
        next
        edit 2
            set interface "dmz"
            set gateway 172.16.208.2
            set gateway6 2000:172:16:208::2
        next
    end
    config health-check
        edit "1"
            set server "2.2.2.2"
            set update-static-route disable
            set members 0
            config sla
                edit 1
                next
            end
        next
        edit "2"
            set addr-mode ipv6
            set server "2000::2:2:2:2"
            set update-static-route disable
            set members 0
            config sla
                edit 1
                next
```

```
            end
        next
    end
    config service
        edit 1
            set name "1"
            set mode sla
            set dst "all"
            set src "172.16.205.0"
            config sla
                edit "1"
                    set id 1
                next
            end
            set priority-members 1 2
        next
        edit 2
            set name "2"
            set addr-mode ipv6
            set mode sla
            config sla
                edit "2"
                    set id 1
                next
            end
            set priority-members 1 2
            set dst6 "all"
            set src6 "2000:172:16:205::0"
        next
    end
end
```

2. Check SD-WAN status:
   The agg1 interface is preferred to forward IPv4 and IPv6 traffic. The 2 GB quota limit has not been reached.

```
# diagnose sys sdwan  service4

Service(1): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 2
  Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 agg1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(2 dmz virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
  Src address(1):
        172.16.205.0-172.16.205.255
  Dst address(1):
        0.0.0.0-255.255.255.255

# diagnose sys sdwan  service6
```

```
Service(2): Address Mode(IPV6) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 2
  Gen(2), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 agg1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(2 dmz virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
  Src6 address(1): 2000:172:16:205::/64
  Dst6 address(1): ::/0

# diagnose sys sdwan bill member 1
Member interface(agg1): Volume overage:enable, Quota_limit:2(GB), Volume is not overaged,
Reset_in_billing_day(0), Intf_set:up, Overage_cost(10), Overage_weight(1), Overage_volume(1),
Billing_start_day:1, Reset_accumulated_bytes:0, Current_bytes:234756493, Billing_start_
bytes:221780698, Billing_latest_bytes:234756049 Used_bytes since billing day:12975351
```

3. From the PC, initiate traffic.
   The traffic matches SD-WAN SLA-mode service and goes through agg1 on FGT_A for IPv4 and IPv6 traffic:

```
# diagnose sniffer packet any 'host 2.0.0.1' 4
interfaces=[any]
filters=[host 2.0.0.1]
4137.955796 port5 in 172.16.205.100 -> 2.0.0.1: icmp: echo request
4137.955895 agg1 out 172.16.205.100 -> 2.0.0.1: icmp: echo request
4137.955901 port4 out 172.16.205.100 -> 2.0.0.1: icmp: echo request
4137.956171 port4 in 2.0.0.1 -> 172.16.205.100: icmp: echo reply
4137.956176 agg1 in 2.0.0.1 -> 172.16.205.100: icmp: echo reply
4137.956203 port5 out 2.0.0.1 -> 172.16.205.100: icmp: echo reply


# diagnose sniffer packet any 'host 2000::2:0:0:1' 4
interfaces=[any]
filters=[host 2000::2:0:0:1]
4126.153276 port5 in 2000:172:16:205::100 -> 2000::2:0:0:1: icmp6: echo request seq 1
[flowlabel 0xb1863]
4126.153392 agg1 out 2000:172:16:205::100 -> 2000::2:0:0:1: icmp6: echo request seq 1
[flowlabel 0xb1863]
4126.153399 port4 out 2000:172:16:205::100 -> 2000::2:0:0:1: icmp6: echo request seq 1
[flowlabel 0xb1863]
4126.153577 port4 in 2000::2:0:0:1 -> 2000:172:16:205::100: icmp6: echo reply seq 1
4126.153585 agg1 in 2000::2:0:0:1 -> 2000:172:16:205::100: icmp6: echo reply seq 1
4126.153629 port5 out 2000::2:0:0:1 -> 2000:172:16:205::100: icmp6: echo reply seq 1
```

4. When accumulated traffic volume on agg1 exceeds 2 GB, traffic switches to dmz because the cost is increased to 10 in memory.

```
# diagnose sys sdwan bill  member 1
Member interface(agg1): Volume overage:enable, Quota_limit:2(GB), Volume is overaged, Reset_
in_billing_day(0) Intf_set:up, Overage_cost(10), Overage_weight(1), Overage_volume(1),
```

```
Billing_start_day:1,Reset_accumulated_bytes:0, Current_bytess:2231650140, Billing_start_
bytes:221780698, Billing_latest_bytes:2231647804 Used_bytes since billing day:2009867106

# diagnose sys sdwan  service4

Service(1): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 2
  Gen(2), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(2 dmz virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    2: Seq_num(1 agg1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
  Src address(1):
        172.16.205.0-172.16.205.255
  Dst address(1):
        0.0.0.0-255.255.255.255

# diagnose sys sdwan  service6

Service(2): Address Mode(IPV6) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 2
  Gen(2), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(2 dmz virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    2: Seq_num(1 agg1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
  Src6 address(1): 2000:172:16:205::/64
  Dst6 address(1): ::/0

# diagnose sniffer packet any 'host 2.0.0.1' 4
interfaces=[any]
filters=[host 2.0.0.1]
....
6012.130521 port5 in 172.16.205.100 -> 2.0.0.1: icmp: echo request
6012.130612 dmz out 172.16.205.100 -> 2.0.0.1: icmp: echo request
6012.130784 dmz in 2.0.0.1 -> 172.16.205.100: icmp: echo reply
6012.130860 port5 out 2.0.0.1 -> 172.16.205.100: icmp: echo reply
6013.154504 port5 in 172.16.205.100 -> 2.0.0.1: icmp: echo request
6013.154557 dmz out 172.16.205.100 -> 2.0.0.1: icmp: echo request

# diagnose sniffer packet any 'host 2000::2:0:0:1' 4
interfaces=[any]
filters=[host 2000::2:0:0:1]
......
6002.927740 port5 in 2000:172:16:205::100 -> 2000::2:0:0:1: icmp6: echo request seq 1835
[flowlabel 0xb1863]
6002.927818 dmz out 2000:172:16:205::100 -> 2000::2:0:0:1: icmp6: echo request seq 1835
[flowlabel 0xb1863]
```

```
6002.927962 dmz in 2000::2:0:0:1 -> 2000:172:16:205::100: icmp6: echo reply seq 1835
6002.928009 port5 out 2000::2:0:0:1 -> 2000:172:16:205::100: icmp6: echo reply seq 1835
6003.947726 port5 in 2000:172:16:205::100 -> 2000::2:0:0:1: icmp6: echo request seq 1836
[flowlabel 0xb1863]
6003.947779 dmz out 2000:172:16:205::100 -> 2000::2:0:0:1: icmp6: echo request seq 1836
[flowlabel 0xb1863]
```

5. On the next billing day, 2 GB quota and cost are reset, and traffic switches back to agg1.

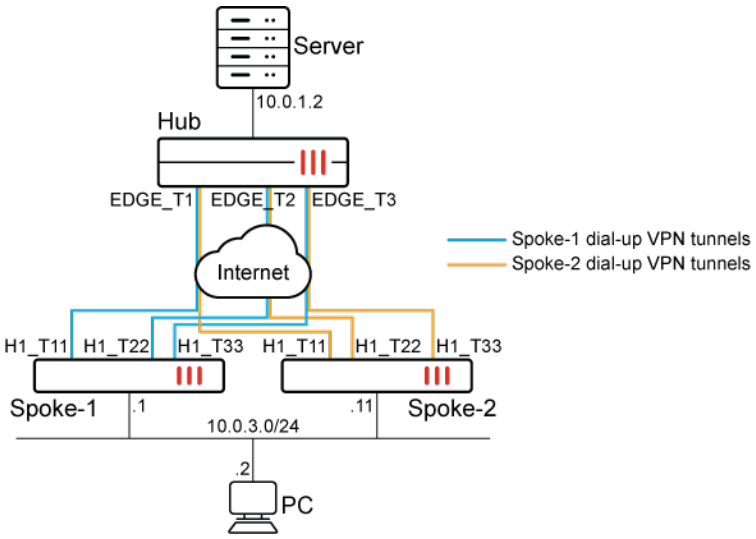
```
# diagnose sys sdwan bill  member 1
Member interface(agg1): Volume overage:enable, Quota_limit:2(GB), Volume is not overaged,
Reset_in_billing_day(1) Intf_set:up, Overage_cost(10), Overage_weight(1), Overage_volume(1),
Billing_start_day:1,Reset_accumulated_bytes:0, Current_bytess:2243437973, Billing_start_
bytes:2233333177, Billing_latest_bytes:2237462051 Used_bytes since billing day:4128874

# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 2
  Gen(3), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 agg1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(2 dmz virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
  Src address(1):
        172.16.205.0-172.16.205.255
  Dst address(1):
        0.0.0.0-255.255.255.255

# diagnose sys sdwan service6

Service(2): Address Mode(IPV6) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 2
  Gen(3), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 agg1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(2 dmz virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
  Src6 address(1): 2000:172:16:205::/64
  Dst6 address(1): ::/0

# diagnose sniffer packet any 'host 2.0.0.1' 4
interfaces=[any]
filters=[host 2.0.0.1]

......

352023.096120 port5 in 172.16.205.100 -> 2.0.0.1: icmp: echo request
```

```
352023.096170 agg1 out 172.16.205.100 -> 2.0.0.1: icmp: echo request
352023.096175 port4 out 172.16.205.100 -> 2.0.0.1: icmp: echo request
352023.096325 port4 in 2.0.0.1 -> 172.16.205.100: icmp: echo reply
352023.096331 agg1 in 2.0.0.1 -> 172.16.205.100: icmp: echo reply
352023.096373 port5 out 2.0.0.1 -> 172.16.205.100: icmp: echo reply
352024.120110 port5 in 172.16.205.100 -> 2.0.0.1: icmp: echo request
352024.120149 agg1 out 172.16.205.100 -> 2.0.0.1: icmp: echo request
352024.120153 port4 out 172.16.205.100 -> 2.0.0.1: icmp: echo request

# diagnose sniffer packet any 'host 2000::2:0:0:1' 4
interfaces=[any]
filters=[host 2000::2:0:0:1]
.......
352013.825337 port5 in 2000:172:16:205::100 -> 2000::2:0:0:1: icmp6: echo request seq 2237
[flowlabel 0xb1863]
352013.825413 agg1 out 2000:172:16:205::100 -> 2000::2:0:0:1: icmp6: echo request seq 2237
[flowlabel 0xb1863]
352013.825418 port4 out 2000:172:16:205::100 -> 2000::2:0:0:1: icmp6: echo request seq 2237
[flowlabel 0xb1863]
352013.825594 port4 in 2000::2:0:0:1 -> 2000:172:16:205::100: icmp6: echo reply seq 2237
352013.825601 agg1 in 2000::2:0:0:1 -> 2000:172:16:205::100: icmp6: echo reply seq 2237
352013.825643 port5 out 2000::2:0:0:1 -> 2000:172:16:205::100: icmp6: echo reply seq 2237
352014.849326 port5 in 2000:172:16:205::100 -> 2000::2:0:0:1: icmp6: echo request seq 2238
[flowlabel 0xb1863]
352014.849377 agg1 out 2000:172:16:205::100 -> 2000::2:0:0:1: icmp6: echo request seq 2238
[flowlabel 0xb1863]
352014.849382 port4 out 2000:172:16:205::100 -> 2000::2:0:0:1: icmp6: echo request seq 2238
[flowlabel 0xb1863]
```

Confine packet duplication across multiple overlays to the same spoke

Packet duplication can be confined across multiple dial-up tunnels to the same spoke. When the same subnet is behind multiple spokes, a hub device will use the location ID setting to uniquely identify and limit the destination for duplicated traffic.

Example



In this example, the Hub has three dial-up tunnels to Spoke-1 and Spoke-2. Duplication, regardless of SLA status, and de-duplication are enabled on the Hub and spokes, and four copies are allowed for duplication.

Since both spokes are connected to the same LAN subnet (10.0.3.0/24), we can optimize duplication by confining the duplication to the Spoke that initiated the traffic based on its location-id. This example demonstrates that when traffic from the PC is sent through Spoke-1, duplication occurs only on the 3 overlays connected to Spoke-1, even when duplication allows 4 copies to be duplicated.

To configure the Hub:

```
config system sdwan
    set status enable
    set duplication-max-num 4
    config zone
     ...
        edit "overlay"
        next
    end
    config members
        edit 1
            set interface "EDGE_T1"
            set zone "overlay"
        next
        edit 2
            set interface "EDGE_T2"
            set zone "overlay"
        next
        edit 3
            set interface "EDGE_T3"
            set zone "overlay"
        next
    ...
```

```
    end
    config duplication
        edit 1
            set srcaddr "CORP_LAN"
            set dstaddr "CORP_LAN"
            set srcintf "lan_zone" "overlay"
            set dstintf "overlay" "lan_zone"
            set service "PING"
            set packet-duplication force
            set packet-de-duplication enable
        next
    end
end
```

```
config system settings
    set location-id 172.31.0.1
end
```

To configure Spoke-1:

```
config system sdwan
    set status enable
    set duplication-max-num 4
    config zone
    ...
        edit "overlay"
        next
    end
    config members
       ...
        edit 4
            set interface "H1_T11"
            set zone "overlay"
            set source 172.31.0.65
            set priority 10
        next
        edit 5
            set interface "H1_T22"
            set zone "overlay"
            set source 172.31.0.65
            set priority 10
        next
        edit 6
            set interface "H1_T33"
            set zone "overlay"
            set source 172.31.0.65
            set priority 10
        next
      ...
    end
    config duplication
```

```
        edit 1
            set srcaddr "CORP_LAN"
            set dstaddr "CORP_LAN"
            set srcintf "lan_zone" "overlay"
            set dstintf "overlay" "lan_zone"
            set service "PING"
            set packet-duplication force
            set packet-de-duplication enable
        next
    end
end
```

```
config system settings
    set location-id 172.31.0.65
end
```

To configure Spoke-2:

```
config system sdwan
    set status enable
    set duplication-max-num 4
    config zone
    ...
        edit "overlay"
        next
    end
    config members
       ...
        edit 4
            set interface "H1_T11"
            set zone "overlay"
            set source 172.31.0.66
            set priority 10
        next
        edit 5
            set interface "H1_T22"
            set zone "overlay"
            set source 172.31.0.66
            set priority 10
        next
        edit 6
            set interface "H1_T33"
            set zone "overlay"
            set source 172.31.0.66
            set priority 10
        next
      ...
    end
    config duplication
        edit 1
            set srcaddr "CORP_LAN"
```

```
            set dstaddr "CORP_LAN"
            set srcintf "lan_zone" "overlay"
            set dstintf "overlay" "lan_zone"
            set service "PING"
            set packet-duplication force
            set packet-de-duplication enable
        next
    end
end
```

```
config system settings
    set location-id 172.31.0.66
end
```

To check the routes:

1. On the hub, route 10.0.3.0/24 can be learned from both Spoke-1 and Spoke-2 and it is then possible to duplicate traffic to Spoke-1 and Spoke-2, but with this enhancement, traffic is only allowed to be duplicated to the same spoke.

```
# get router info routing-table details 10.0.3.0/24

Routing table for VRF=0
Routing entry for 10.0.3.0/24
  Known via "bgp", distance 200, metric 0, best
  Last update 01:35:35 ago
  * vrf 0 172.31.0.65 priority 1 (recursive via EDGE_T3 tunnel 10.0.0.7 [1])
                                                (recursive via EDGE_T1 tunnel 172.31.0.65 [10])
                                                (recursive via EDGE_T2 tunnel 10.0.0.6 [15])
  * vrf 0 172.31.0.66 priority 1 (recursive via EDGE_T3 tunnel 172.31.0.66 [1])
                                                (recursive via EDGE_T1 tunnel 10.0.0.4 [10])
                                                (recursive via EDGE_T2 tunnel 10.0.0.5 [15])
```

2. On Spoke-1, route 10.0.1.0/24 is learned from the Hub.

```
# get router info routing-table details 10.0.1.0/24

Routing table for VRF=0
Routing entry for 10.0.1.0/24
  Known via "bgp", distance 200, metric 0, best
  Last update 01:52:01 ago
  * vrf 0 172.31.0.1, tag 1 priority 1 (recursive via H1_T11 tunnel 172.31.1.1), best-match
                                                        (recursive via H1_T22 tunnel
172.31.1.5), best-match
                                                        (recursive via H1_T33 tunnel
172.31.2.1), best-match
```

3. From the PC, ping the server through Spoke-1:

```
> ping 10.0.1.2

Pinging 10.0.1.2 with 32 bytes of data:
```

```
Reply from 10.0.1.2: bytes=32 time<1ms TTL=62
Ping statistics for 10.0.1.2:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

4. On Spoke-1, traffic is duplicated to all tunnels (H1_11, H1_T22, and H1_T33), duplicated traffic is received from the Hub, and traffic is de-duplicated then sent to the PC:

```
# diagnose sniffer packet any 'host 10.0.1.2' 4
interfaces=[any]
filters=[host 10.0.1.2]
8.129527 port4 in 10.0.3.2 -> 10.0.1.2: icmp: echo request
8.129913 H1_T22 out 10.0.3.2 -> 10.0.1.2: icmp: echo request                    // traffic is
duplicated to all tunnel H1_11, H1_T22 and H1_T33
8.129939 H1_T33 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
8.129961 H1_T11 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
8.130481 H1_T11 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply                           //
duplicated traffic is received from Hub
8.130504 H1_T33 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
8.130521 H1_T22 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
8.130565 port4 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply                           //
traffic is de-duplicated and sent to PC
```

5. On the Hub, duplicated traffic is received from Spoke-1, traffic is de-duplicated and sent to the server, and traffic is only duplicated to dial-up tunnels destined for Spoke-1 :

```
#  diagnose sniffer packet any 'host 10.0.1.2' 4
interfaces=[any]
filters=[host 10.0.1.2]
3.905902 EDGE_T3 in 10.0.3.2 -> 10.0.1.2: icmp: echo request
3.905910 EDGE_T2 in 10.0.3.2 -> 10.0.1.2: icmp: echo request
3.905951 EDGE_T1 in 10.0.3.2 -> 10.0.1.2: icmp: echo request
3.906110 port4 out 10.0.3.2 -> 10.0.1.2: icmp: echo request
3.906213 port4 in 10.0.1.2 -> 10.0.3.2: icmp: echo reply
3.906295 EDGE_T1 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
3.906332 EDGE_T2 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
3.906360 EDGE_T3 out 10.0.1.2 -> 10.0.3.2: icmp: echo reply
```

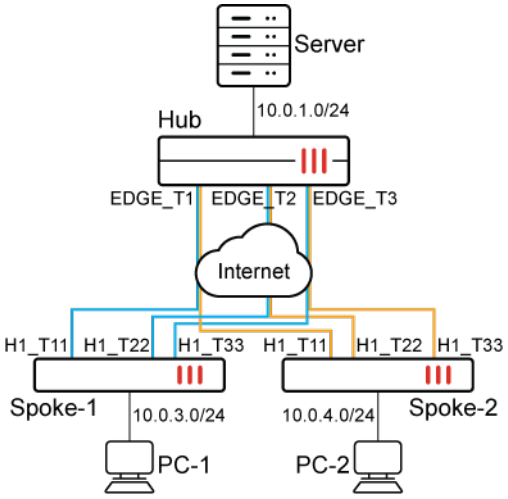
6. On Spoke-2, the Hub echo reply packets are not duplicated to dial-up tunnels destined for Spoke-2:

```
# diagnose sniffer packet any 'host 10.0.1.2' 4
interfaces=[any]
filters=[host 10.0.1.2]
```

On-demand duplication at the Hub using remote health-check

Packet duplication on hub devices can be configured to automatically activate when a spoke begins sending out-of-SLA remote health checks. Duplication continues until the hub once again receives in-SLA health checks from

that spoke. The duplication is limited to the affected spoke's tunnels. For each original packet, the hub sends one duplicate per tunnel until either all tunnels have transmitted the same packet or the configured maximum (`duplication-max-num`) is reached.



In this example, the Hub has three dial-up tunnels to Spoke-1 and Spoke-2. On-demand duplication is configured on the Hub, and active health-check is configured on the spokes to detect the overlays' SLA.

To configure the Hub:

```
config system sdwan
    set status enable
    set duplication-max-num 4
    config zone
        ...
        edit "overlay"
        next
    end
    config members
        edit 1
            set interface "EDGE_T1"
            set zone "overlay"
        next
        edit 2
            set interface "EDGE_T2"
            set zone "overlay"
        next
        edit 3
            set interface "EDGE_T3"
            set zone "overlay"
        next
       ...
    end
    config health-check
        edit "passive_hc"
            set detect-mode remote
            set sla-id-redistribute 1
```

```
            set members 1 2 3
            config sla
                edit 1
                    set link-cost-factor remote
                next
            end
        next
    end
    ...
    config duplication
        edit 1
            set srcaddr "CORP_LAN"
            set dstaddr "CORP_LAN"
            set srcintf "lan_zone" "overlay"
            set dstintf "overlay" "lan_zone"
            set service "PING"
            set packet-duplication on-demand
        next
    end
end
```

To configure Spoke-1:

```
config system sdwan
    set status enable
    config zone
        ...
        edit "overlay"
        next
    end
    config members
        ...
        edit 4
            set interface "H1_T11"
            set zone "overlay"
            set source 172.31.0.65
            set priority 10
        next
        edit 5
            set interface "H1_T22"
            set zone "overlay"
            set source 172.31.0.65
            set priority 10
        next
        edit 6
            set interface "H1_T33"
            set zone "overlay"
            set source 172.31.0.65
            set priority 10
        next
        ...
    end
```

```
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set embed-measured-health enable
            set sla-id-redistribute 1
            set members 4 5 6
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
    ...
end
```

To configure Spoke-2:

```
config system sdwan
    set status enable
    config zone
    ...
        edit "overlay"
    end
    config members
        edit 4
            set interface "H1_T11"
            set zone "overlay"
            set source 172.31.0.66
            set cost 2
            set priority 10
        next
        edit 5
            set interface "H1_T22"
            set zone "overlay"
            set source 172.31.0.66
            set priority 10
        next
        edit 6
            set interface "H1_T33"
            set zone "overlay"
            set source 172.31.0.66
            set priority 10
        next
    ...
    end
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set embed-measured-health enable
            set sla-id-redistribute 1
```

```
            set members 4 5 6
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 200
                next
            end
        next
    end
   ...
end
```

To check the routes:

1. On the Hub, check the subnets behind Spoke-1 (10.0.3.0/24) and Spoke-2 (10.0.4.0/24):

```
# get router info routing-table details 10.0.3.0/24

Routing table for VRF=0
Routing entry for 10.0.3.0/24
  Known via "bgp", distance 200, metric 0, best
  Last update 01:10:23 ago
  * vrf 0 172.31.0.65 priority 1 (recursive via EDGE_T1 tunnel 172.31.0.65 [1])
                                 (recursive via EDGE_T2 tunnel 10.0.0.14 [1])
                                 (recursive via EDGE_T3 tunnel 10.0.0.15 [1])
```

```
# get router info routing-table details 10.0.4.0/24

Routing table for VRF=0
Routing entry for 10.0.4.0/24
  Known via "bgp", distance 200, metric 0, best
  Last update 01:09:45 ago
  * vrf 0 172.31.0.66 priority 1 (recursive via EDGE_T1 tunnel 172.31.0.66 [1])
                                 (recursive via EDGE_T2 tunnel 10.0.0.16 [1])
                                 (recursive via EDGE_T3 tunnel 10.0.0.17 [1])
```

2. On Spoke-1, check the subnet behind the Hub:

```
# get router info routing-table details 10.0.1.0/24

Routing table for VRF=0
Routing entry for 10.0.1.0/24
  Known via "bgp", distance 200, metric 0, best
  Last update 01:40:18 ago
  * vrf 0 172.31.0.1, tag 1 priority 1 (recursive via H1_T11 tunnel 172.31.1.1), best-match
                                       (recursive via H1_T22 tunnel 172.31.1.5), best-match
                                       (recursive via H1_T33 tunnel 172.31.2.1), best-match
```

3. On Spoke-2, check the subnet behind the Hub:

```
# get router info routing-table details 10.0.1.0/24
```

```
Routing table for VRF=0
Routing entry for 10.0.1.0/24
  Known via "bgp", distance 200, metric 0, best
  Last update 01:11:37 ago
  * vrf 0 172.31.0.1, tag 1 priority 1 (recursive via H1_T11 tunnel 172.31.1.1), best-match
                                       (recursive via H1_T22 tunnel 172.31.1.5), best-match
                                       (recursive via H1_T33 tunnel 172.31.2.1), best-match
```

4. On the Hub, check the remote health status and that all child tunnels are in-SLA:

```
# diagnose sys sdwan health-check remote
Remote Health Check: passive_hc(3)
  Passive remote statistics of EDGE_T3(47):
EDGE_T3_0(10.0.0.15): timestamp=02-19 12:10:03.556, src=172.31.0.65, latency=0.150,
jitter=0.007, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_prio=0,
last_sla_change=02-19 10:23:45.877
EDGE_T3_1(10.0.0.17): timestamp=02-19 12:10:03.368, src=172.31.0.66, latency=0.159,
jitter=0.022, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_prio=0,
last_sla_change=02-19 10:51:17.377
Remote Health Check: passive_hc(2)
  Passive remote statistics of EDGE_T2(46):
EDGE_T2_1(10.0.0.14): timestamp=02-19 12:10:03.556, src=172.31.0.65, latency=0.210,
jitter=0.004, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_prio=0,
last_sla_change=02-19 10:21:44.967
EDGE_T2_0(10.0.0.16): timestamp=02-19 12:10:03.368, src=172.31.0.66, latency=0.229,
jitter=0.017, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_prio=0,
last_sla_change=02-19 10:51:08.667
Remote Health Check: passive_hc(1)
  Passive remote statistics of EDGE_T1(45):
EDGE_T1_1(172.31.0.65): timestamp=02-19 12:10:03.556, src=172.31.0.65, latency=0.262,
jitter=0.019, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_prio=0,
last_sla_change=02-19 12:05:34.137
EDGE_T1_0(172.31.0.66): timestamp=02-19 12:10:03.368, src=172.31.0.66, latency=0.285,
jitter=0.021, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_prio=0,
last_sla_change=02-19 10:51:08.157
```

5. Ping from the server to PC-1 and PC-2 and on the Hub, confirm that traffic goes on EDGE_T1 and no duplication occurs because the child tunnels of EDGE_T1 are in-SLA.
   a. PC-1:

```
test@server:~$ ping 10.0.3.2
PING 10.0.3.2 (10.0.3.2) 56(84) bytes of data.
64 bytes from 10.0.3.2: icmp_seq=1 ttl=62 time=1.09 ms
64 bytes from 10.0.3.2: icmp_seq=2 ttl=62 time=0.684 ms
64 bytes from 10.0.3.2: icmp_seq=3 ttl=62 time=0.679 ms
```

   Packet sniffer on the Hub:

```
# diagnose sniffer packet any 'host 10.0.3.2' 4
interfaces=[any]
filters=[host 10.0.3.2]
39.588602 port4 in 10.0.1.2 -> 10.0.3.2: icmp: echo request
```

```
39.588763 EDGE_T1 out 10.0.1.2 -> 10.0.3.2: icmp: echo request
39.589555 EDGE_T1 in 10.0.3.2 -> 10.0.1.2: icmp: echo reply
39.589626 port4 out 10.0.3.2 -> 10.0.1.2: icmp: echo reply
40.588399 port4 in 10.0.1.2 -> 10.0.3.2: icmp: echo request
40.588472 EDGE_T1 out 10.0.1.2 -> 10.0.3.2: icmp: echo request
40.588946 EDGE_T1 in 10.0.3.2 -> 10.0.1.2: icmp: echo reply
40.588998 port4 out 10.0.3.2 -> 10.0.1.2: icmp: echo reply
41.587411 port4 in 10.0.1.2 -> 10.0.3.2: icmp: echo request
41.587491 EDGE_T1 out 10.0.1.2 -> 10.0.3.2: icmp: echo request
41.587972 EDGE_T1 in 10.0.3.2 -> 10.0.1.2: icmp: echo reply
41.588028 port4 out 10.0.3.2 -> 10.0.1.2: icmp: echo reply
```

b. PC-2:

```
test@server:~$ ping 10.0.4.2
PING 10.0.4.2 (10.0.4.2) 56(84) bytes of data.
64 bytes from 10.0.4.2: icmp_seq=1 ttl=62 time=0.959 ms
64 bytes from 10.0.4.2: icmp_seq=2 ttl=62 time=0.630 ms
64 bytes from 10.0.4.2: icmp_seq=3 ttl=62 time=0.650 ms
```

Packet sniffer on the Hub:

```
# diagnose sniffer packet any 'host 10.0.4.2' 4
interfaces=[any]
filters=[host 10.0.4.2]
2.598946 port4 in 10.0.1.2 -> 10.0.4.2: icmp: echo request
2.599055 EDGE_T1 out 10.0.1.2 -> 10.0.4.2: icmp: echo request
2.599777 EDGE_T1 in 10.0.4.2 -> 10.0.1.2: icmp: echo reply
2.599810 port4 out 10.0.4.2 -> 10.0.1.2: icmp: echo reply
3.598778 port4 in 10.0.1.2 -> 10.0.4.2: icmp: echo request
3.598819 EDGE_T1 out 10.0.1.2 -> 10.0.4.2: icmp: echo request
3.599262 EDGE_T1 in 10.0.4.2 -> 10.0.1.2: icmp: echo reply
3.599295 port4 out 10.0.4.2 -> 10.0.1.2: icmp: echo reply
4.598782 port4 in 10.0.1.2 -> 10.0.4.2: icmp: echo request
4.598819 EDGE_T1 out 10.0.1.2 -> 10.0.4.2: icmp: echo request
4.599279 EDGE_T1 in 10.0.4.2 -> 10.0.1.2: icmp: echo reply
4.599316 port4 out 10.0.4.2 -> 10.0.1.2: icmp: echo reply
```

6. Increase the latency to 120ms on overlay H1_T11 of Spoke-1, then confirm that the corresponding child tunnel on the Hub is out-of-SLA. Also confirm that duplication occurs from the Hub to Spoke-1.

a. Check the health status on the HUB:

```
# diagnose sys sdwan health-check  remote
Remote Health Check: passive_hc(3)
  Passive remote statistics of EDGE_T3(47):
EDGE_T3_0(10.0.0.15): timestamp=02-19 13:28:14.680, src=172.31.0.65, latency=0.158,
jitter=0.009, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_
prio=0, last_sla_change=02-19 10:23:45.877
EDGE_T3_1(10.0.0.17): timestamp=02-19 13:28:14.504, src=172.31.0.66, latency=0.149,
jitter=0.019, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_
prio=0, last_sla_change=02-19 10:51:17.377
Remote Health Check: passive_hc(2)
```

```
  Passive remote statistics of EDGE_T2(46):
EDGE_T2_1(10.0.0.14): timestamp=02-19 13:28:14.680, src=172.31.0.65, latency=0.220,
jitter=0.009, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_
prio=0, last_sla_change=02-19 10:21:44.967
EDGE_T2_0(10.0.0.16): timestamp=02-19 13:28:14.504, src=172.31.0.66, latency=0.213,
jitter=0.014, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_
prio=0, last_sla_change=02-19 10:51:08.667
Remote Health Check: passive_hc(1)
  Passive remote statistics of EDGE_T1(45):
EDGE_T1_1(172.31.0.65): timestamp=02-19 13:28:14.680, src=172.31.0.65, latency=120.247,
jitter=0.017, pktloss=0.000%, mos=4.338, SLA id=1(remote), rmt_ver=1, rmt_sla=out, rmt_
prio=0, last_sla_change=02-19 13:25:41.227
EDGE_T1_0(172.31.0.66): timestamp=02-19 13:28:14.504, src=172.31.0.66, latency=0.255,
jitter=0.031, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_
prio=0, last_sla_change=02-19 10:51:08.157
```

b. Ping PC-1:

The duplication takes place from Hub to Spoke-1.

```
test@server:~$ ping 10.0.3.2

64 bytes from 10.0.3.2: icmp_seq=51 ttl=62 time=120 ms
64 bytes from 10.0.3.2: icmp_seq=52 ttl=62 time=120 ms
64 bytes from 10.0.3.2: icmp_seq=53 ttl=62 time=120 ms
64 bytes from 10.0.3.2: icmp_seq=54 ttl=62 time=0.899 ms
64 bytes from 10.0.3.2: icmp_seq=54 ttl=62 time=0.931 ms (DUP!)
64 bytes from 10.0.3.2: icmp_seq=54 ttl=62 time=120 ms (DUP!)
64 bytes from 10.0.3.2: icmp_seq=55 ttl=62 time=0.820 ms
64 bytes from 10.0.3.2: icmp_seq=55 ttl=62 time=0.855 ms (DUP!)
64 bytes from 10.0.3.2: icmp_seq=55 ttl=62 time=120 ms (DUP!)
64 bytes from 10.0.3.2: icmp_seq=56 ttl=62 time=0.793 ms
64 bytes from 10.0.3.2: icmp_seq=56 ttl=62 time=0.827 ms (DUP!)
64 bytes from 10.0.3.2: icmp_seq=56 ttl=62 time=120 ms (DUP!)
```

Because one tunnel is out-of-SLA, the ping test changes from single echo replies (as indicated by the sequence number increasing by one and the latency remaining at 120ms) to three echo replies per sequence and only one with 120ms latency, indicating that all tunnels are used at once. Note that the duplication was configured to send four duplicate packets, but only two duplicates are received as a result of only three eligible tunnels.

7. Increase the latency to 220ms on overlay H1_T11 of Spoke-2, then confirm that corresponding child tunnel on the Hub is out-of-SLA. Also confirm that duplication occurs from the Hub to Spoke-2.

a. Check the health status on the HUB:

```
# diagnose sys sdwan health-check  remote
Remote Health Check: passive_hc(3)
  Passive remote statistics of EDGE_T3(47):
EDGE_T3_0(10.0.0.15): timestamp=02-19 13:40:38.693, src=172.31.0.65, latency=0.157,
jitter=0.008, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_
prio=0, last_sla_change=02-19 10:23:45.877
EDGE_T3_1(10.0.0.17): timestamp=02-19 13:40:38.766, src=172.31.0.66, latency=0.145,
jitter=0.017, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_
```

```
prio=0, last_sla_change=02-19 10:51:17.377
Remote Health Check: passive_hc(2)
  Passive remote statistics of EDGE_T2(46):
EDGE_T2_1(10.0.0.14): timestamp=02-19 13:40:38.692, src=172.31.0.65, latency=0.220,
jitter=0.006, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_
prio=0, last_sla_change=02-19 10:21:44.967
EDGE_T2_0(10.0.0.16): timestamp=02-19 13:40:38.766, src=172.31.0.66, latency=0.219,
jitter=0.012, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_
prio=0, last_sla_change=02-19 10:51:08.667
Remote Health Check: passive_hc(1)
  Passive remote statistics of EDGE_T1(45):
EDGE_T1_1(172.31.0.65): timestamp=02-19 13:40:38.692, src=172.31.0.65, latency=120.245,
jitter=0.010, pktloss=0.000%, mos=4.338, SLA id=1(remote), rmt_ver=1, rmt_sla=out, rmt_
prio=0, last_sla_change=02-19 13:25:41.227
EDGE_T1_0(172.31.0.66): timestamp=02-19 13:40:38.766, src=172.31.0.66, latency=220.242,
jitter=0.018, pktloss=0.000%, mos=4.103, SLA id=1(remote), rmt_ver=1, rmt_sla=out, rmt_
prio=0, last_sla_change=02-19 13:39:12.427
```

b. Ping PC-2:

The duplication takes place from Hub to Spoke-2.

```
test@server:~$ ping 10.0.4.2

64 bytes from 10.0.4.2: icmp_seq=103 ttl=62 time=220 ms
64 bytes from 10.0.4.2: icmp_seq=104 ttl=62 time=220 ms
64 bytes from 10.0.4.2: icmp_seq=105 ttl=62 time=220 ms
64 bytes from 10.0.4.2: icmp_seq=106 ttl=62 time=0.735 ms
64 bytes from 10.0.4.2: icmp_seq=106 ttl=62 time=0.742 ms (DUP!)
64 bytes from 10.0.4.2: icmp_seq=106 ttl=62 time=220 ms (DUP!)
64 bytes from 10.0.4.2: icmp_seq=107 ttl=62 time=0.721 ms
64 bytes from 10.0.4.2: icmp_seq=107 ttl=62 time=0.729 ms (DUP!)
64 bytes from 10.0.4.2: icmp_seq=107 ttl=62 time=220 ms (DUP!)
64 bytes from 10.0.4.2: icmp_seq=108 ttl=62 time=0.719 ms
64 bytes from 10.0.4.2: icmp_seq=108 ttl=62 time=0.731 ms (DUP!)
64 bytes from 10.0.4.2: icmp_seq=108 ttl=62 time=220 ms (DUP!)
```

8. Remove the latency impairment and see that the duplication stops.

a. Check the health status on the HUB:

```
# diagnose sys sdwan health-check  remote
Remote Health Check: passive_hc(3)
  Passive remote statistics of EDGE_T3(47):
EDGE_T3_0(10.0.0.15): timestamp=02-19 13:44:03.213, src=172.31.0.65, latency=0.156,
jitter=0.013, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_
prio=0, last_sla_change=02-19 10:23:45.877
EDGE_T3_1(10.0.0.17): timestamp=02-19 13:44:03.307, src=172.31.0.66, latency=0.151,
jitter=0.022, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_
prio=0, last_sla_change=02-19 10:51:17.377
Remote Health Check: passive_hc(2)
  Passive remote statistics of EDGE_T2(46):
EDGE_T2_1(10.0.0.14): timestamp=02-19 13:44:03.213, src=172.31.0.65, latency=0.211,
jitter=0.008, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_
```

```
prio=0, last_sla_change=02-19 10:21:44.967
EDGE_T2_0(10.0.0.16): timestamp=02-19 13:44:03.307, src=172.31.0.66, latency=0.224,
jitter=0.016, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_
prio=0, last_sla_change=02-19 10:51:08.667
Remote Health Check: passive_hc(1)
  Passive remote statistics of EDGE_T1(45):
EDGE_T1_1(172.31.0.65): timestamp=02-19 13:44:03.213, src=172.31.0.65, latency=0.269,
jitter=0.028, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_
prio=0, last_sla_change=02-19 13:43:36.967
EDGE_T1_0(172.31.0.66): timestamp=02-19 13:44:03.307, src=172.31.0.66, latency=0.263,
jitter=0.036, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=1, rmt_sla=in, rmt_
prio=0, last_sla_change=02-19 13:43:35.947
```

b. Ping PC-1:
Note where the duplication stops.

```
test@server:$ ping 10.0.3.2

64 bytes from 10.0.3.2: icmp_seq=150 ttl=62 time=0.579 ms
64 bytes from 10.0.3.2: icmp_seq=150 ttl=62 time=0.640 ms (DUP!)
64 bytes from 10.0.3.2: icmp_seq=150 ttl=62 time=0.643 ms (DUP!)
64 bytes from 10.0.3.2: icmp_seq=151 ttl=62 time=0.531 ms
64 bytes from 10.0.3.2: icmp_seq=152 ttl=62 time=0.578 ms
64 bytes from 10.0.3.2: icmp_seq=153 ttl=62 time=0.572 ms
64 bytes from 10.0.3.2: icmp_seq=154 ttl=62 time=0.557 ms
```

c. Ping PC-2:
Note where the duplication stops.

```
test@server:~$ ping 10.0.4.2

64 bytes from 10.0.4.2: icmp_seq=200 ttl=62 time=0.730 ms
64 bytes from 10.0.4.2: icmp_seq=200 ttl=62 time=0.767 ms (DUP!)
64 bytes from 10.0.4.2: icmp_seq=200 ttl=62 time=0.769 ms (DUP!)
64 bytes from 10.0.4.2: icmp_seq=201 ttl=62 time=0.672 ms
64 bytes from 10.0.4.2: icmp_seq=202 ttl=62 time=0.551 ms
64 bytes from 10.0.4.2: icmp_seq=203 ttl=62 time=0.543 ms
64 bytes from 10.0.4.2: icmp_seq=204 ttl=62 time=0.568 ms
```

Performance SLA

This section includes information about performance SLA related new features:

- SD-WAN speed test scheduling improvements on page 188

SD-WAN speed test scheduling improvements

Speed-test scheduling is now coordinated automatically between the FortiGate appliance and the backend Fortinet servers, removing the need to configure retry settings. Using speed-test results for traffic shaping has also been simplified, with all related settings moved directly into the interface configuration. Speed-test traffic now ignores interface traffic shaping but is affected by values for `update-inbandwidth-maximum` and `update-outbandwidth-maximum` settings when configured within that speed-test's settings. These changes reduce administrative overhead and make speed-test-driven shaping more consistent and easier to manage.

Speed-test scheduling and usage improvement

Speed-test scheduling and usage has been improved in two ways:

- Speed-test settings

  A coordination mechanism between the FortiGate appliance and the backend server has been introduced to streamline speed-test operations. With this enhancement, scheduled speed tests no longer require explicit configuration of `retry` or `retry-pause` parameters, as these behaviors are now managed automatically by the coordination logic. Speed-test traffic no longer uses dedicated shaping settings, and now respects any configured maximum interface bandwidth values for that speed-test-schedule.
- Speed-test result usage

  The configuration required to use speed-test results for interface traffic shaping has been simplified, and settings are now located in the interface configuration.

Affected CLI commands

Speed test settings

This section covers the existing, removed, and new commands for speed test settings.

Existing commands

These commands control maximum and minimum downloading bandwidth values to be considered effective, in Kbps (0 - 16776000). The update-inbandwidth-maximum and update-outbandwidth-maximum settings also function as traffic shaping for speed-tests. When minimums are not met, or maximums are exceeded, the minimum or maximum will be used instead of the measured value.

```
config system speed-test-schedule
    edit <interface>
        set update-inbandwidth-maximum <integer>
        set update-inbandwidth-minimum <integer>
        set update-outbandwidth-maximum <integer>
        set update-outbandwidth-minimum <integer>
```

Removed commands

The `retries` and `retry-pause` commands have been removed and the functionality has been migrated to the FortiGate and Server coordination mechanism.

```
config system speed-test-schedule
    edit <interface>
        set retries <int>
        set retry-pause <int>
```

The `update-inbandwidth` and `update-outbandwidth` commands are no longer used to apply shaping to speed-tests. The existing commands `update-inbandwidth-maximum` and `update-outbandwidth-maximum` are used to apply shaping during speed-tests.

```
config system speed-test-schedule
    edit <interface>
        set update-inbandwidth {enable | disable}
        set update-outbandwidth {enable | disable}
```

The `update-shaper` command has been removed, and the functionality has been replaced by the new command `set inbandwidth-source measured`. See New commands on page 189 for details.

```
config system speed-test-schedule
    edit <interface>
        set update-shaper
```

New commands

The `update-bandwidth-limit-unit` option controls how the values of `update-inbandwidth-maximum/minimum` and `update-outbandwidth-maximum/minimum` are interpreted. By default, these values are treated as kilobits per second (kbps). When set to `percentage`, the values are interpreted as percentages of the interface's configured inbound or outbound bandwidth.

```
config system speed-test-schedule
    edit <interface>
        set update-bandwidth-limit-unit {kbps* | percentage}
```

Speed test results used for traffic shaping

This section covers the existing, removed, and new commands for speed test results used for traffic shaping.

Existing commands

```
config system interface
    set inbandwidth <integer and manually configured>
    set outbandwidth <integer and manually configured>
    set measured-upstream-bandwidth <integer and populated by speed-test results>
    set measured-downstream-bandwidth <integer and populated by speed-test results>
```

Removed commands

The function controlled by the optional `update-int-shaping` field in the `execute speed-test` command is now controlled by the new settings `set inbandwidth-source measured` and `set outbandwidth-source measured`. If a speed-test is manually executed, these commands will control whether the speed-test results are applied to the interface.

```
execute speed-test ... <update-intf-shaping> ...
```

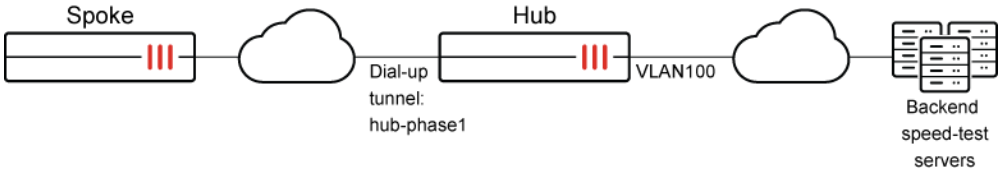
New commands

```
config system interface
    set inbandwidth-source {default* | measured}
    set inbandwidth-source {default* | measured}
```

These commands determine which value from the Existing commands on page 189 takes effect, and replace the following FortiOS 7.6 command:

```
config system speed-test-schedule
    set update-interface-shaping {enable | disable}
```

Example



This section contains the configuration settings for the Hub and Spoke FortiGates to demonstrate two uses of the speed-test:

1. A speed-test is scheduled and run from the Hub to Spoke just before the business opens at 8 am. These results are applied to the traffic shaping on the spoke dial-up tunnel.
2. A second speed-test is run between the Hub and the internet, and the result is applied to the corresponding Hub interface (vlan100 in this example).

Key settings on Hub:

Use speed-test measured bandwidth as inbandwidth:

```
Hub# show sys interface vlan100
config system interface
    edit "vlan100"
        set vdom "root"
        set ip 172.16.206.1 255.255.255.0
        set allowaccess ping https ssh snmp http
        set egress-shaping-profile "profile_1"
        set ingress-shaping-profile "profile_1"
        set inbandwidth-source measured
        set outbandwidth-source measured
        ...
        set vlanid 100
    next
end
```

Allow `speed-test` protocol on overlay for IPsec speed test. Apply egress-shaping-profile. Use speed-test measured bandwidth as outbandwidth:

```
Hub# show sys interface hub-phase1
config system interface
    edit "hub-phase1"
        set vdom "root"
        set ip 10.10.15.253 255.255.255.255
        set allowaccess ping https ssh snmp http telnet radius-acct probe-response fabric ftm
speed-test
        set type tunnel
        set egress-shaping-profile "profile_1"
        set outbandwidth-source measured
        set remote-ip 10.10.15.254 255.255.255.0
        ...
        set interface "port1"
    next
end
```

Allow `speed-test` protocol on underlay for IPsec speed test.

```
Hub# show sys interface  port1
config system interface
    edit "port1"
        set vdom "root"
        set ip 172.16.200.1 255.255.255.0
        set allowaccess ping https ssh snmp http telnet radius-acct probe-response fabric ftm
speed-test
        ...
    next
end
```

Enable and configure the speed test:

```
Hub# show sys speed-test-schedule
config system speed-test-schedule
    edit "hub-phase1"
        set mode TCP
        set schedules "1"
        set dynamic-server enable //Enable IPsec speed test
        set update-bandwidth-limit-unit value //Set the update bandwidth limit by values in kbps
        set update-inbandwidth-maximum 1000000 //Maximum downloading bandwidth (kbps) of the
interface's inbandwidth to be used in shaping
        set update-inbandwidth-minimum 500000 //Minimum downloading bandwidth (kbps) of the
interface's inbandwidth to be used in shaping
        set update-outbandwidth-maximum 1000000 //Maximum uploading bandwidth (kbps) of the
interface's outbandwidth to be used in shaping
        set update-outbandwidth-minimum 500000 //Minimum uploading bandwidth (kbps) of the
interface's outbandwidth to be used in shaping
    next
    edit "vlan100"
        set mode TCP
        set schedules "2"
```

```
        set legacy-server-mode disable //Utilize the coordination mechanism between Fortigate and
backend server to do internet speed-test
        set update-bandwidth-limit-unit value //Set the update bandwidth limits by values in kbps
        set update-inbandwidth-maximum 1000000 //Maximum downloading bandwidth (kbps) of the
interface's inbandwidth to be used in shaping
        set update-inbandwidth-minimum 500000 //Minimum downloading bandwidth (kbps) of the
interface's inbandwidth to be used in shaping
        set update-outbandwidth-maximum 1000000 //Maximum uploading bandwidth (kbps) of the
interface's outbandwidth to be used in shaping
        set update-outbandwidth-minimum 500000 //Minimum uploading bandwidth (kbps) of the
interface's outbandwidth to be used in shaping
    next
end
```

The first schedule:

```
Hub# show firewall schedule recurring 1
config firewall schedule recurring
    edit "1"
        set start 07:43
        set day sunday monday tuesday wednesday thursday friday saturday
    next
end
```

The second schedule:

```
Hub# show firewall schedule recurring 2
config firewall schedule recurring
    edit "2"
        set start 07:44
        set day sunday monday tuesday wednesday thursday friday saturday
    next
end
```

Key settings on Spoke:

Enable speed-test server for IPsec speed test:

```
Spoke# show sys global
config system global
    ...
    set speedtest-server enable
    ...
end
```

Allow `speed-test` protocol on overlay for IPsec speed test:

```
Spoke# show sys interface  spoke11-p1
config system interface
    edit "spoke11-p1"
        set vdom "root"
        set allowaccess ping https ssh snmp http telnet radius-acct probe-response fabric ftm
```

```
speed-test
        set type tunnel
        ...
        set interface "port1"
    next
end
```

Allow `speed-test` protocol on underlay for IPsec speed test.

```
Spoke# show sys interface  port1
config system interface
    edit "port1"
        set vdom "root"
        set ip 172.16.200.2 255.255.255.0
        set allowaccess ping https ssh snmp http telnet radius-acct probe-response fabric ftm
speed-test
        ...
    next
end
```

Example 1: Hub to Spoke speed test

View the speed-test output by running the following debug command prior to the start of the scheduled test:

```
Hub # diagnose debug application speedtestd -1
```

Recurring firewall schedule 1 begins:

```
Hub# fcron_speedtest_arm_sched()-405: Speed test (0x5561afd510) for hub-phase1 will run in 48 s
fcron_speedtest_arm_sched()-405: Speed test (0x5561afd510) for hub-phase1 will run in 86400 s
fcron_speedtest_ipsec_send_request()-574: root: hub-phase1(hub-phase1_0) try=0 token
request=0.0.0.0:0 -> 10.10.15.1:5200, test= 172.16.200.1:0 -> 172.16.200.2:5201
fcron_sptest_ipsec_on_start()-528: root: (00790005) hub-phase1(hub-phase1_0) server notify start
token=42c4abc4-1292-51f1-07b1-9054c39ee49f
__fork_run_ipsec_test()-987: [15270] Run test 00790005 for 'hub-phase1'(port1) to server
172.16.200.2:5201 (tunnel:hub-phase1_0)
[speedtest(15270)] start uploading test.
[speedtest(15270)] Connecting to host 172.16.200.2, port 5201
[speedtest(15270)] [ 28] local 172.16.200.1 port 14251 connected to 172.16.200.2 port 5201
[speedtest(15270)] [ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[speedtest(15270)] [ 28]   0.00-1.01   sec  83.0 MBytes   686 Mbits/sec    0    134 KBytes
…
[speedtest(15270)] client(recver): bytes_recv=425522920, bytes_sent=426768480, sender_time=5.000,
recver_time=5.001
[speedtest(15270)] client(recver): down_speed:  681 Mbits/sec
[speedtest(15270)]
[speedtest(15270)] speed test Done.
fcron_speedtest_notify_func()-1570: Speed test pid=15270 done
fcron_speedtest_on_test_finish()-1530: test 0x00790005 for 'hub-phase1' succeed with up=692880,
down=680722
fcron_speedtest_save_results()-1428: Write logs to disk: succ=1, fail=0
fcron_speedtest_sync_results()-1456: Sync cached results to secondary devices.
```

The output from the speed-test debug indicates the speed-test completed with an upload throughput of 692880 kbps. To verify that this value has been applied to the corresponding interface, review the VPN tunnel as follows:

```
Hub# diagnose vpn tunnel list
------------------------------------------------------
name=hub-phase1_0 ver=2 serial=9 172.16.200.1:0->172.16.200.2:0 nexthop=172.16.200.2 tun_
id=10.10.15.1 tun_id6=2000:10:10:15::1 status=up dst_mtu=1500 weight=1 country=ZZ
bound_if=11 real_if=11 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none options[0x22a8]=npu
rgwy-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=10
...
egress traffic control:
        bandwidth=692880(kbps) lock_hit=0 default_class=2 n_active_class=3
```

Example 2: Hub to Fortinet server internet speed-test

The speed-test output can be seen by running the following debug command prior to the start of the scheduled test:

```
Hub # diagnose debug application speedtest -1
```

Recurring firewall schedule 2 begins:

```
Hub# diagnose vpn tunnel list
------------------------------------------------------
name=hub-phase1_0 ver=2 serial=9 172.16.200.1:0->172.16.200.2:0 nexthop=172.16.200.2 tun_
id=10.10.15.1 tun_id6=2000:10:10:15::1 status=up dst_mtu=1500 weight=1 country=ZZ
bound_if=11 real_if=11 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none options[0x22a8]=npu
rgwy-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=10
...
egress traffic control:
        bandwidth=692880(kbps) lock_hit=0 default_class=2 n_active_class=3


Hub# fcron_speedtest_arm_sched()-405: Speed test (0x5561aa61f0) for vlan100 will run in 33 s
...
[speedtest(15619)] speed test Done.
fcron_speedtest_notify_func()-1570: Speed test pid=15619 done
fcron_speedtest_on_test_finish()-1530: test 0x00630000 for 'vlan100' succeed with up=638101,
down=930393
fcron_sptest_cloud_report_result()-715: test(0x00630000): if=vlan100, server=FTNT_CA_Burnaby
(154.52.1.124:5213), uuid=6a82a804-45b1-477f-b94b-f4a21e17596 report result=0, up=638101,
dw=930393
fcron_sptest_cloud_result_reply()-663: test(0x00630000): if=vlan100, uuid=6a82a804-45b1-477f-b94b-
f4a21e17596 report result succeed (status=2).
```

The output from the speed-test debug indicates the speed-test completed with an upload throughput of 638101 kbps and a download throughput of 930393 kbps. To verify that this value has been applied to the corresponding interface, review the VLAN interface as follows:

```
Hub# diagnose netlink interface list vlan100
if=vlan100 alias= family=00 type=1 index=99 mtu=1500 link=0 master=0
ref=28 state=start present fw_flags=10010400 flags=up broadcast run multicast
```

```
Qdisc=noqueue hw_addr=e0:23:ff:3e:7f:eb broadcast_addr=ff:ff:ff:ff:ff:ff
ingress traffic control:
        bandwidth=930393(kbps) lock_hit=0 default_class=2 n_active_class=3
...
egress traffic control:
        bandwidth=638101(kbps) lock_hit=0 default_class=2 n_active_class=3
```

Policy and objects

This section includes information about policy and object related new features:

- Policies on page 196
- Objects on page 199

Policies

This section includes information about policy related new features:

- Upstream SSL/TLS support for virtual load-balancing servers on page 196

Upstream SSL/TLS support for virtual load-balancing servers

FortiOS now supports SSL/TLS connectivity for upstream HTTPS virtual load-balancing servers. When enabled, FortiGate converts an incoming HTTP connection to HTTPS between itself and the real server. This can apply to cases where resource-constrained devices within an internal, trusted network connect through a VIP to upstream real servers that are in untrusted networks, thereby securing the connection in the untrusted network.

The `config firewall vip` command includes a new option when `server-type` is set to `http`:

```
config firewall vip
    edit <id>
        set type server-load-balance
        set server-type http
        ...
        set ssl-upstream {enable | disable}
        set ssl-http-strip-secure-cookies {enable | disable}
        set ssl-http-location-conversion {enable | disable}
        set ssl-server-client-certificate <string>
        ...
    next
end
```

Option	Description
`ssl-upstream {enable | disable}`	Enable/disable SSL encryption between the FortiGate and the upstream server (default = disable). Available when `server-type` is set to `http`.

Option	Description
`ssl-http-strip-secure-cookies {enable | disable}`	Enable/disable removal of HTTPS-only directives in the reply's Set-Cookie HTTP header fields (default = disable).
`ssl-http-location-conversion {enable | disable}`	Enable to convert HTTP/HTTPS in the reply's Location HTTP header field (default = disable).
`ssl-server-client-certificate`	Name of the client certificate presented to realserver during SSL/TLS handshake if requested.

When the `server-type` is `http` and `ssl-upstream` is set to `enable`:

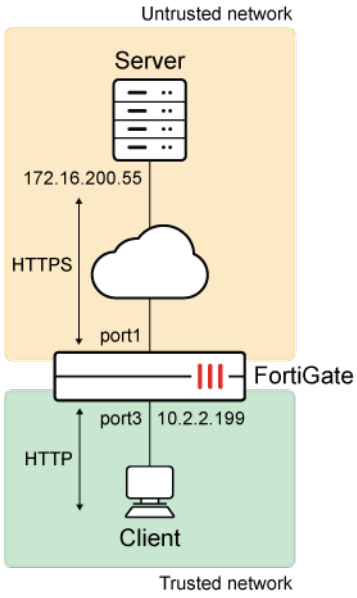
- The following parameters are supported:

  `ssl-certificate`, `ssl-dh-bits`, `ssl-accept-ffdhe-groups`, `ssl-send-empty-frags`, `h2-support`, `h3-support`, `ssl-server-session-state-type`, `ssl-server-algorithm`, `ssl-server-min-version`, `ssl-server-max-version`, `ssl-server-renegotiation`, and `verify-cert`.
- The `client` option in `ssl-server-algorithm/min-version/max-version` is hidden.
- The `ssl-certificate` and `ssl-accept-ffdhe-groups` are hidden.
- If an existing `client` option is enabled, it is set to the default value following the same logic as `ssl-algorithm/min-version/max-version`, with an info message printed.
- If a conflict occurs between `min-version` and `max-version`, an error message is printed.
- FortiGate removes HTTPS-only response headers, such as HSTS and HPKP, by default

A new option, `ssl-http-strip-secure-cookies`, allows HTTPS-only attributes (including `Secure`, `Partitioned`, and `SameSite=None`) to be stripped from `Set-Cookie` headers before forwarding responses to HTTP clients.

The `ssl-http-location-conversion` option is now supported and visible when SSL upstream is enabled. When enabled, HTTPS URLs in the `Location` response header are rewritten to HTTP.

Example



In this example, a resource-constrained client inside a trusted network is trying to reach a real server over a virtual server load-balancing VIP. This connection is established over HTTP between the client and the FortiGate, but the upstream connection to the real server is converted to HTTPS.

To enable VIP server with ssl-upstream:

1. Create a VIP with `server-type` set to `http`:

```
config firewall vip
    edit test_vip4
        set type server-load-balance
        set server-type http
        set extip 10.2.2.199
        set extintf port3
        set extport 80
        config realservers
            edit 1
                set ip 172.16.200.55
                set port 443
                set max-connections 100
                set verify-cert disable
            next
        end
    next
end
```

2. Enable SSL parameters, including `ssl-upstream`:

```
config firewall vip
    edit test_vip4
        set http-multiplex enable
        set ssl-upstream enable
        set ssl-certificate Fortinet_Factory
        set ssl-server-algorithm high
        set ssl-server-min-version tls-1.1
        set ssl-server-min-version tls-1.3
    next
end
```

3. Reference the configured VIP in a firewall policy:

```
config firewall policy
    edit 0
        set srcintf "port3"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "test_vip4"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
```

```
        set nat enable
    next
end
```

4. Confirm HTTP access to the VIP server from the client.

   In this example, cURL is used to confirm FortiGate accepts the HTTP request and securely forwards it to the VIP server (10.2.2.199) over HTTPS. A successful test returns an `HTTP/1.1 200 OK` response; failure may result in connection errors or HTTP 404.

```
fosqa@pc01:~$ curl -vk http://10.2.2.199:80
   *   Trying 10.2.2.199:80...
   * TCP_NODELAY set
   * Connected to 10.2.2.199 (10.2.2.199) port 80 (#0)
   > GET / HTTP/1.1
   > Host: 10.2.2.199
   > User-Agent: curl/7.68.0
   > Accept: */*
   >
   * Mark bundle as not supporting multi-use
   < HTTP/1.1 200 OK
   < Date: Tue, 19 Aug 2025 21:54:56 GMT
   < Server: Apache/2.4.41 (Ubuntu)
```

Objects

This section includes information about object related new features:

- Custom tags for addresses, policies, and dynamic tag address groups on page 199

Custom tags for addresses, policies, and dynamic tag address groups

This information is also available in the FortiOS 8.0 Administration Guide:

- Custom tags

Custom tags can be created and applied to various address types, address groups, and policy types. This helps administrators organize their addresses and policies, quickly visualize the category, and easily filter based on the assigned tags when many addresses and policies are configured.

The following objects can have custom tags assigned:

- Firewall address and address groups (IPv4 and IPv6)
- Proxy address and address groups
- Multicast addresses (IPv4 and IPv6)
- Firewall policy

- NGFW security policy
- Multicast policy
- DoS policy
- Local-in policy
- Central SNAT mapping
- Proxy policy

When address groups use custom tags as group members, they are called dynamic tag address groups. You can define more address objects for the tag associated with the address group for the firewall policy to automatically use the new address objects. No changes to the firewall policy are required.

Custom tags have tablesize limits. Refer to the Max Values table for exact size limits.

CLI syntax

New options are available to create custom tags:

```
config firewall custom-tag
    edit <tag name>
        set abbreviation <string>
        set comment <string>
        set color <integer>
    next
end
```

Command	Descriptions
`abbreviation <string>`	Custom tag name abbreviation. Maximum of four uppercase letters and numbers.
`comment <string>`	Comment.
`color <integer>`	Color of the custom tag on the GUI (0 to 32, default = 1).

To apply the tag to an address or policy:

```
config firewall address
    edit "taggedAddress"
        set custom-tags <tag> [tag]
    next
end
```

```
config firewall policy
    edit <ID>
        set custom-tags <tag> [tags]
    next
end
```

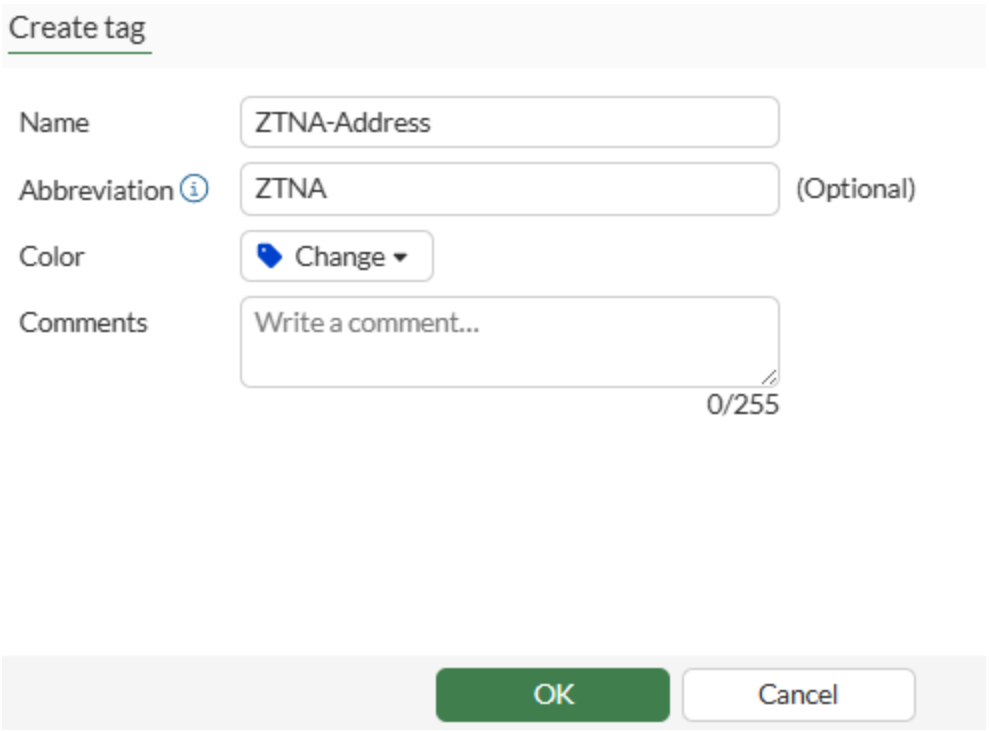
New options are available to create dynamic tag address groups with tags as group members:

```
config firewall addrgrp
    edit <name>
        set type {default | folder | dynamic-tag}
        set custom-tags <string>
    next
end
```

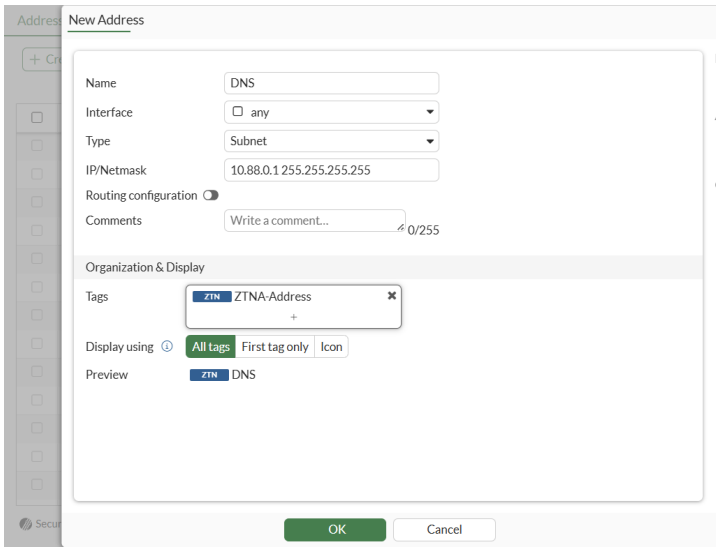
Command	Descriptions
`type {default | folder | dynamic-tag}`	Type of address group: • `default`: Default address group type (address may belong to multiple groups). • `folder`: Address folder group (members may not belong to any other group). • `dynamic-tag`: Dynamic tag group (members are addresses only if they have any one of the configured tags).
`custom-tags <string>`	Name of the custom tag. Available when `type` is set to `dynamic-tag`.

GUI

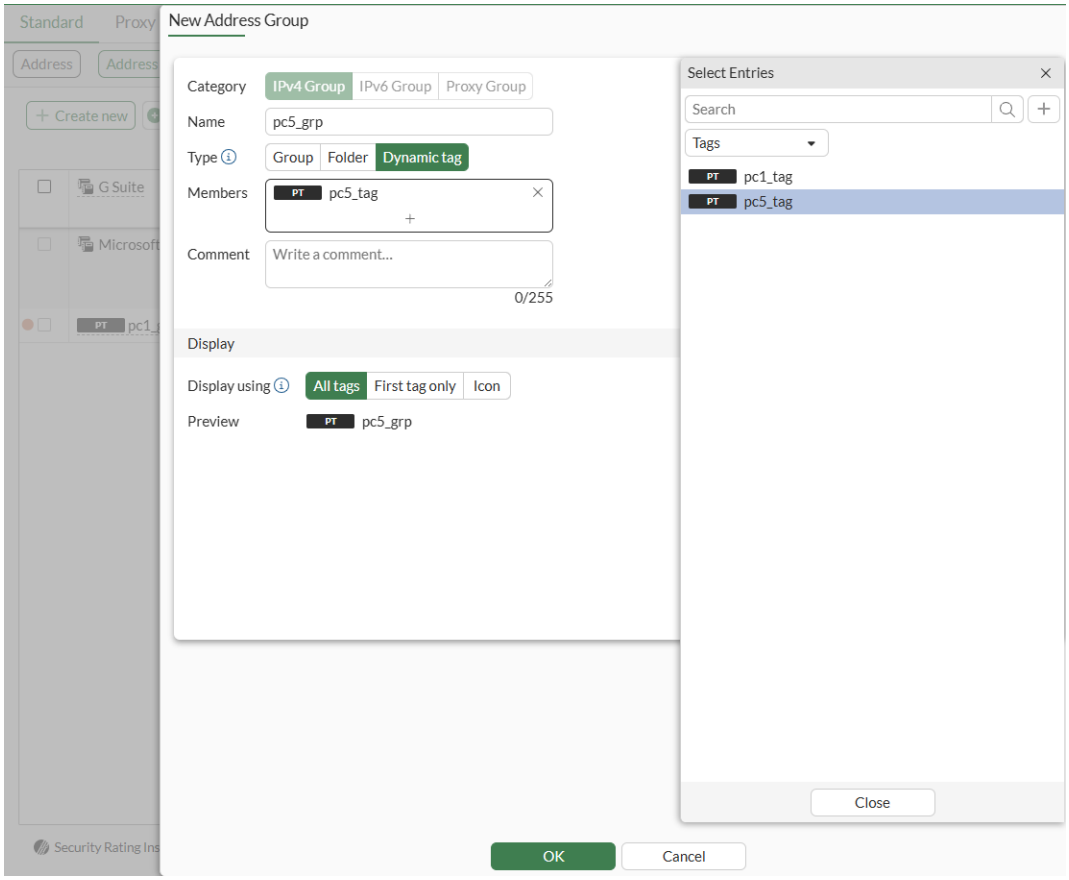
Custom tags can be configured from *Policy & Objects* > *Tags*.



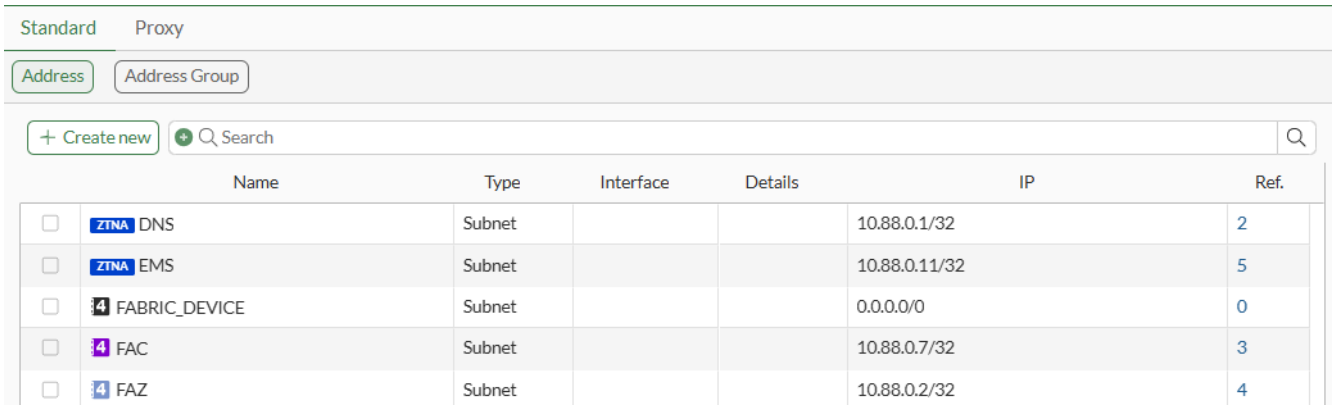
The abbreviation can be up to three uppercase letters and numbers.



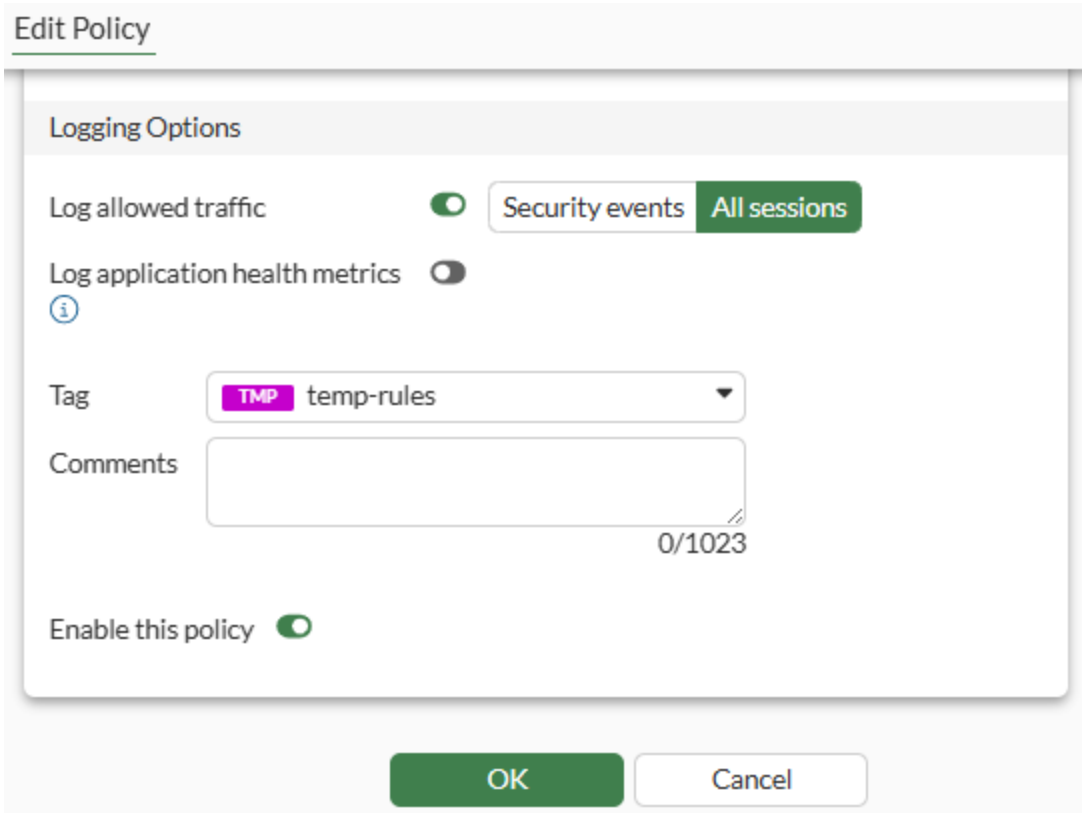
In address groups, set *Type* to *Dynamic* tag to select one or more tags:



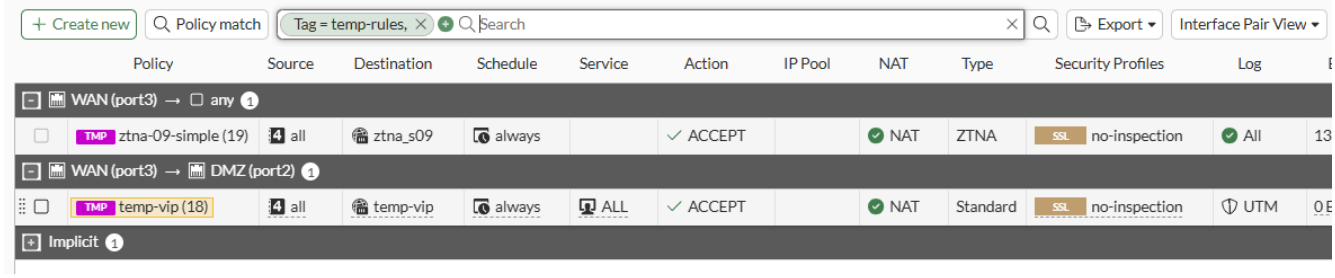
When *All tags* or *First tag only* is chosen as the display, the tag is shown as opposed to the default colored icon in the Address objects page for that particular object.



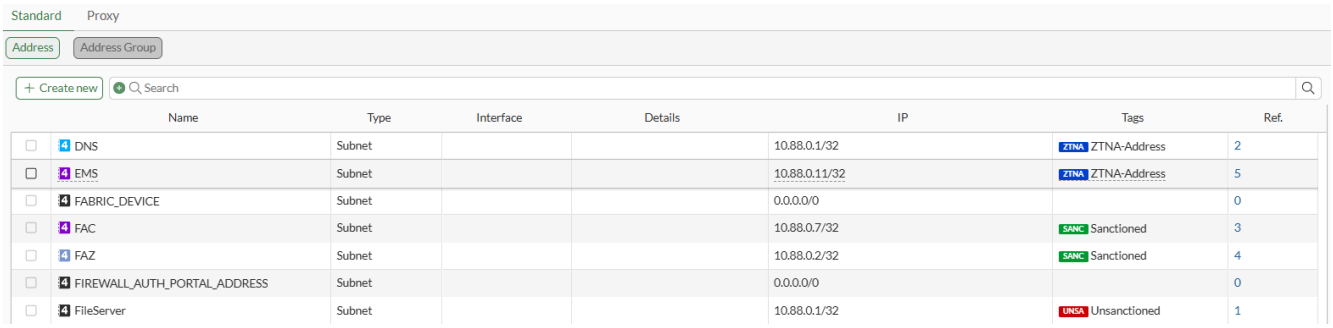
Policy Tagging is not enabled by default and must be enabled in *System* > *Feature Visibility*. Once enabled, a tag or dynamic address group of tag(s) can be applied to the policy.



Tags are visible in the Policy list page and can be filtered in the Search bar.



Display changes slightly when multiple object tags are used. Use the *Tags* column to view the assigned tags.



Example

This example describes how to create two custom tags, associate the custom tags with address objects, and configure dynamic tag address groups. Dynamic tag address groups use tag names as group members instead of object names. The firewall policy is configured to use the dynamic tag address groups to allow traffic from PC1 to PC5.

The following items are created:

- Two custom tags:
  - pc1_tag
  - pc5_tag
- Two address objects:
  - pc1 assigned the pc1_tag
  - pc5 assigned the pc5_tag
- Two dynamic tag address groups:
  - pc1_grp assigned the pc1_tag
  - pc5_grp assigned the pc5_tag

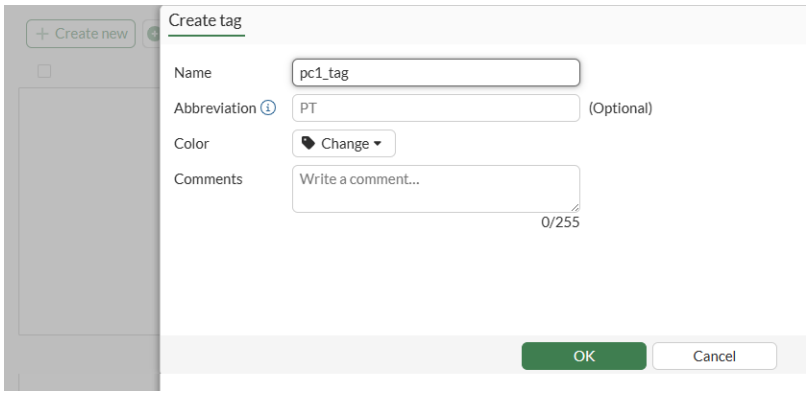
The firewall policy is configured to use the dynamic tag address groups (pc1_grp and pc5_grp) to allow traffic from PC1 to PC5.

When you add more address objects with tags, the tags are used to automatically update the dynamic tag address group used by the firewall policy. No changes are required on the associated firewall policy.

For example, you could create more address objects assigned the pc1_tag for the firewall policy to automatically use the address objects for the pc1_grp, which is configured to use address objects assigned the pc1_tag.

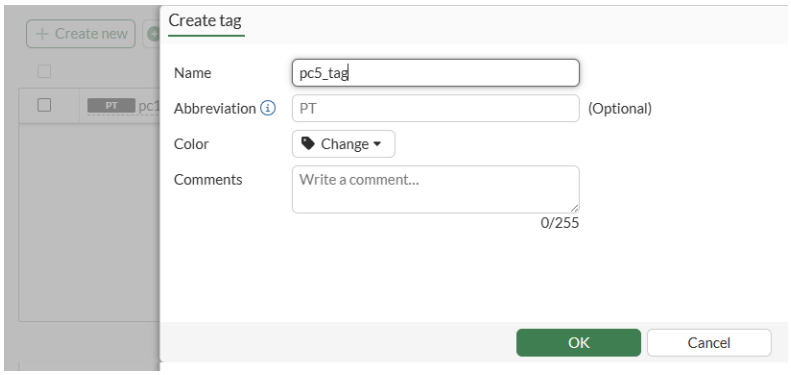
To create dynamic tag address groups in the GUI:

1. Create two custom tags:
   a. Go to *Policy & Objects* > *Tags*.
   b. Click *Create new*.
   c. Enter a *Name*, such as pc1_tag.



d. Click *OK*.

e. Repeat this procedure to create another custom tag named pc5_tag.



Two custom tags are available: pc1_tag and pc5.

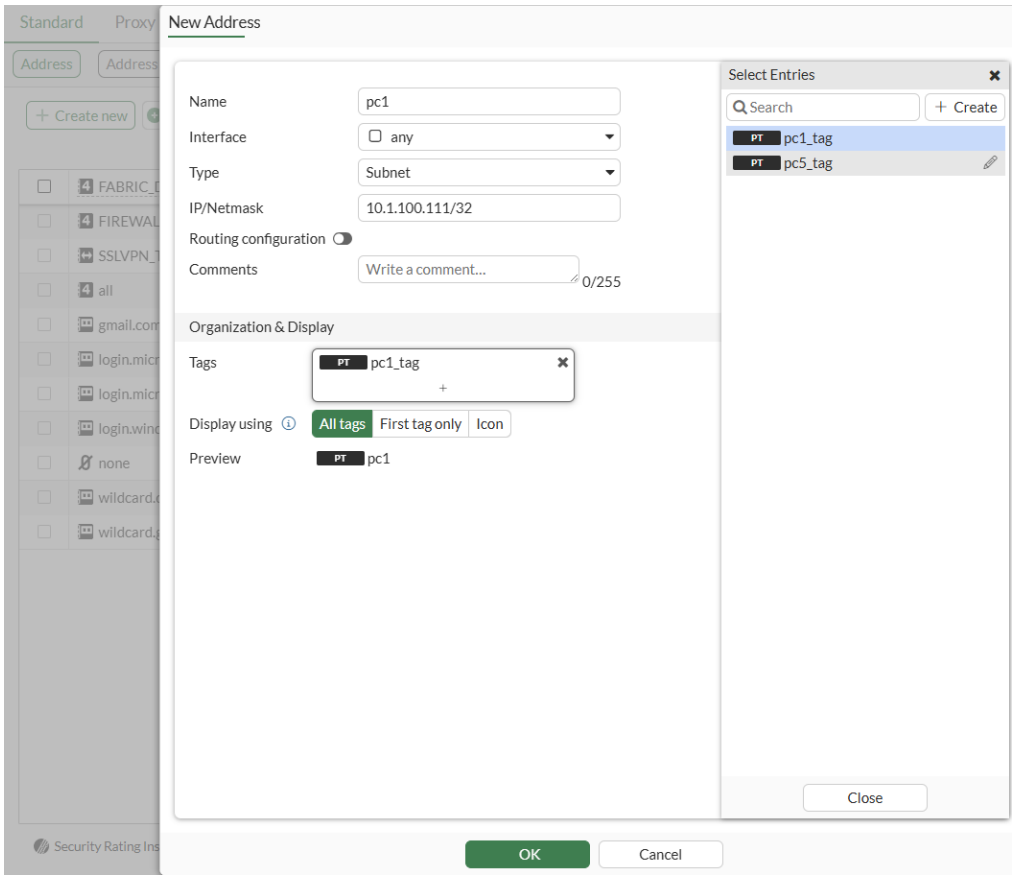
2. Create two address objects for PC1 and PC5 and assign to them the tags you created:

a. Go to *Policy & Objects* > *Addresses*.

b. On the *Address* tab, click *Create new*.

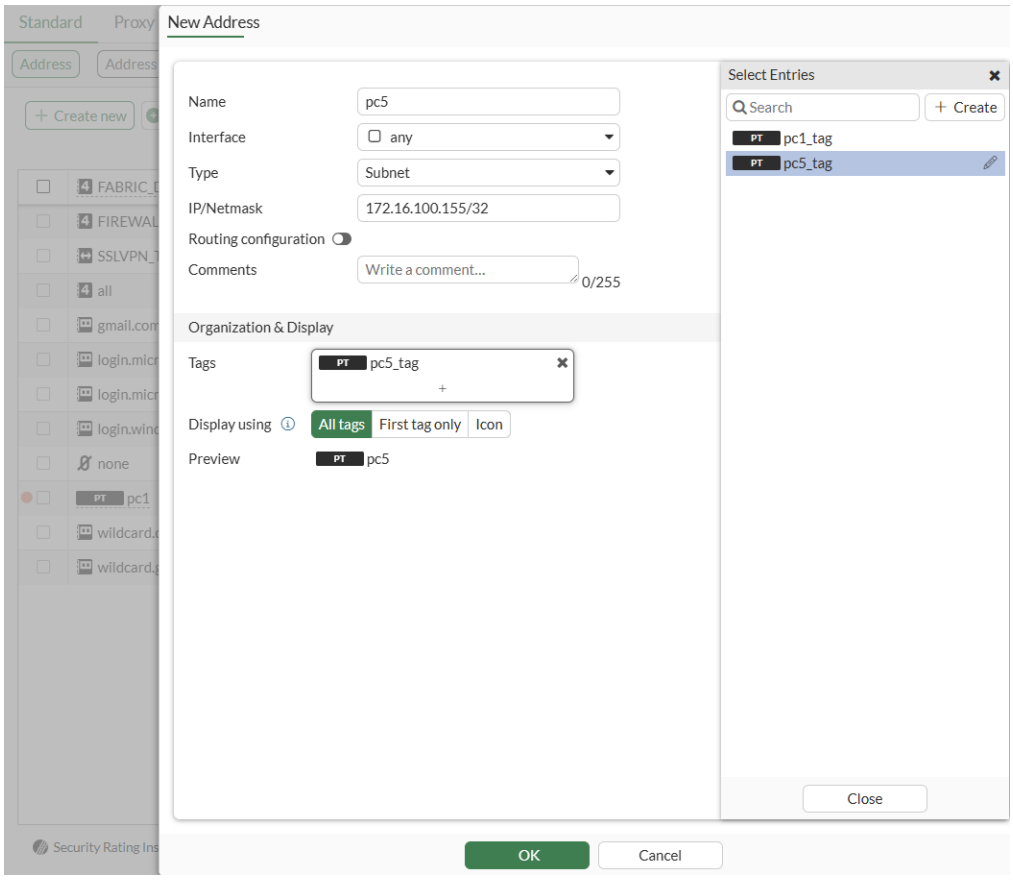
c. Enter a *Name*, such as pc1.

d. Beside *Tags*, click *+*, and select the pc1_tag you created.

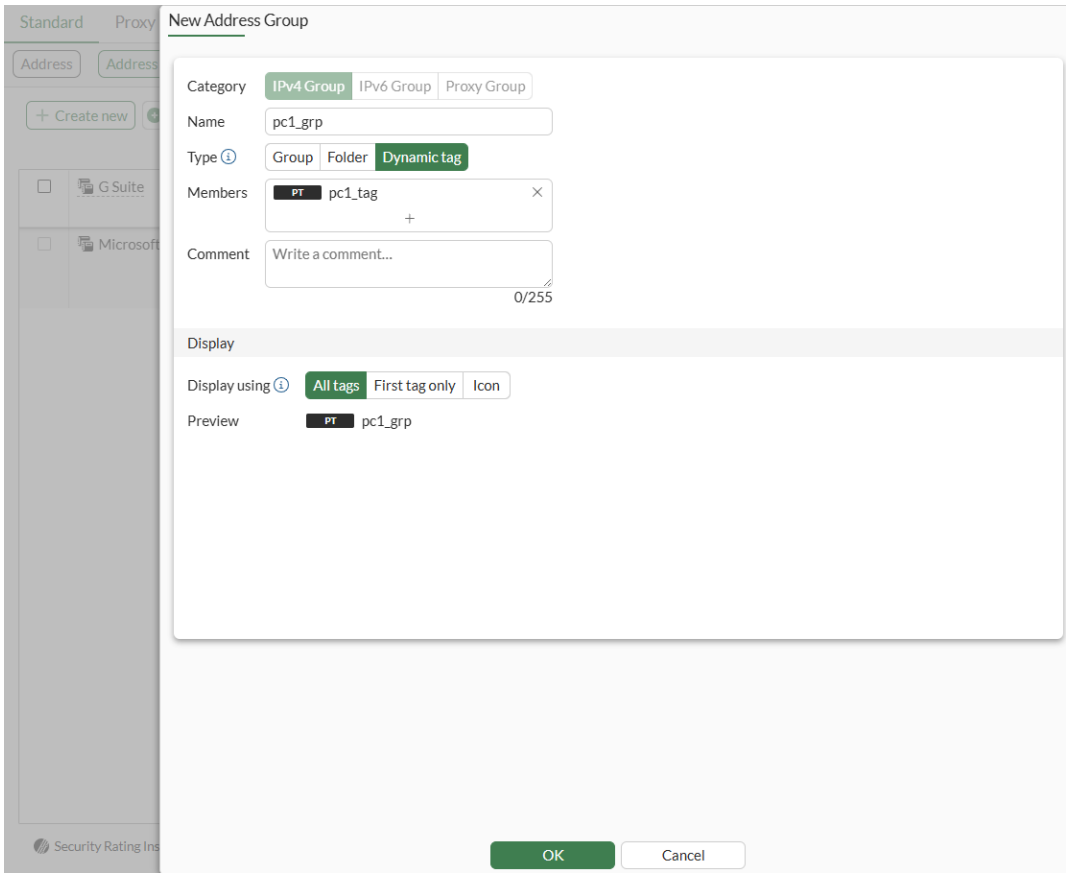


e. Click *OK*.

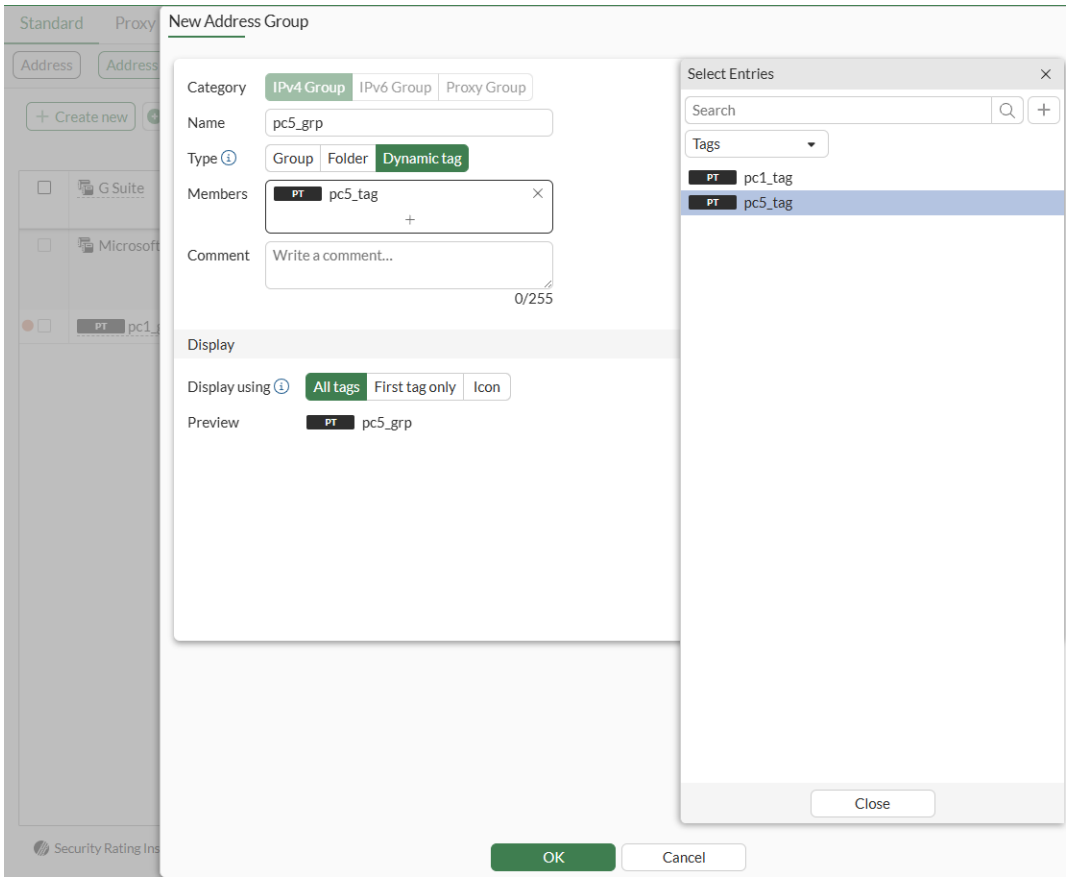
f. Repeat this procedure to create an address for pc5 and select the pc5_tag.



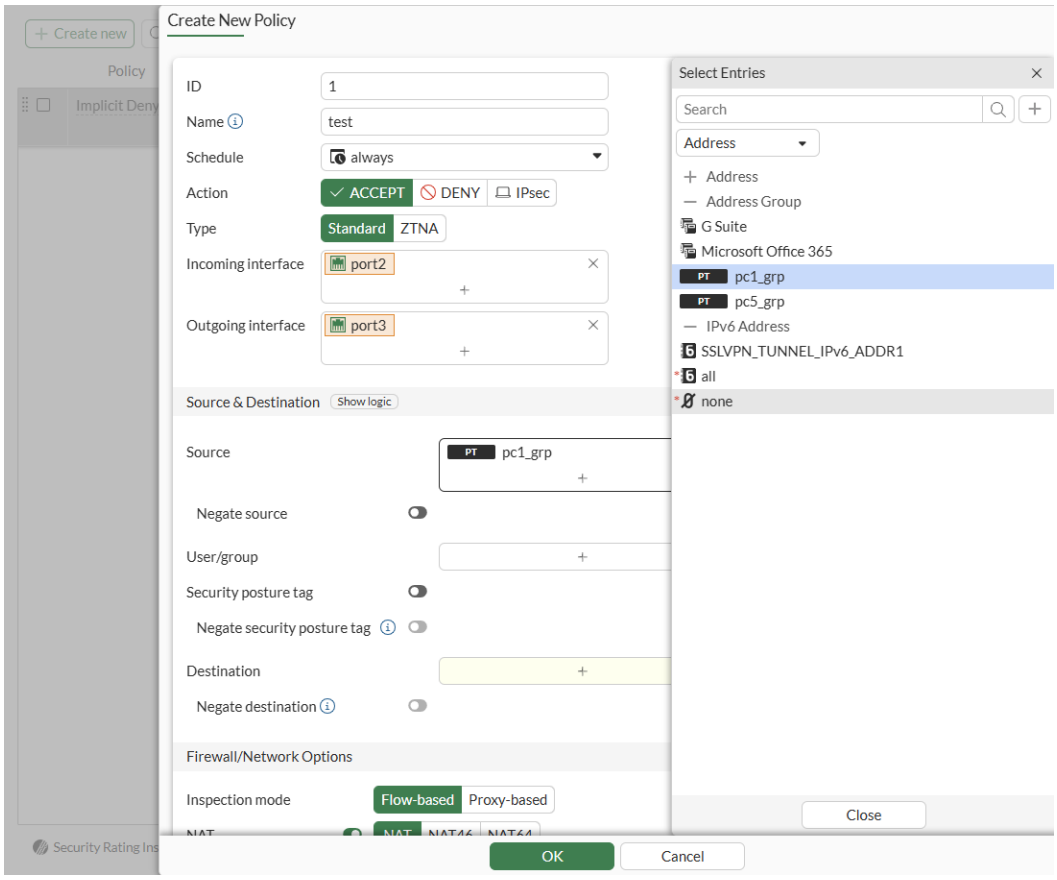
3. Create two dynamic tag address groups, with the tag names (pc1_tag and pc5_tag) as group members instead of the object names.
   a. In *Policy & Objects* > *Addresses*, click the *Address Group* tab.
   b. Click *Create new*.
   c. Enter a *Name*, such as pc1_grp.
   d. Set *Type* to *Dynamic tag*.
   e. Beside *Members*, click *+*, select the tag name (pc1_tag), and click *Close*.



f. Repeat this procedure to create an address group named pc5_grp and add custom tag pc5_tag as group member.



4. Enable policy tagging:
   a. Go to *System* > *Feature Visibility*.
   b. Under *Additional Features*, enable *Policy Tagging*.
   c. Click *Apply*.
5. Configure a firewall policy to allow traffic from PC1 to PC5.
   a. Go to *Policy & Objects* > *Firewall Policy*.
   b. Click *Create new*.
   c. Enter a *Name*.
   d. Beside *Source*, click *+*, select the address group named pc1_grp, and click *Close*.



e. Beside *Destination*, click *+*, select the address group named pc5_grp, and click *Close*

f. Set the remaining options as needed.

g. Click *OK*.

To create dynamic tag address groups in the CLI:

1. Create two custom tags:

```
config firewall custom-tag
    edit "pc1_tag"
    next
    edit "pc5_tag"
    next
end
```

2. Create two address objects named pc1 and pc5 with the pc1_tag and pc5_tag:

```
config firewall address
    edit "pc1"
        set custom-tags "pc1_tag"
        set subnet 10.1.100.111 255.255.255.255
    next

    edit "pc5"
```

```
        set custom-tags "pc5_tag"
        set subnet 172.16.200.155 255.255.255.255
    next
end
```

3. Create two dynamic tag address groups with the following tags as group members: pc1_tag and pc5_tag:

```
config firewall addrgrp
    edit "pc1_grp"
        set type dynamic-tag
        set custom-tags "pc1_tag"
    next
    edit "pc5_grp"
        set type dynamic-tag
        set custom-tags "pc5_tag"
    next
end
```

4. Configure a firewall policy to allow traffic from PC1 to PC5 using the address groups:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port3"
        set action accept
        set srcaddr "pc1_grp"
        set dstaddr "pc5_grp"
        set schedule "always"
        set service "ALL"
        set auto-asic-offload disable
    next
end
```

Zero Trust Network Access

This section includes information about ZTNA related new features:

- General on page 212
- Security posture and EMS connectors on page 242

General

This section includes information about general ZTNA related new features:

- ZTNA configuration simplification on page 212
- ZTNA service connector on page 235

ZTNA configuration simplification

FortiOS 8.0.0 introduces various configuration simplifications and modularization. Various objects are decoupled so they can be paired together in different combinations within the ZTNA policy.

CLI changes:

- firewall.access-proxy configs now split into different objects:
  - ztna.traffic-forward-proxy
  - ztna.web-proxy
  - ztna.web-portal
- firewall.proxy-policy now references a new ztna-proxy type referring to the 3 objects above
- Introduction of the ztna.destination object for traffic forwarding destinations
- Introduction of the firewall.vip.auth-portal setting to replace SAML SP access-proxy configuration
- ZTNA firewall policy and proxy policy references the ztna.destination object directly for ZTNA traffic forwarding

GUI changes:

- Configure ZTNA Ports (ZTNA VIPs) in the new ports page
- Allow ZTNA Port to be shared by multiple servers
- Configure ZTNA web proxy and traffic forwarding on different pages
- Support ZTNA Agentless web-portal

Naming conventions:

- A ZTNA HTTP/HTTPS access proxy is now a ZTNA web proxy
- A ZTNA TCP Forwarding access proxy (TFAP) is now a ZTNA Traffic Forwarding proxy (TFP)
- A ZTNA VIP object is now a ZTNA Port object

- A ZTNA destination refers to a Traffic Forwarding server

No changes:

- ZTNA Port CLI configurations

The changes are described in the following sections:

- Detail changes on page 213
- SAML authentication on page 229
- Upgrade on page 230
- Limitations and known issues on page 235

Detail changes

This section covers the following changes:

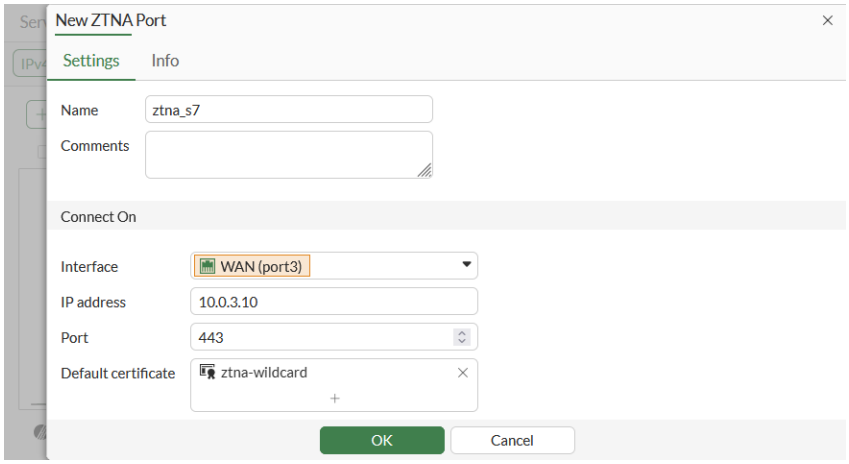
- Configuring a ZTNA Port object on page 213
- Configuring a ZTNA web proxy server on page 214
- Configuring a Traffic forwarding server on page 218
- Configuring a Web Portal for Agentless Access on page 219
- Configuring a ZTNA traffic policy on page 222
- Configuring a ZTNA firewall policy for web applications on page 222
- Configuring a ZTNA proxy policy for traffic forwarding on page 224
- Configuring a ZTNA proxy policy for Web Portal access on page 226

Configuring a ZTNA Port object

Previously, a ZTNA Port object is configured in the *ZTNA > ZTNA Server* page, which constrained one ZTNA Port object to one set of ZTNA real-servers within the same ZTNA server definition.

Now a ZTNA Port object is configured independently, and can be referenced in multiple different ZTNA server definitions.

Under *ZTNA > Ports*, configure your ZTNA Port object for IPv4 or IPv6:



No changes are made to the config firewall vip CLI settings for ZTNA Port object:

```
config firewall vip
    edit "ztna_s7"
        set type access-proxy
        set server-type https
        set extip 10.0.3.10
        set extintf "port3"
        set extport 443
        set ssl-certificate "ztna-wildcard"
    next
end
```

A ZTNA Port can also be configured directly in a ZTNA Web server, Traffic Forwarding Server, or Web Portal definition in the GUI.

Configuring a ZTNA web proxy server

In the GUI, the web proxy server can be configured using basic or advanced configuration mode:

- **Basic mode (default)**: use this mode when you need to configure a single real-server that is tied to this Web server object and virtual host.
- **Advanced mode**: use this mode when you need to configure multiple real-servers tied to this Web server object and virtual host. For example, you plan to specify different paths for various real-servers.

Using the basic method

1. Go to *Policy & Objects* > *ZTNA* > *Servers & Portals*.
2. Go to the *Web Server* sub-menu and click *Create New*.
3. Configure the following:

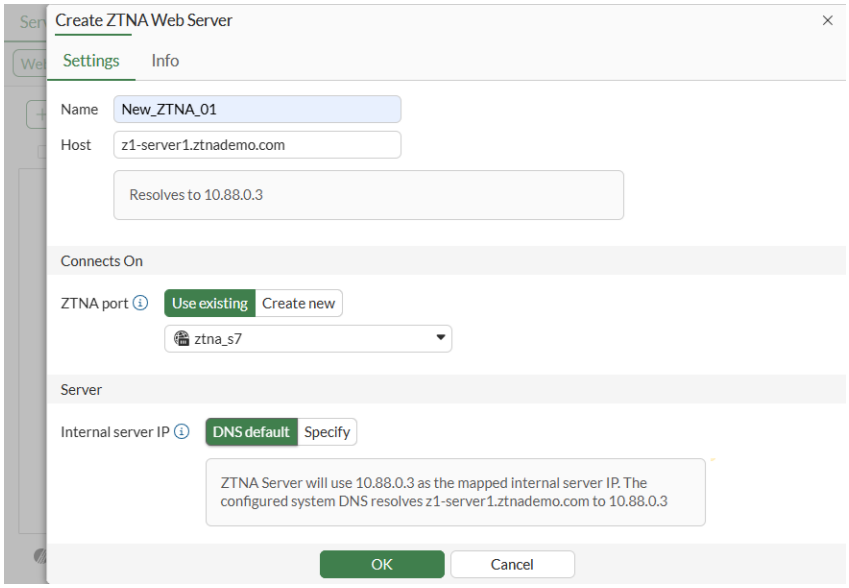
Setting	Description
Name	Name of the web server object.
IP version	Select IPv4 or IPv6. This option only appears if IPv6 is enabled on Feature Select.
Host*	Name or IP of the virtual host that your end users are accessing.
ZTNA port	• *Use existing*: Select a previously configured ZTNA Port. • *Create new*: Input settings for a new ZTNA Port.
Internal server IP	• *DNS default*: Automatically resolves the virtual host and configure the real server to be the IP of the resolved host. • *Specify*: Configure the real server manually. Select Service (HTTP/HTTPS) and IP. By default, the port will be 443. To change the port, use Advanced mode.

* Host (aka Virtual Host) fields are mandatory, since multiple entries can be paired with the same ZTNA Port. To differentiate between different servers, the virtual host field is used for matching.

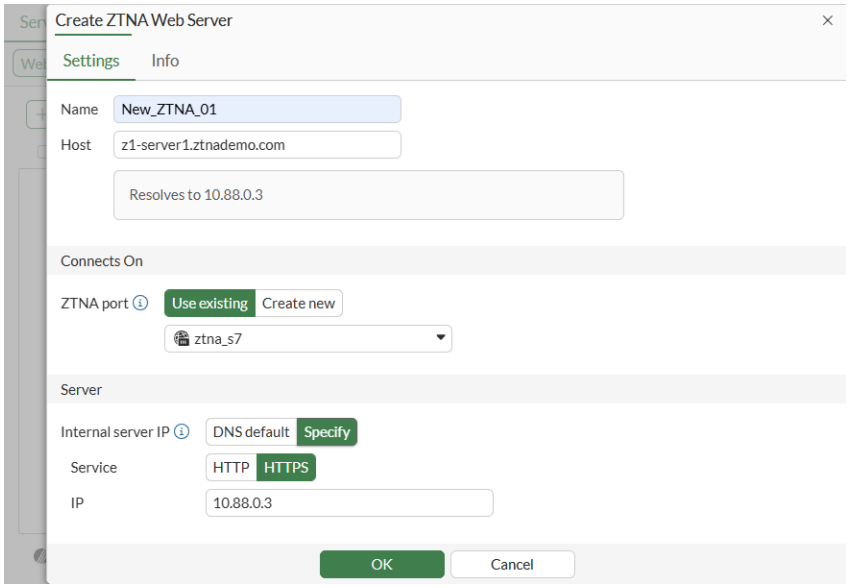
From the CLI, only one ZTNA server entry can have an empty host.

As a best practice, an external DNS should be used by remote clients to resolve the virtual-host to the ZTNA Port.

Example using DNS default for Internal server IP



Manually specify Internal server IP



4. Click *OK* to save.

This method automatically configures a virtual host object in the CLI and references it in the ZTNA web-proxy object.

```
config firewall access-proxy-virtual-host
    edit "web-proxy_vh_mggshssu"
        set ssl-certificate "ztna-wildcard"
        set host "z1-server1.ztnademo.com"
    next
end
config ztna web-proxy
    edit "New_ZTNA_01"
```

```
        set vip "ztna_s7"
        set host "web-proxy_vh_mggshssu"
        config api-gateway
            edit 1
                config realservers
                    edit 1
                        set ip 10.88.0.3
                    next
                end
            next
        end
    next
end
```

For other advanced virtual host settings, use the CLI:

```
config firewall access-proxy-virtual-host
    edit <host>
        set host-type { substring | fqdn* | wildcard }
        set empty-cert-action { accept | block* | accept-unmanageable }
        set client-cert { enable* | disable }
    next
end
```

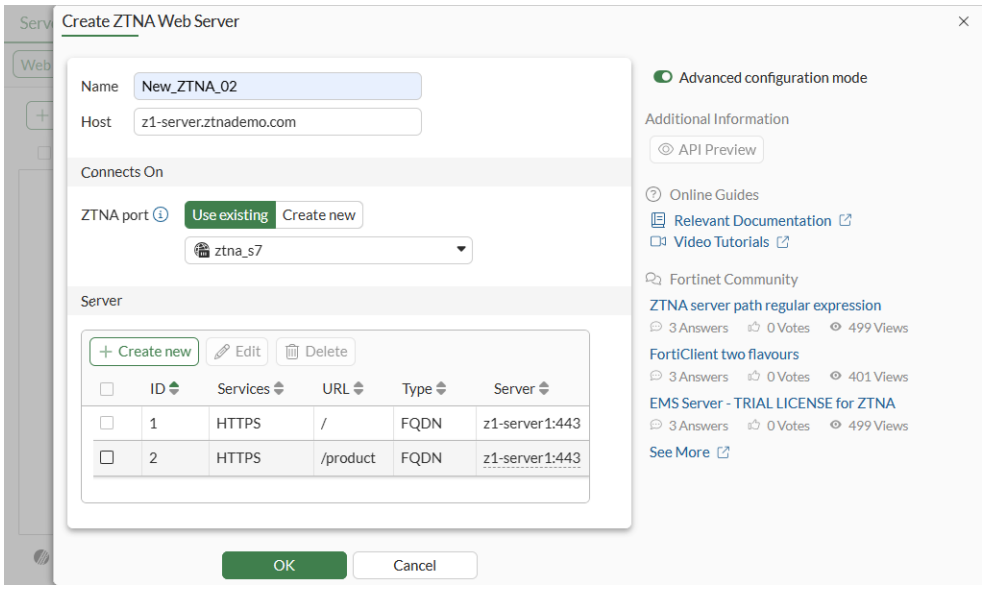
When using `host-type fqdn`, the host will only match when the subdomain matches the host. For example, the host *test.com* will match *vhost1.test.com*, but will not match *test.com.vhost*. On the other hand, `host-type substring` will match both.

Using the Advanced method

1. Go to *Policy & Objects* > *ZTNA* > *Servers & Portals*.
2. Go to the *Web Server* sub-menu and click *Create New*.
3. Enable *Advanced configuration mode*.
4. Enter the *Name*, *IP version*, *Host*, and *ZTNA port* settings as in the Basic method.
5. In the *Server* table click *Create new* to add a new real server.

Setting	Description
Service	HTTP or HTTPS
Path	Optional. Specify a specific path for matching. When creating more than one real server, the paths must be distinct.
Address Type	• *IP*: Input the Internal Server IP of the real server. • *FQDN*: The address of the real server can be chosen from pre-configured FQDN addresses.
Port	Specify the port used to connect to the real-server.

6. Click *OK*.



7. Click *OK* to save.

This method also automatically configures a virtual host object in the CLI and references it in the ZTNA web-proxy object:

```
config firewall access-proxy-virtual-host
        edit "web-proxy_vh_mggsn6e5"
        set ssl-certificate "ztna-wildcard"
        set host "z1-server.ztnademo.com"
    next
end
config ztna web-proxy
    edit "New_ZTNA_02"
        set vip "ztna_s7"
        set host "web-proxy_vh_mggsn6e5"
        config api-gateway
            edit 1
                config realservers
                    edit 1
                        set addr-type fqdn
                        set address "z1-server1"
                    next
                end
            next
            edit 2
                set url-map "/product"
                config realservers
                    edit 1
                        set addr-type fqdn
                        set address "z1-server1"
                    next
                end
            next
        end
```

```
    next
end
```

Configuring a Traffic forwarding server

Traffic forwarding (TCP and UDP) servers can now be configured on a separate tab. First, create the ZTNA destination, then add a traffic forwarding server.

1. Go to *Policy & Objects* > *ZTNA* > *Traffic Forwarding*.
2. Go to the *Destination* sub-menu and click *Create New*.
3. Configure the following:

Setting	Description
Name	Name of the destination
Type	• *On-premise*: Typically internal servers behind the firewall. Specify Address and Port. • *SaaS*: Access to known SaaS services gmail, dropbox, ms-teams,and others. Select SaaS application from list.
Port	Specify a port or range of ports separated by '-' from 1-65535.
Protocol	Specify All, TCP, or UDP.

4. Click *OK* to complete.
5. Go to the *Traffic Forwarding Server* sub-menu and click *Create New*.
6. Configure the following:

Setting	Description
Name	Name of the traffic forwarding server
Host*	Name or IP of the virtual host that your end users are accessing. Typically, for a traffic forwarding server, this is the external IP of the ZTNA port.
ZTNA port	• *Use existing*: Select a previously configured ZTNA VIP. • *Create new*: Input settings for a new ZTNA VIP.

* When a traffic forwarding server and destination is applied in a ZTNA policy, this triggers FortiGate to share the information with EMS in order to create a ZTNA application catalog entry. This ZTNA application catalog entry uses the external IP of the ZTNA port object as the ZTNA gateway address. Hence, the Host/virtual host definition within the traffic forwarding server should use the External IP of the ZTNA port.

7. Click *OK*.

This automatically configures a virtual host object in the CLI and references it in the ZTNA traffic forwarding object.

```
config firewall access-proxy-virtual-host
    edit "traffic-forward-proxy_vh_mggvr66n"
        set ssl-certificate "ztna-wildcard"
        set host "10.0.3.10"
    next
end
```

```
config ztna traffic-forward-proxy
    edit "New_TFS_01"
        set vip "ztna_s7"
        set host " traffic-forward-proxy_vh_mggvr66n"
    next
end
config ztna destination
    edit "z1-ws1"
        set uuid dea9af22-a3ba-51f0-7972-9a5de8d89e37
        set address "z1-workstation1"
        set mappedport 3389
    next
end
```

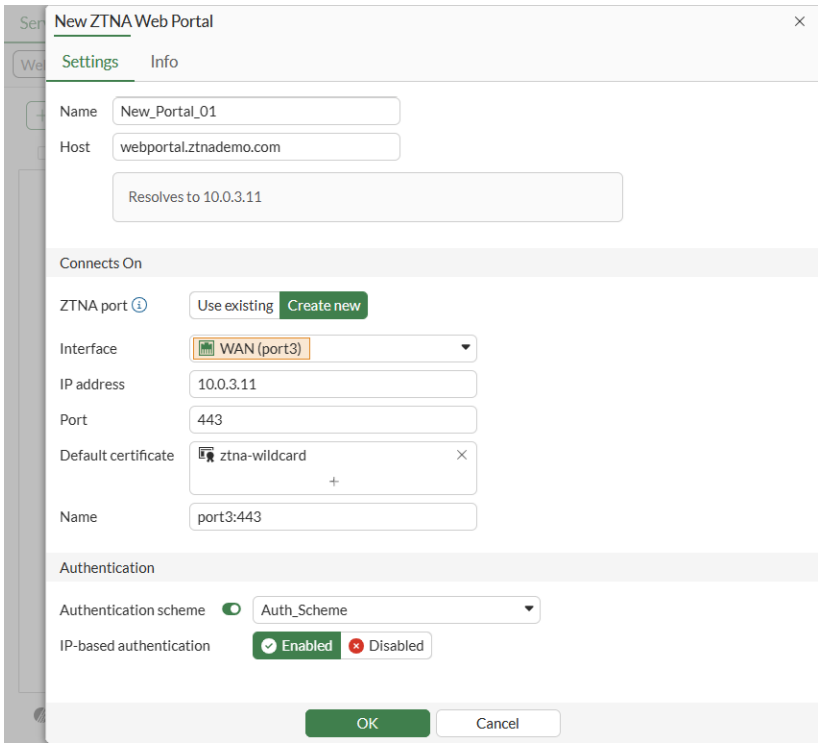
Note that the Traffic Forwarding Destination will not be synchronized to EMS until the destination is used in a proxy-policy.

Configuring a Web Portal for Agentless Access

For agentless access, FOS 7.6 introduced the CLI for configuring a ZTNA web portal. FOS 8.0 enhances this by introducing GUI configurations for the web portal settings.

1. Go to *Policy & Objects* > *ZTNA* > *Servers & Portals*.
2. Go to the *Web Portal* sub-menu and click *Create New*.
3. Configure the following:

Setting	Description
Name	Name of the web portal object
Host	Name of the virtual host that your end users are accessing. This should resolve to the ZTNA web portal itself
ZTNA port	• *Use existing*: Select a previously configured ZTNA VIP. • *Create new*: Input settings for a new ZTNA VIP.
Authentication Scheme	Enable to support authentication using an Authentication Scheme and Rule.
IP-based authentication	Enable authentication per IP address or per session.



4. Click *OK*.

The previous configuration generates the following CLI settings:

```
config firewall vip
    edit "port3:443"
        set type access-proxy
        set server-type https
        set extip 10.0.3.11
        set extintf "port3"
        set client-cert disable
        set extport 443
        set ssl-certificate "ztna-wildcard"
    next
end
config firewall access-proxy-virtual-host
    edit "web-portal_vh_mgh2dyqp"
        set ssl-certificate "ztna-wildcard"
        set host "webportal.ztnademo.com"
    next
end
config authentication rule
    edit "New_Portal_01_authrule"
        set status enable
        set protocol ztna-portal
        set ip-based enable
        set active-auth-method "Auth_Scheme"
        set sso-auth-method ''
        set comments ''
    next
```

```
end
config ztna web-portal
    edit "New_Portal_01"
        set vip "port3:443"
        set host "web-portal_vh_mgh2dyqp"
        set auth-rule "New_Portal_01_authrule"
    next
end
```

The authentication scheme is pre-configured in this example, which references a LDAP server:

```
config authentication scheme
    edit "Auth_Scheme"
        set method basic
        set user-database "LDAP-fortiad"
    next
end
```

Web portal bookmarks still need to be configured from the CLI. Most importantly, your user and groups must be defined in the bookmark setting to trigger access control on the ZTNA web portal login:

```
config ztna web-portal-bookmark
    edit "b"
        set groups "LDAP-Finance"
        config bookmarks
            edit "Webserver"
                set url "https://s2.ztnademo.com:9043"
            next
            edit "Server-S1-Web"
                set sso enable
                set url "https://s1.ztnademo.com"
            next
            edit "Server-S1-SSh"
                set apptype ssh
                set host "10.88.0.7"
            next
            edit "FortiGate-Internal-SSH"
                set apptype ssh
                set host "10.88.0.254"
            next
            edit "RDP"
                set apptype rdp
                set host "10.88.0.1"
                set port 3389
            next
        end
    next
end
```

See ZTNA agentless web-based application access for details.

Configuring a ZTNA traffic policy

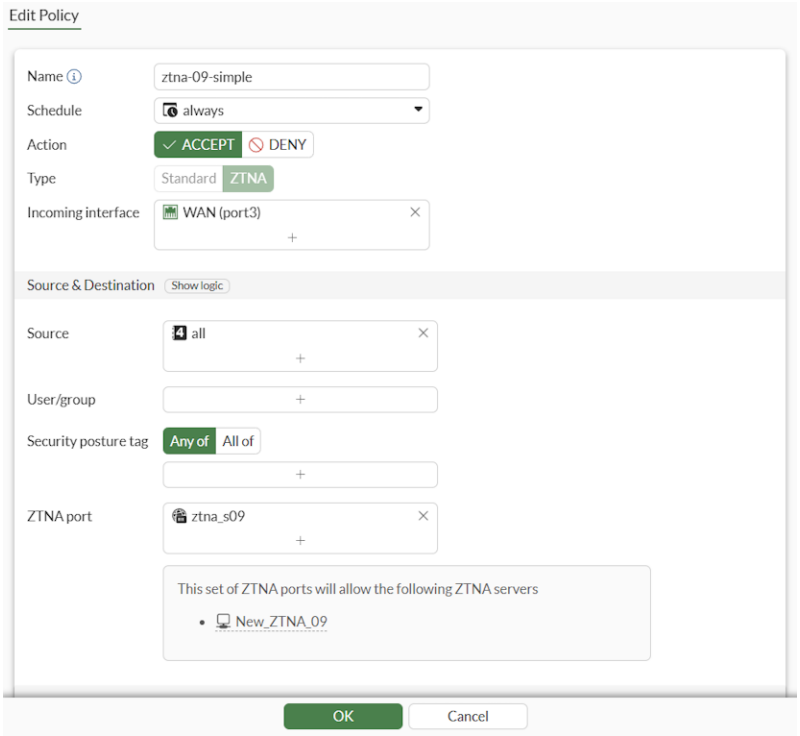
ZTNA traffic policy can be configured using the ZTNA firewall policy, or a ZTNA proxy policy.

The difference is ZTNA firewall policy simplifies the config so specifying a ZTNA port will automatically allow all web-proxy associated with it. Also, all traffic forwarding servers are added.

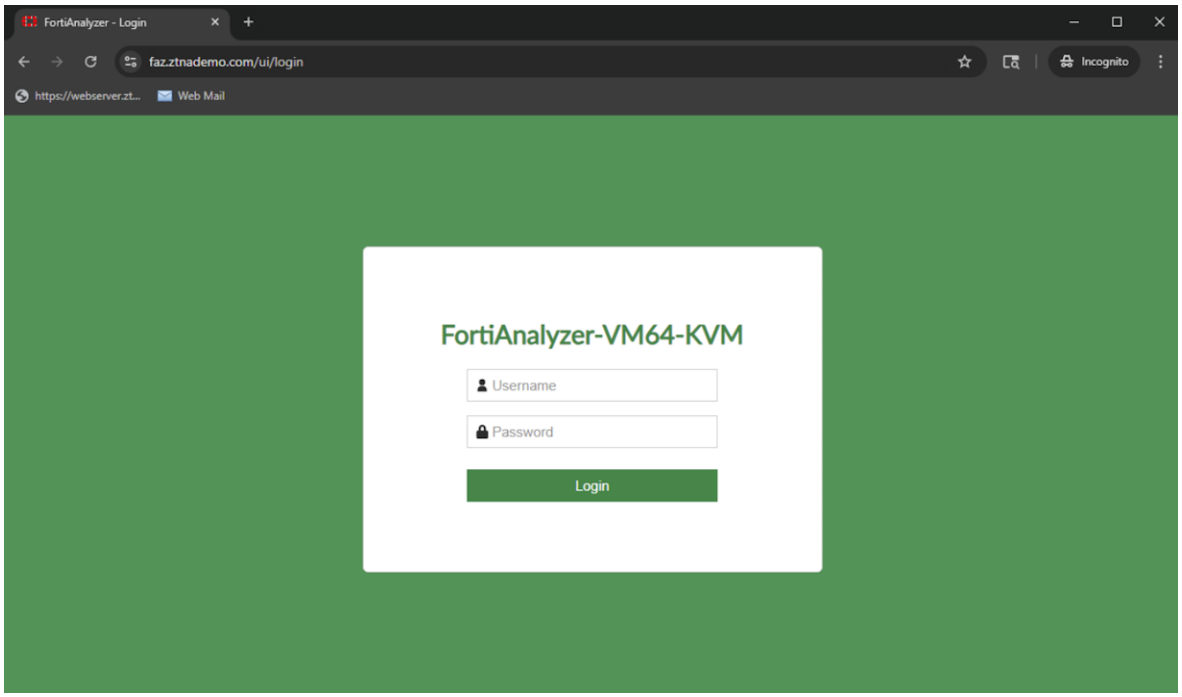
ZTNA proxy policy allows more granularity for controlling the exact ZTNA server and ZTNA destination you wish to allow. Adding a traffic forwarding destination also triggers this destination to be synchronized to the EMS ZTNA Application Catalog.

Configuring a ZTNA firewall policy for web applications

In this scenario, a ZTNA Port is selected for accessing a ZTNA web server.



From a client, connecting to faz.ztnademo.com resolves to the ZTNA Port and user can access the ZTNA server accordingly.



Logs:

```
# execute log filter field subtype ztna
# execute log display

16 logs found.
10 logs returned.

1: date=2025-10-07 time=19:00:56 eventtime=1759888856776286391 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=4816 srcintf="port3"
srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.2 dstport=443
dstintf="port2" dstintfrole="dmz" sessionid=2059 srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1"
service="HTTPS" proxyapptype="ztna-proxy" proto=6 action="accept" policyid=19 policytype="policy"
poluuid="91b5d936-a3d1-51f0-99c6-2d2b1023f83f" policyname="ztna-09-simple" appcat="unscanned"
duration=8 gatewayid=1 realserverid=1 vip="ztna_s09" vipincomingip=10.0.3.9 accessproxy="New_ZTNA_
09" clientdevicemanageable="manageable" clientcert="yes" wanin=4199365 rcvdbyte=4199365
wanout=42022 lanin=9513 sentbyte=9513 lanout=4229931
```

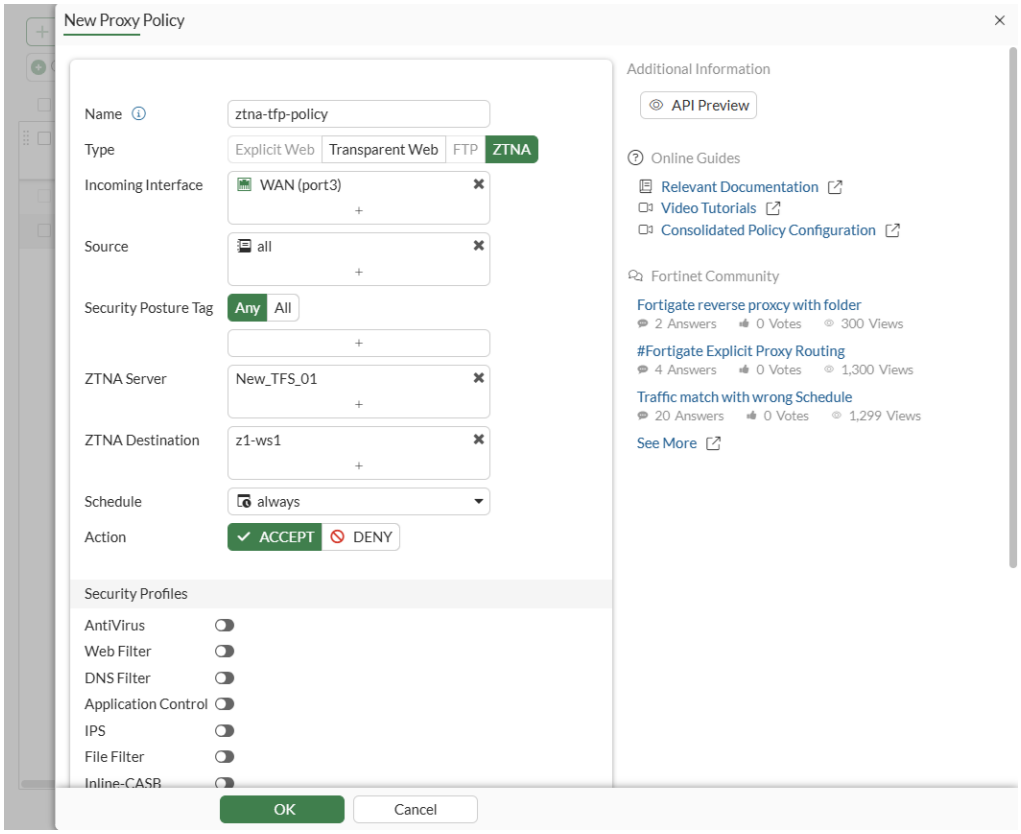
Note that in some scenarios where the FortiGate cannot verify the certificate of the server, you can bypass this check for the specific realserver.

```
config ztna web-proxy
   edit <server>
       config api-gateway
           edit <ID>
               config realservers
                   edit <ID>
                       set verify-cert disable
                   next
               end
           next
```

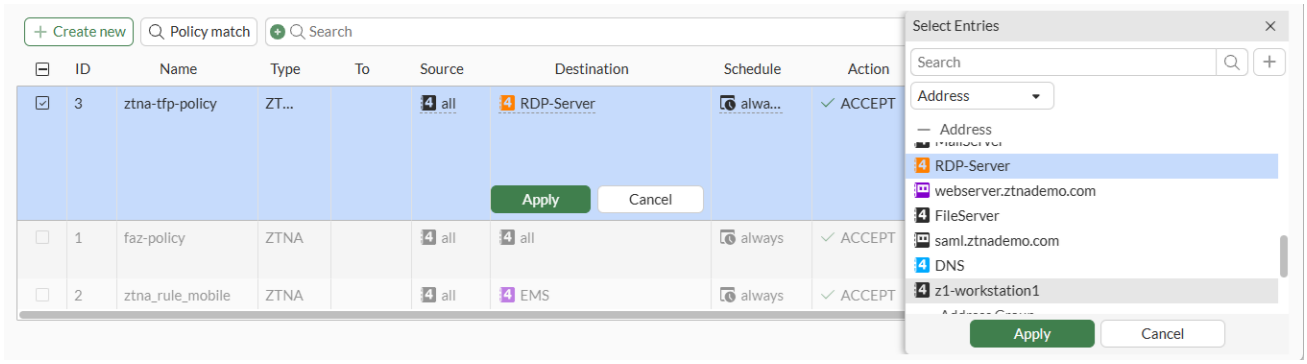
```
        end
    next
end
```

Configuring a ZTNA proxy policy for traffic forwarding

In this scenario, a proxy policy is used to allow traffic to a specific Traffic forwarding destination.



Once the ZTNA proxy policy is created, you edit the Destination Address from the policy view.

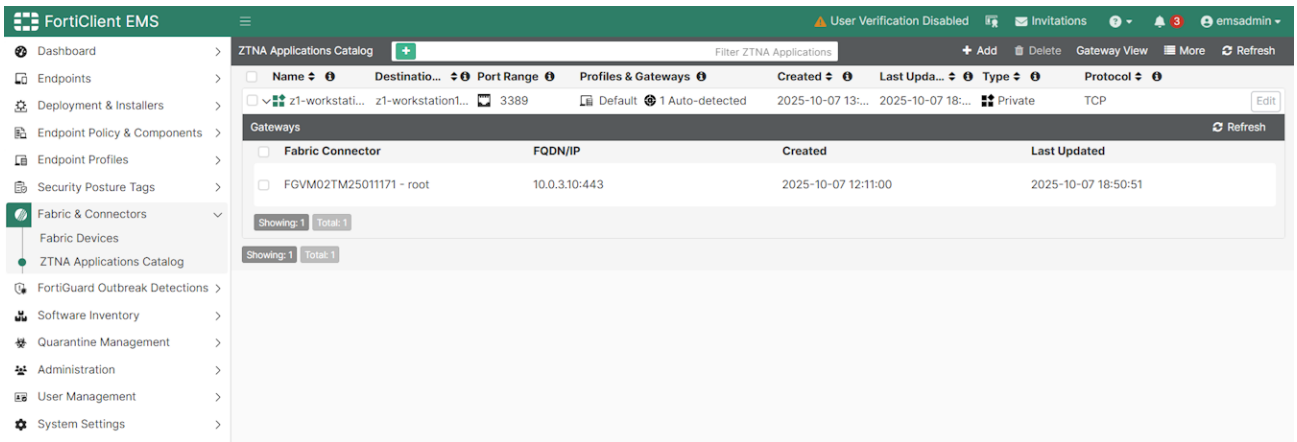


From the CLI:

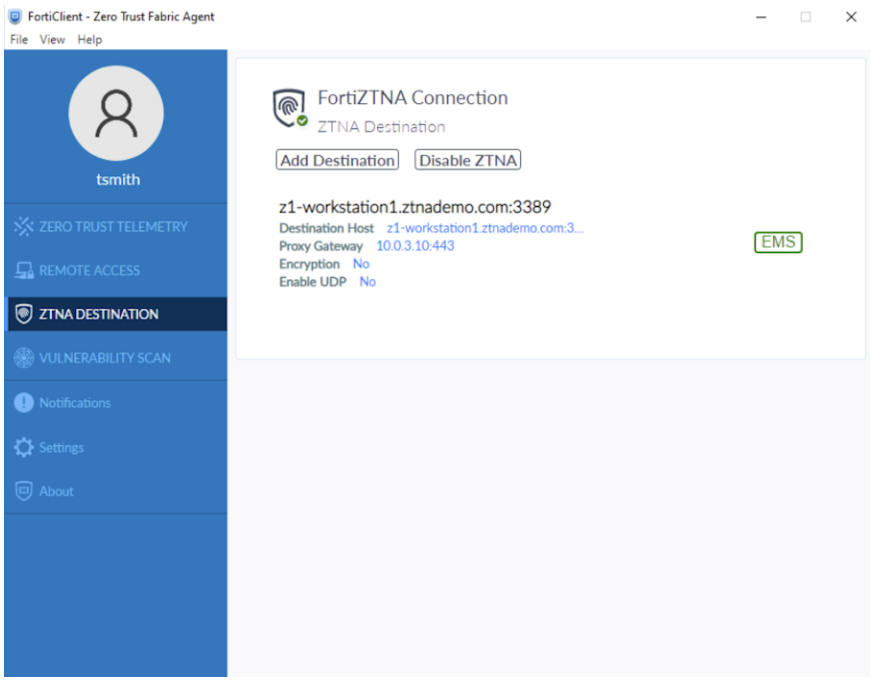
```
config firewall proxy-policy
    edit 1
```

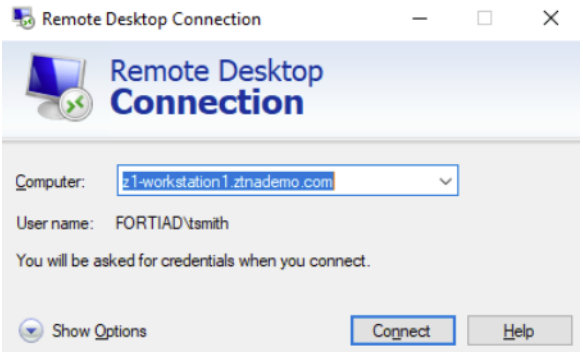
```
        set name “ztna-tfp-policy”
        set proxy ztna-proxy
        set ztna-proxy "New_TFS_01"
        set ztna-destination "z1-ws1"
        set srcintf "port3"
        set srcaddr "all"
        set dstaddr "RDP-Server"
        set action accept
        set schedule "always"
        set logtraffic all
    next
end
```

With the ZTNA traffic forwarding proxy and destination defined, this mapping will be pushed to EMS.



Once applied to an Endpoint Profile and pushed to a FortiClient, the user on the endpoint can now access the service.



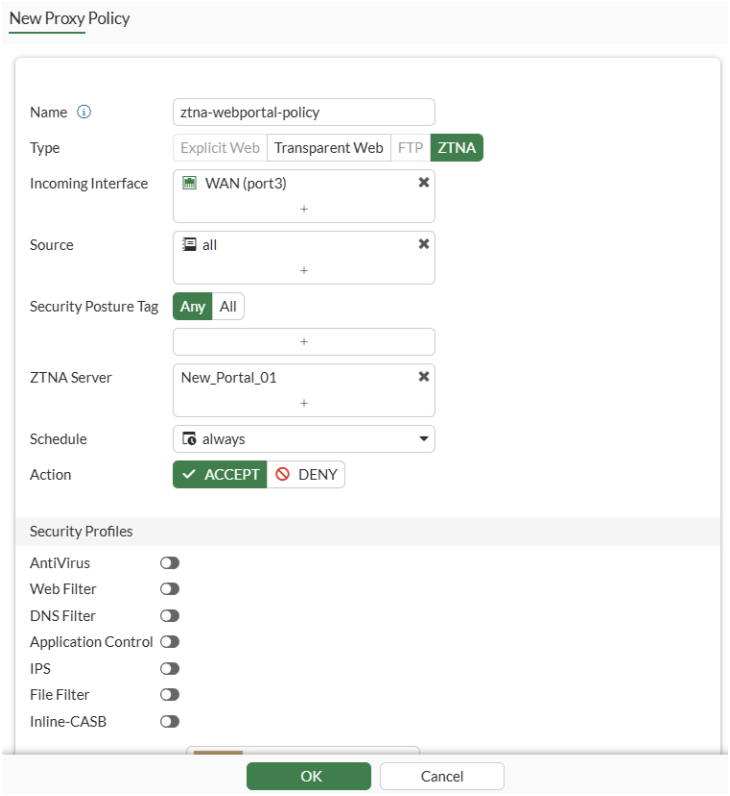


Logs:

```
# execute log filter field subtype ztna
# execute log display
21 logs found.
10 logs returned.

1: date=2026-01-23 time=11:48:57 eventtime=1769197736891659763 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=8033 srcintf="port3"
srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.1 dstport=3389
dstintf="port2" dstintfrole="dmz" sessionid=761460 srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1"
dstuuid="b872542a-94ce-51ec-993a-608548a2ab65" service="RDP" proxyapptype="ztna-proxy" proto=6
action="accept" policyid=1 policytype="proxy-policy" poluuid="a2e00a54-f892-51f0-f475-
047ec06ea3a7" policyname="ztna-tfp-policy" appcat="unscanned" duration=7 gatewayid=1 vip="ztna_s7"
vipincomingip=10.0.3.10 accessproxy="New_TFS_01" clientdevicemanageable="manageable"
clientcert="yes" wanin=154496 rcvdbyte=154496 wanout=67763 lanin=69733 sentbyte=69733
lanout=157625
```

Configuring a ZTNA proxy policy for Web Portal access

A proxy policy must be configured to allow Web Portal access.



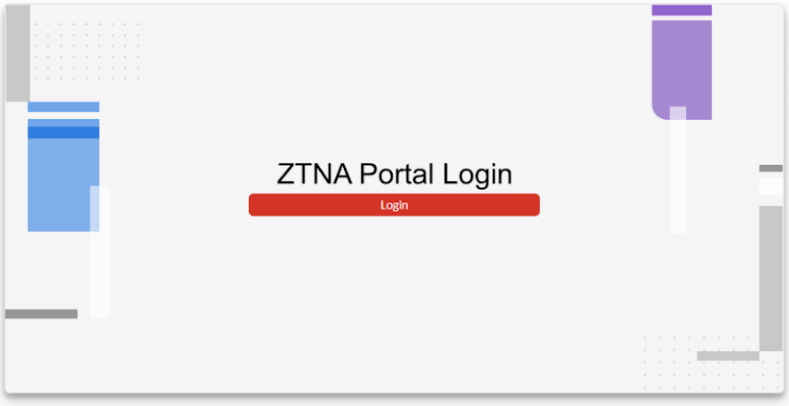

```
config firewall proxy-policy
    edit 2
        set proxy ztna-proxy
        set name "ztna-webportal-policy"
        set ztna-proxy "New_Portal_01"
        set srcintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set logtraffic all
    next
end
```

The user group is taken care of in the web portal bookmarks settings. Therefore, the proxy policy above simply define the proxy that is allowed.

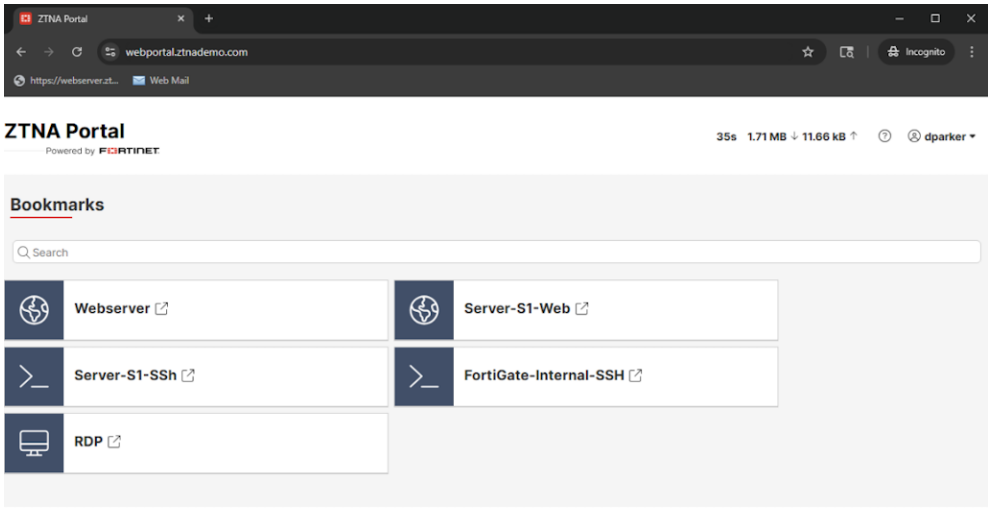
For complete agentless access without client certificate check, perform one more step:

```
config firewall access-proxy-virtual-host
    edit "web-portal_vh_mgh2dyqp"
        set ssl-certificate "ztna-wildcard"
        set host "webportal.ztnademo.com"
        set client-cert disable
    next
end
```

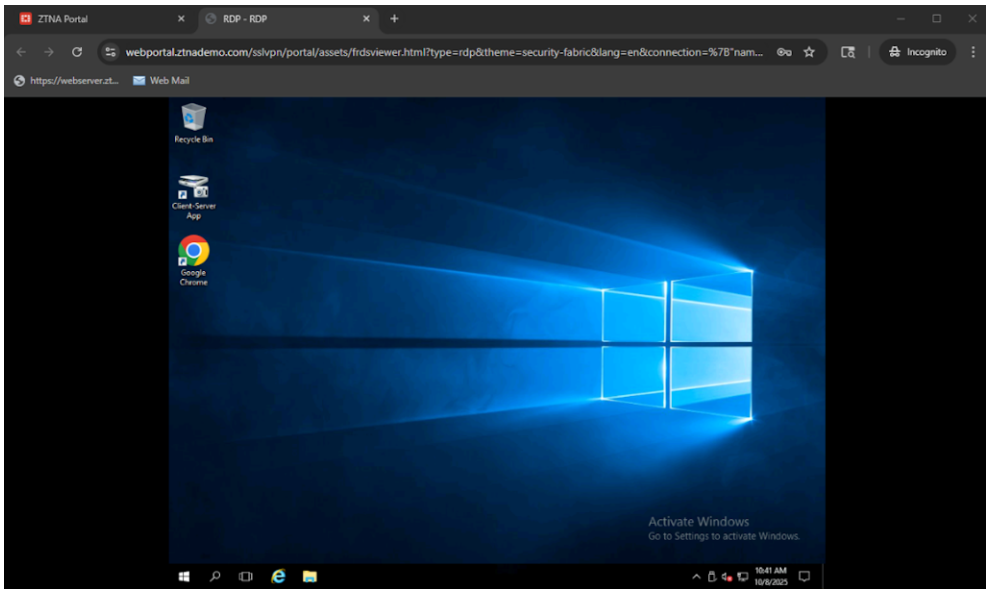
From a web client which does not have any FortiClient agent installed, access webportal.ztandemo.com. The client must be able to resolve this to the ZTNA Port.



Sign in to the web portal:



Access a shortcut:



Logs:

```
# execute log filter field subtype ztna
# execute log display
26 logs found.
10 logs returned.

1: date=2025-10-08 time=10:40:56 eventtime=1759945256077145902 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.3 srcport=30265
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.1
dstport=3389 dstintf="port2" dstintfrole="dmz" sessionid=14266 srcuuid="b458a65a-f759-51ea-d7df-
ef2e750026d1" dstuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" service="RDP" proxyapptype="ztna-
proxy" proto=6 action="accept" policyid=2 policytype="proxy-policy" poluuid="62fc2e70-a3f8-51f0-
b249-0d2835753986" appcat="unscanned" duration=0 gatewayid=1 vip="port3:443"
vipincomingip=10.0.3.11 clientdevicemanageable="unknown" clientcert="yes" wanin=0 rcvdbyte=0
wanout=0 lanin=3576 sentbyte=3576 lanout=463
```

SAML authentication

Previously, the access-proxy linked a VIP and a samlsp service and was referenced directly in a proxy policy to allow SAML authentication for a ZTNA policy:

```
config firewall access-proxy
    edit "ztna-saml"
        set vip "ztna-saml-vip"
        config api-gateway
            edit 1
                set service samlsp
                set saml-server "fac-saml"
            next
        end
```

```
   next
end
```

```
config firewall proxy-policy
    edit 2
        set name "SAML_AUTH_ACCESS"
        set proxy access-proxy
        set access-proxy "ztna-saml"
        set srcintf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set logtraffic all
    next
end
```

Now a new setting in the VIP `set auth-portal` controls SAML authentication access by matching the VIP's extip:extport tuple to a user.saml.single-sign-on-url. In this example, demo.url.com resolves to 172.18.62.67.

```
config firewall vip
    edit "ztna-saml"
        set vip-id 2
        set type access-proxy
        set server-type https
        set extip 172.18.62.67
        set extintf "port2"
        set auth-portal enable
        set extport 9443
        set ssl-certificate "default.test.com"
    next
end
```

```
config user saml
    edit "ztna-saml"
        set entity-id "http://demo.url.com:9443/remote/saml/metadata/"
        set single-sign-on-url "https://demo.url.com:9443/remote/saml/login"
        set single-logout-url "https://demo.url.com:9443/remote/saml/logout"
        set idp-entity-id "http://fac.demo.url.com/saml-idp/ztna/metadata/"
        set idp-single-sign-on-url "https://fac.demo.url.com/saml-idp/ztna/login/"
        set idp-single-logout-url "https://fac.demo.url.com/saml-idp/ztna/logout/"
        set idp-cert "REMOTE_Cert_3"
        set user-name "username"
    next
end
```

Upgrade

When upgrading from FortiOS 7.6 to 8.0, configurations can be migrated.

Example CLI configuration of ZTNA SAML authentication and the changes between FortiOS 7.6 and 8.0. The configuration previously found under firewall.access-proxy that enabled SAML service for the VIP has been removed and replaced by a new setting added to the VIP: `set auth-portal enable`.

FOS 7.6	FOS 8.0

`config firewall vip`
`  edit "ztna-saml-vip"`
`  set vip-id 2`
`  set type access-proxy`
`  set server-type https`
`  set extip 172.18.62.67`
`  set extintf "port2"`
`  set extport 9443`
`  set ssl-certificate "default.test.com"`
`  next`
`end`
	`config firewall vip`
	`  edit "ztna-saml"`
	`  set vip-id 2`
	`  set type access-proxy`
	`  set server-type https`
	`  set extip 172.18.62.67`
	`  set extintf "port2"`
	`  set auth-portal enable`
	`  set extport 9443`
	`  set ssl-certificate "default.test.com"`
	`  next`
	`end`

`config firewall access-proxy`
`  edit "ztna-saml"`
`  set vip "ztna-saml-vip"`
`  config api-gateway`
`  edit 1`
`  set service samlsp`
`  set saml-server "fac-saml"`
`  next`
`  end`
`  next`
`end`
	Removed

`config user saml`
`  edit "ztna-saml"`
`  set entity-id`
`"http://demo.url.com:9443/remote/saml/metadata/"`
`  set single-sign-on-url`
`"https://demo.url.com:9443/remote/saml/login"`
`  set single-logout-url`
`"https://demo.url.com:9443/remote/saml/logout"`
`  set idp-entity-id`
`"http://fac.demo.url.com/saml-idp/ztna/metadata/"`
`  set idp-single-sign-on-url`
`"https://fac.demo.url.com/saml-idp/ztna/login/"`
`  set idp-single-logout-url`
`"https://fac.demo.url.com/saml-idp/ztna/logout/"`
`  set idp-cert "REMOTE_Cert_3"`
`  set user-name "username"`
`  next`
`end`
	`config user saml`
	`  edit "ztna-saml"`
	`  set entity-id`
	`"http://demo.url.com:9443/remote/saml/metadata/"`
	`  set single-sign-on-url`
	`"https://demo.url.com:9443/remote/saml/login"`
	`  set single-logout-url`
	`"https://demo.url.com:9443/remote/saml/logout"`
	`  set idp-entity-id`
	`"http://fac.demo.url.com/saml-idp/ztna/metadata/"`
	`  set idp-single-sign-on-url`
	`"https://fac.demo.url.com/saml-idp/ztna/login/"`
	`  set idp-single-logout-url`
	`"https://fac.demo.url.com/saml-idp/ztna/logout/"`
	`  set idp-cert "REMOTE_Cert_3"`
	`  set user-name "username"`
	`  next`
	`end`

Example CLI configuration of a ZTNA web proxy and the changes between FOS 7.6 to FOS 8.0. A single firewall.access-proxy entry is split into two ztna.web-proxy entries.

FOS 7.6

```
config firewall access-proxy
    edit "ztna_s7"
        set vip "ztna_s7"
        set auth-portal enable
        config api-gateway
            edit 1
                config realservers
                    edit 1
                        set addr-type fqdn
                        set address
"server1.ztnademo.com"
                        set status standby
                    next
                end
            next
            edit 2
                set virtual-host "auto-ztna_s7-
0"
                config realservers
                    edit 1
                        set addr-type fqdn
                        set address
"s25.qa.fortinet.com"
                        set verify-cert disable
                    next
                end
            next
        end
    next
end
```

FOS 8.0

```
config ztna web-proxy
    edit "ap_ztna_s7_gwy_1_webproxy"
        set vip "ztna_s7"
        config api-gateway
            edit 1
                config realservers
                    edit 1
                        set addr-type fqdn
                        set address
"server1.ztnademo.com"
                    next
                end
            next
        end
    next
    edit "ap_ztna_s7_gwy_2_webproxy"
        set vip "ztna_s7"
        set host "auto-ztna_s7-0"
        config api-gateway
            edit 2
                config realservers
                    edit 1
                        set addr-type fqdn
                        set address
"s25.qa.fortinet.com"
                        set verify-cert disable
                    next
                end
            next
        end
    next
end
```

Example CLI configuration of a ZTNA traffic forwarding server and the changes between FOS 7.6 to FOS 8.0. A single firewall.access-proxy entry is split into a new ztna.traffic-forward-proxy entry and multiple ztna.destination entries.

FOS 7.6

```
config firewall access-proxy
    edit "ztna_s7"
```

FOS 8.0

```
config ztna traffic-forward-proxy
    edit "ap_ztna_s7_host__ztfp"
        set vip "ztna_s7"
```

FOS 7.6

```
        set vip "ztna_s7"
        set auth-portal enable
        config api-gateway
            edit 3
                set url-map "/tcp"
                set service tcp-forwarding
                config realservers
                    edit 1
                        set address "fgt"
                        set mappedport 445 443
4400-4401 23
                    next
                    edit 2
                        set address
"s27.qa.fortinet.com"
                    next
                    edit 3
                        set address "server25"
                        set mappedport 22
                        set type ssh
                        set ssh-client-cert
"ssh-access-proxy"
                        set ssh-host-key-
validation enable
                        set ssh-host-key
"ed25519"
                    next
                    edit 4
                        set address "smb101245"
                        set mappedport 445
                    next
                    edit 5
                        set address
"internal1.fortinet.com"
                    next
                end
            next
            edit 4
                set url-map "/saas"
                set service saas
                set application "MS" "Google"
"dropbox"
            next
        end
    next
end
```

FOS 8.0

```
    next
end
config ztna destination
    edit "ap_ztna_s7_host__svr_1_zdst"
        set uuid c8dff876-a000-51f0-4a68-
10edb63a5d59
        set address "fgt"
        set mappedport 445 443 4400-4401 23
    next
    edit "ap_ztna_s7_host__svr_2_zdst"
        set uuid c8dffb5a-a000-51f0-2edc-
e3f8802f351f
        set address "s27.qa.fortinet.com"
    next
    edit "ap_ztna_s7_host__svr_3_zdst"
        set uuid c8dffcae-a000-51f0-d56c-
3de66b250af5
        set address "server25"
        set mappedport 22
    next
    edit "ap_ztna_s7_host__svr_4_zdst"
        set uuid c8dffdd0-a000-51f0-4a4a-
b3f75bb3d353
        set address "smb101245"
        set mappedport 445
    next
    edit "ap_ztna_s7_host__svr_5_zdst"
        set uuid c8dffef2-a000-51f0-dd59-
55cf4f11afab
        set address "internal1.fortinet.com"
    next
    edit "ap_ztna_s7_host__saas_4_zdst"
        set uuid c8e00078-a000-51f0-d9a4-
bbd4e6c76b74
        set type saas
        set saas-application "MS" "Google"
"dropbox"
    next
end
```

Example CLI configuration of a ZTNA proxy policy and the changes between FOS 7.6 to FOS 8.0. A single firewall.proxy-policy entry is split into two firewall.proxy-policy entries for web-proxy and traffic forwarding servers and destinations.

FOS 7.6	FOS 8.0
config firewall proxy-policy edit 3 set uuid 9af7a056-4337-51f0-7da1-1b8273a78ae4 set name "ZTNA_proxy_policy_02" set proxy access-proxy set access-proxy "ztna_s7" set srcintf "port2" set srcaddr "all" set dstaddr "all" set ztna-ems-tag "EMS7_ZTNA_ems_linux30_management_tag" set action accept set schedule "always" set logtraffic all next end	config firewall proxy-policy edit 3 set uuid 9af7a056-4337-51f0-7da1-1b8273a78ae4 set name "ZTNA_proxy_policy_02" set proxy ztna-proxy set ztna-proxy "ap_ztna_s7_gwy_1_webproxy" "ap_ztna_s7_gwy_2_webproxy" set srcintf "port2" set srcaddr "all" set dstaddr "all" set ztna-ems-tag "EMS7_ZTNA_ems_linux30_management_tag" set action accept set schedule "always" next edit 5 set uuid 40843cae-9aff-51f0-0e7a-2f1cbb3c1432 set proxy ztna-proxy set ztna-proxy "ap_ztna_s7_host__ztfp" set ztna-destination "ap_ztna_s7_host__saas_4_zdst" "ap_ztna_s7_host__svr_1_zdst" "ap_ztna_s7_host__svr_2_zdst" "ap_ztna_s7_host__svr_3_zdst" "ap_ztna_s7_host__svr_4_zdst" "ap_ztna_s7_host__svr_5_zdst" set srcintf "port2" set srcaddr "all" set dstaddr "all" set ztna-ems-tag "EMS7_ZTNA_ems_linux30_management_tag" set action accept set schedule "always" next end

Example CLI configuration of a ZTNA firewall policy and the changes between FOS 7.6 to FOS 8.0. ztna.destinations are defined within the new ZTNA firewall policy for traffic forwarding destinations.

FOS 7.6	FOS 8.0
config firewall policy	config firewall policy

FOS 7.6	FOS 8.0
edit 3 set name "ZTNA_policy_01" set uuid 760d00d4-7497-51ef-cf53-73b4f106a8a0 set srcintf "port2" set dstintf "any" set action accept set srcaddr "all" set dstaddr "ztna_s7" set ztna-ems-tag "EMS7_ZTNA_ems_linux30_management_tag" set schedule "always" set utm-status enable set ssl-ssh-profile "custom-deep-inspection" set av-profile "g-default" set logtraffic all set nat enable next end	edit 3 set name "ZTNA_policy_01" set uuid 760d00d4-7497-51ef-cf53-73b4f106a8a0 set srcintf "port2" set dstintf "any" set action accept set srcaddr "all" set dstaddr "ztna_s7" set ztna-ems-tag "EMS7_ZTNA_ems_linux30_management_tag" **set ztna-destination "ap_ztna_s7_host__saas_4_zdst" "ap_ZTNA14_host_auto-ZTNA_svr_1_zdst" "ap_ZTNA14_host__saas_3_zdst" "ap_ztna_s7_host__svr_1_zdst" "ap_ztna_s7_host__svr_2_zdst" "ap_ztna_s7_host__svr_3_zdst" "ap_ztna_s7_host__svr_4_zdst" "ap_ztna_s7_host__svr_5_zdst"** set schedule "always" set utm-status enable set ssl-ssh-profile "custom-deep-inspection" set av-profile "g-default" set nat enable set groups "grp_radius" next end

Limitations and known issues

Limitations

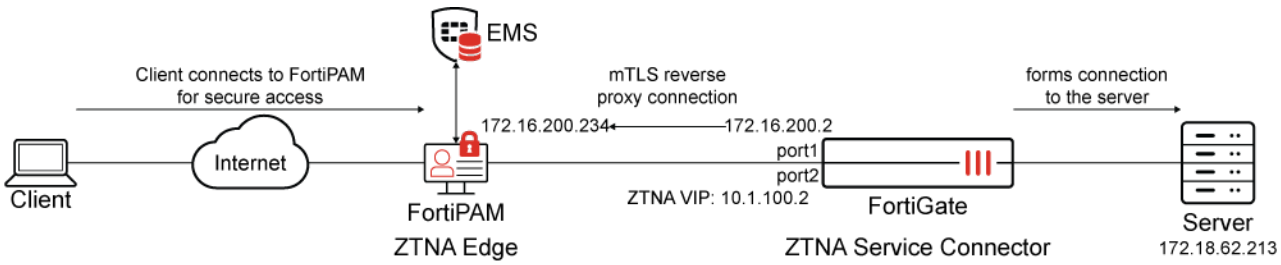
1. The host field is required for Web proxy server and Traffic forwarding server whereas this was not a required field in 7.6
2. Hosts are by default FQDN type when defined in the GUI. To define a wildcard or substring type virtual host, use the CLI.
3. ZTNA destinations can be defined for a ZTNA firewall policy, but only in the CLI.
4. Traffic forwarding destinations are only pushed to EMS when applied in a ZTNA policy

ZTNA service connector

Up until FortiOS 7.6, FortiGate was able to act as the ZTNA access proxy and provide forward proxy services directly between clients and protected services.

In certain environments, we may want to prevent users from accessing the FortiGate ZTNA access proxy directly or expose ports and services on the FortiGate to the public. In these scenarios, another layer of ZTNA proxy may be required to serve as the ZTNA Edge.

FortiOS 8.0 introduces the ability for a FortiGate to act as a ZTNA service connector to reverse proxy service connections upstream to the ZTNA Edge. In this scenario, the ZTNA Edge may be a FortiPAM or FortiProxy which acts as the ZTNA access proxy to the end users. The FortiGate, as the ZTNA service connector, forms a persistent control connection with ZTNA Edge. The ZTNA Edge forwards connection requests through the control tunnel to the FortiGate ZTNA service connector, which proxies the request to the protected server.



The ZTNA Edge in this scenario completes the ZTNA checks and enforces the ZTNA policy. If the connection is allowed, the request is forwarded to the ZTNA service connector.

The ZTNA service connector is responsible for proxying the connection and establishing a data connection back to the ZTNA Edge to forward the response. This effectively forms a proxy chain where the FortiGate service connector provides reverse proxy connection back to the ZTNA Edge.

Usage

CLI syntax

The following CLI settings configure the connection with the ZTNA Edge:

```
config ztna reverse-connector
    edit <name>
        set address <IP or FQDN address>
        set port <port number>
        set certificate <local certificate>
        set trusted-server-ca <CA certificate>
    next
end
```

Field	Description
address <IP or FQDN address>	Address of the ZTNA Edge
port <port number>	The port number for connecting to the ZTNA Edge
certificate <local certificate>	The local client certificate used for mTLS negotiation
trusted-server-ca <CA certificate>	The root CA certificate to authenticate the mTLS connection

GUI settings

1. Go to *System* > *Feature Visibility* and enable *Zero Trust Network Access* and *ZTNA Reverse Proxy Connector*.
2. Go to *Policy & Objects* > *ZTNA* and select the *Traffic Forwarding* tab.
3. Select the *Reverse Proxy Connector* sub-menu and click *Create New*.
4. Configure the following, then click *OK*:

Field	Description
Name	Name of the connector
Address	IP or FQDN Address of the ZTNA Edge
Port	The port number for connecting to the ZTNA Edge
CA certificate	The root CA certificate to authenticate the mTLS connection
Default certificate	The local client certificate used for mTLS negotiation

Example

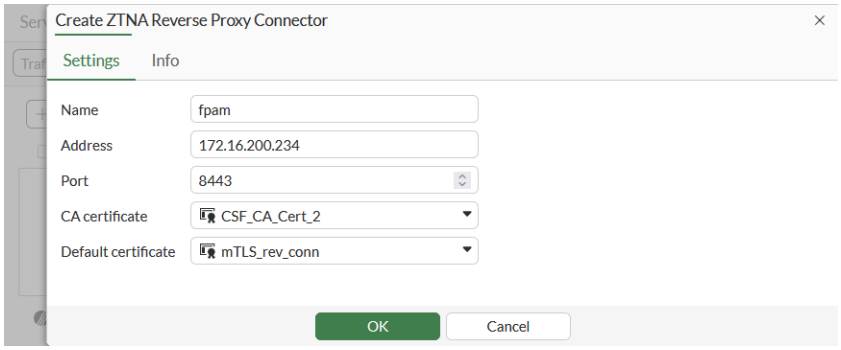
In the example topology shown in the introduction, we will configure a TCP forwarding destination on the FortiGate service connector to allow RDP connection to the Server (172.18.62.213). We will also configure a ZTNA reverse-connector to connect to the upstream FortiPAM.

The FortiPAM will follow the FortiPAM server (ZTNA Edge) configuration as illustrated in the Configuring FortiPAM/FortiGate as reverse gateway example. We will configure a secret on the FortiPAM server to allow users to RDP into the Server.

Alternatively, if deploying using a FortiProxy on the ZTNA Edge, follow the FortiProxy as reverse proxy to service connector example in the FortiProxy admin guide.

To configuring the reverse proxy connection in the FortiOS GUI:

1. Go to *System* > *Feature Visibility* and enable *Zero Trust Network Access* and *ZTNA Reverse Proxy Connector*.
2. Go to *Policy & Objects* > *ZTNA* and select the *Traffic Forwarding* tab.
3. Select the *Reverse Proxy Connector* sub-menu and click *Create New*.
4. Configure the following:



Field	Description
Name	Fpam
Address	172.16.200.234
Port	8443
CA certificate	CSF_CA_Cert_2
Default certificate	mTLS_rev_conn

5. Click *OK*

To configure the ZTNA reverse-connector in the FortiOS CLI:

```
config ztna reverse-connector
    edit "fpam"
        set address "172.16.200.234"
        set port 8443
        set certificate "mTLS_rev_conn"
        set trusted-server-ca "CSF_CA_Cert__2"
    next
end
```

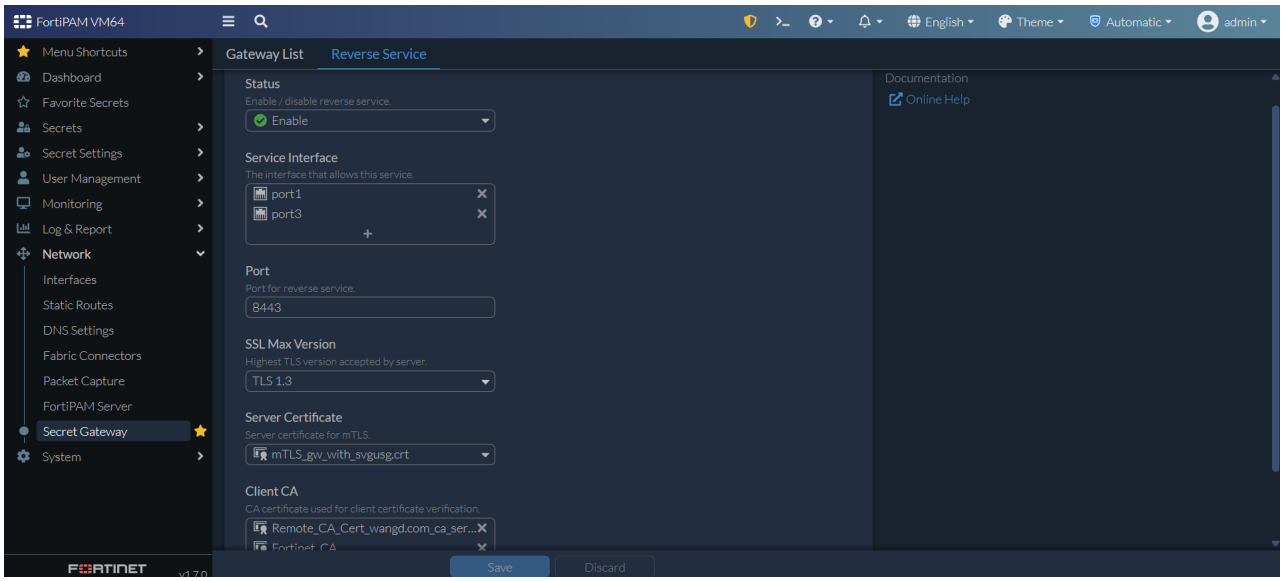
To configure the TCP forwarding proxy, destination, and rules in the FortiOS CLI:

```
config firewall vip
    edit "vip"
        set type access-proxy
        set server-type https
        set extip 10.1.100.2
        set extintf "port2"
        set client-cert disable
        set extport 4443
        set ssl-certificate "Fortinet_SSL"
    next
end
config ztna traffic-forward-proxy
    edit "ztfp"
        set vip "vip"
    next
end
config firewall address
    edit "to_server_214"
        set subnet 172.18.62.214 255.255.255.255
    next
end
config ztna destination
    edit "RDP_214"
        set address "to_server_214"
        set mappedport 3389
    next
```

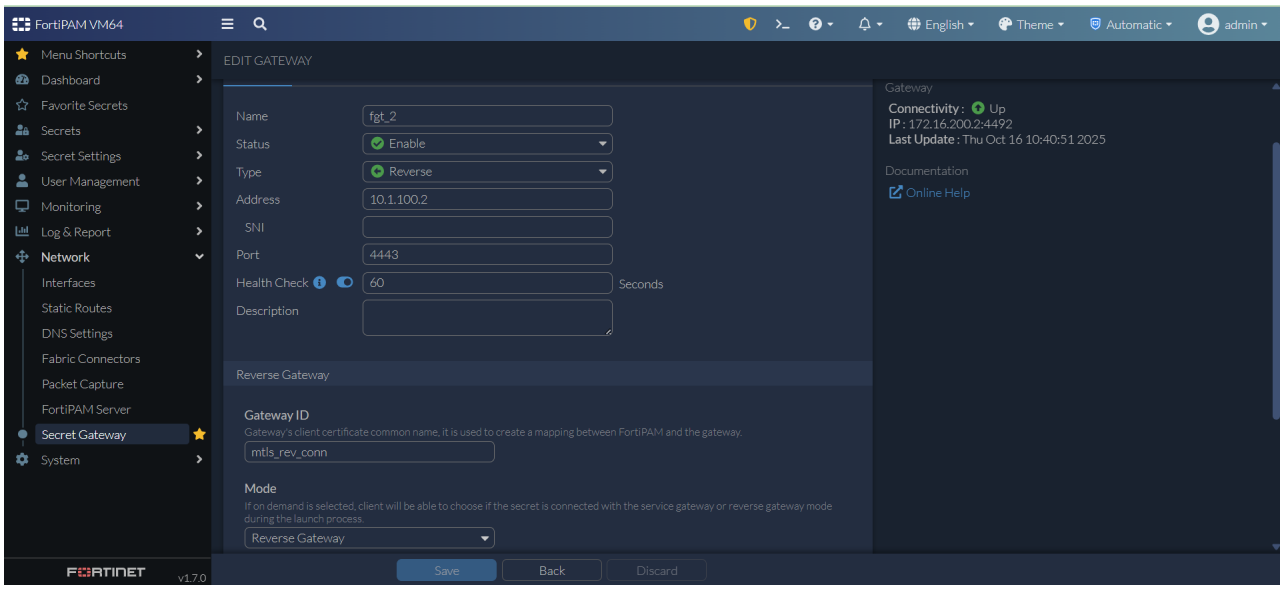
```
end
config firewall proxy-policy
    edit 1
        set proxy ztna-proxy
        set ztna-proxy "ztfp"
        set ztna-destination "RDP_214"
        set srcintf "any"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set logtraffic all
        set srcaddr6 "all"
        set dstaddr6 "all"
        set utm-status enable
        set profile-protocol-options "protocol"
    next
end
```

To configure the FortiPAM ZTNA Edge:

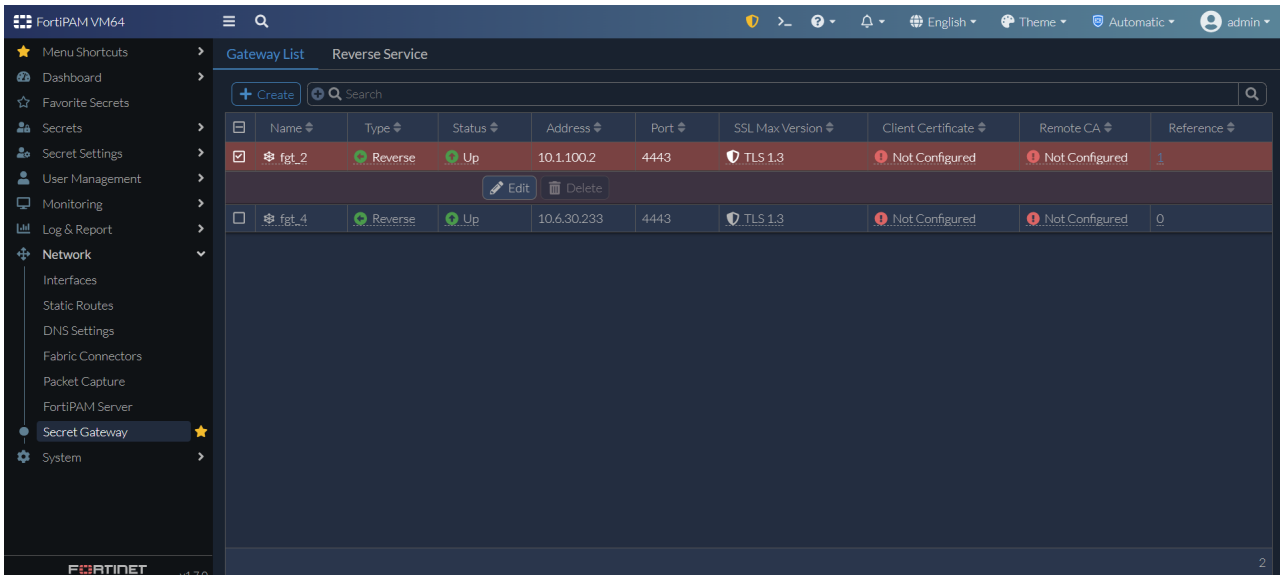
1. On the FortiPAM, go to *Network* > *Secret Gateway* > *Reverse Service*.
2. Enable the service and enter the server settings.



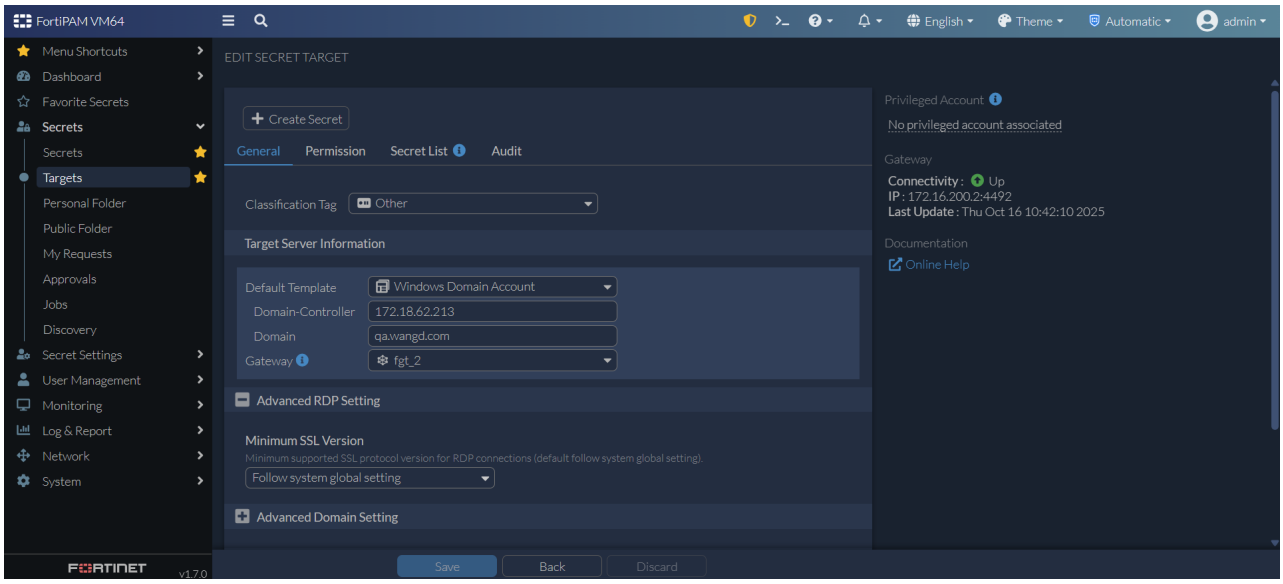
3. Click *Save*.
4. Under *Network* > *Secret Gateway*, create a new *Gateway*.
5. Enter the settings below and input the FortiGate VIP address to identify the desired gateway.
   For the *Gateway ID*, enter the CN string from the FortiGate client certificate (mTLS_rev_conn in this example).



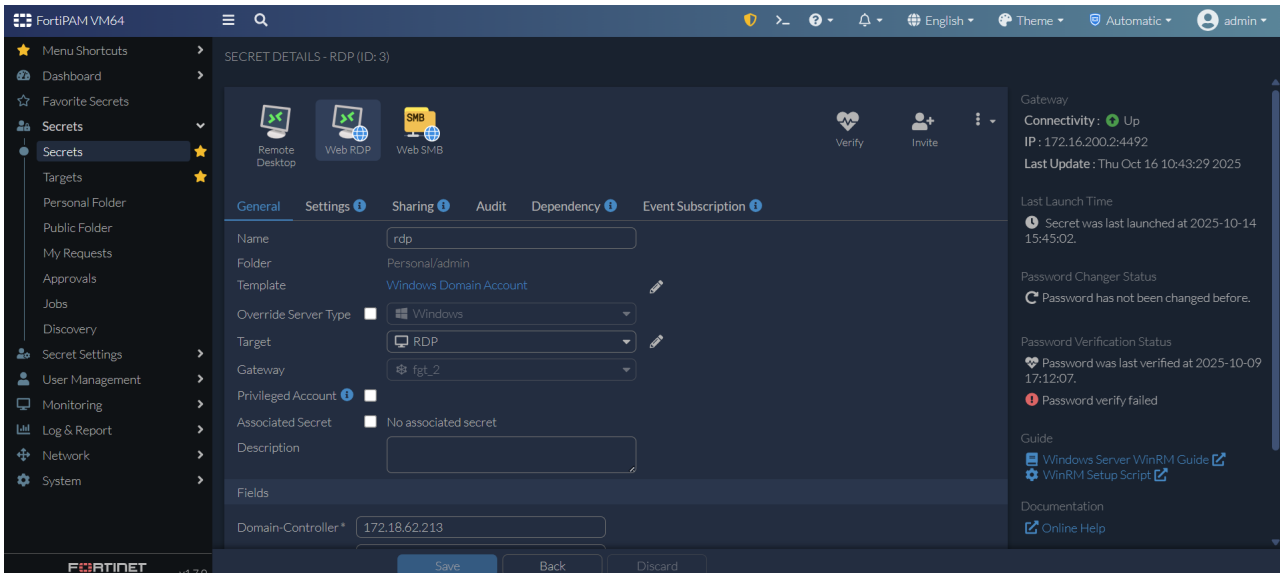
6. Save and view the *Gateway* from the *Gateway List*.



7. Go to *Secrets* > *Targets* and define a *Target* using this *Gateway*. Enter the address of the *Server* as the *Domain-Controller* and select the *Gateway* that was created earlier.



8. Go to *Secrets* > *Secrets* and define a *Secret* using this *Target*.

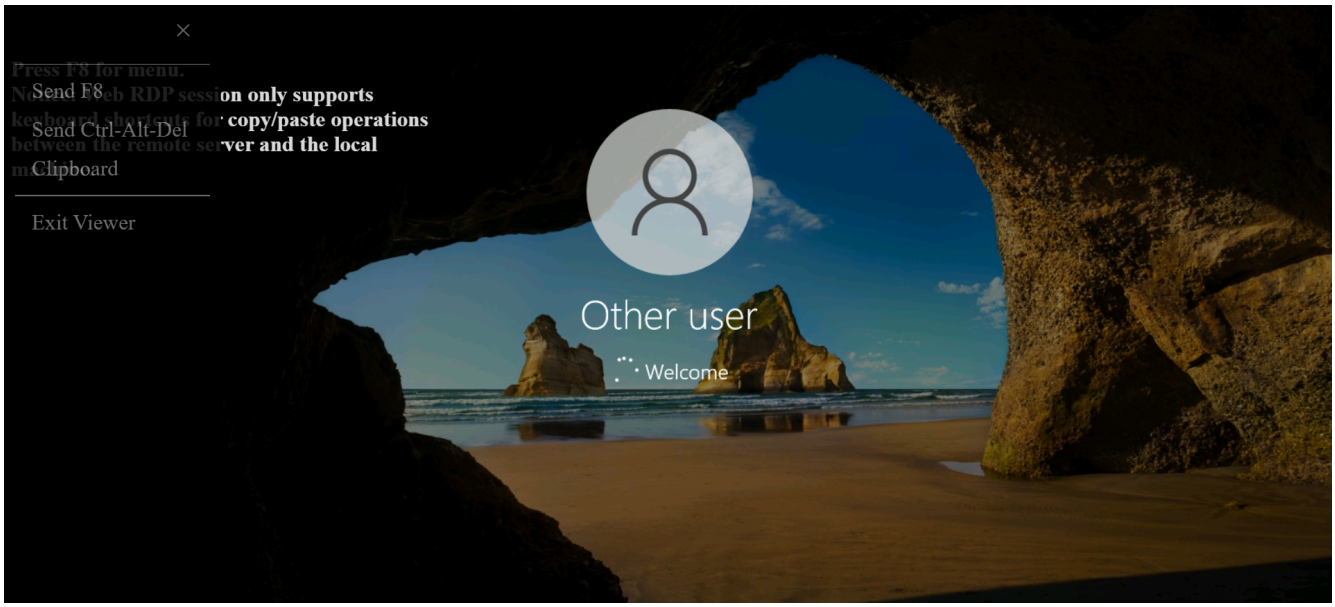


9. Save the settings.

To access the server through RDP:

On the Windows Client, you can login to the FortiPAM as a user, then access a secret from the FortiPAM portal. The client itself will not directly connect to the server itself. In fact, the FortiPAM will not initiate a connection to the server or the ZTNA service connector. It is the FortiGate ZTNA service connector which initiates the control channel and data channel to the FortiPAM to facilicate the communications.

On the client, use the Web RDP button from the FortiPAM GUI to RDP to the server.



Security posture and EMS connectors

This section includes information about tag and EMS connector related new features:

- Support tags in dual stack IPv4/IPv6 ZTNA policies on page 242

Support tags in dual stack IPv4/IPv6 ZTNA policies

IPv6 security posture tags and groups can be applied to dual stack IPv4/IPv6 ZTNA policies.

FortiClient and FortiClient EMS 7.4.5 or 8.0.0 or later are required to support this feature.

A new option is available to set the address sub-type to `ems-tag`:

```
config firewall address6
    edit 1
        set type dynamic
        set sub-type {sdn | ems-tag | 8021x}
        ...
    next
end
```

Option	Description
`sub-type`	Set the sub-type for the address: • `sdn`: SDN address. • `ems-tag`: FortiClient EMS tag.

Option	Description
	• `8021x`: 802.1x address Available when `type` is set to `dynamic`.

A new parameter is available to identify IPv6 security posture tags from FortiClient EMS:

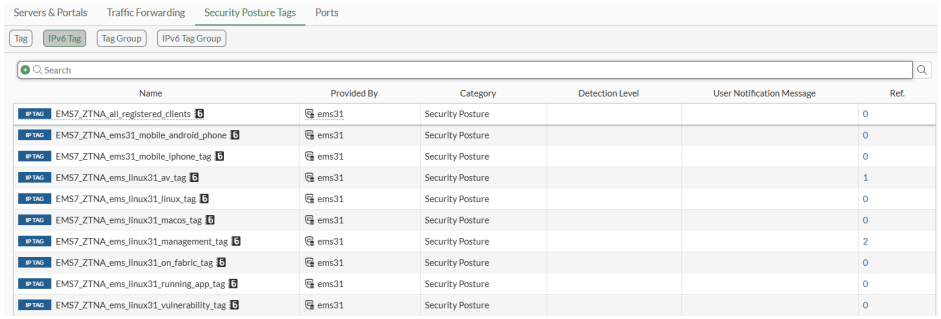
```
config firewall policy
    edit 1
        ...
        set ztna-status enable
        set ztna-ems-tag6 <name>
        ...
    next
end
```

Option	Description
`ztna-ems-tag6 <name>`	Specify the name of the IPv6 security posture tag. Available when `ztna-status` is set to `enable`.

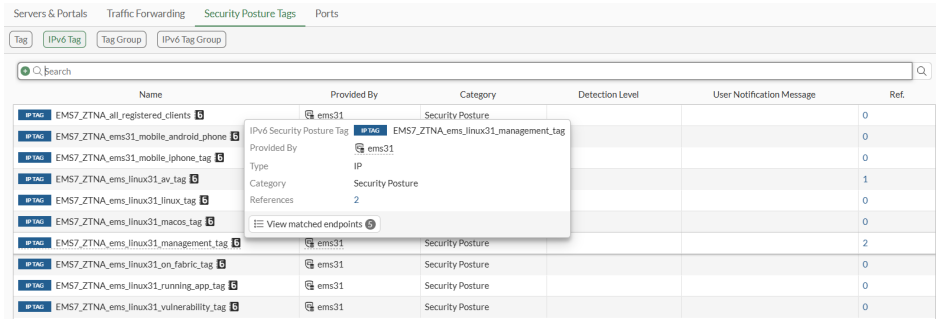
Example

To add IPv6 security posture tags to a firewall policy in the GUI:

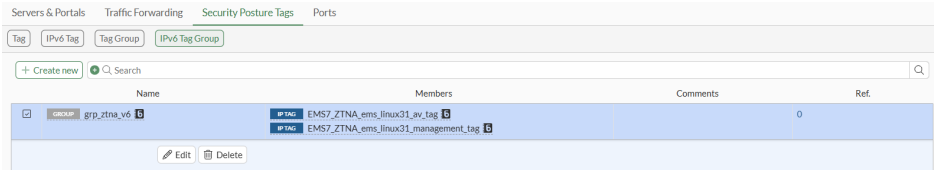
1. Go to *Policy & Objects* > *ZTNA* > *Security Posture Tags* > *IPv6 Tag* to view a list of security posture tags for IPv6.



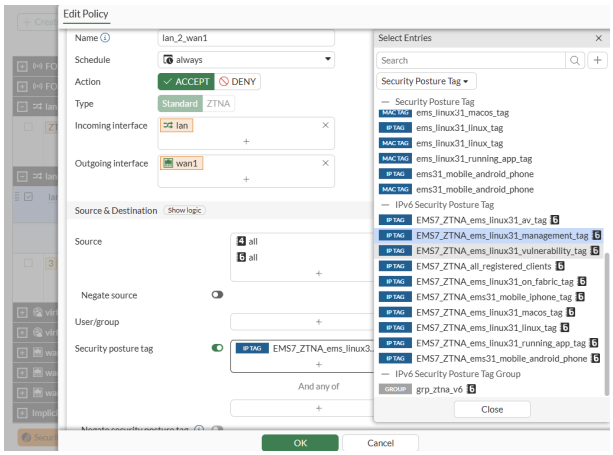
Hover over a tag to display a tooltip of information.



2. Click *IPv6 Tag Group* to view a group of IPv6 tags.



3. Go to *Policy & Objects* > *Firewall Policy* and open for editing a firewall policy with Source configured to support IPv4 and IPv6.
4. Enable *Security posture tag* and select a security posture tag for IPv6.



5. Click *OK* to save the policy.

To add IPv6 security posture tags to a firewall policy in the CLI:

1. Enable the `sub-type ems-tag` option for an IPv6 security posture tag:

   In this example, `sub-type ems-tag` is set for the `EMS7_ZTNA_ems_linux31_management_tag` IPv6 security posture tag.

```
config firewall address6
    edit "EMS7_ZTNA_ems_linux31_management_tag"
        set uuid 880ef31c-fcba-51f0-e348-1f01acd82c22
        set type dynamic
        set sub-type ems-tag
        set dirty clean
        set obj-tag "ems_linux31_management_tag"
        set tag-type "zero_trust"
    next
end
```

2. Select the IPv6 security posture tag in a firewall policy configured to support IPv4 and IPv6 sources:

   In this example `ztna-ems-tag6` is set to the `EMS7_ZTNA_ems_linux31_management_tag` IPv6 tag.

```
config firewall policy
    edit 1
        set name "lan_2_wan1"
        set uuid de52dcac-fc17-51ee-6a37-645e65221a24
        set srcintf "lan"
        set dstintf "wan1"
```

```
        set action accept
        set ztna-status enable
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set ztna-ems-tag6 "EMS7_ZTNA_ems_linux31_management_tag"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
    next
end
```

Security profiles

This section includes information about security profile related new features:

- Others on page 246
- Protecting LLM and GenAI on page 286

Others

This section includes information about other security profile related new features:

- File filter warning action for HTTP downloads on page 246
- Local URL filtering with custom FortiGuard categories on page 250
- New classification framework for application signatures and FortiView support on page 256
- Support for hybrid PQC SSL deep inspection in proxy mode on page 272
- Enhanced UTM protection with WebSocket traffic inspection on page 275
- FortiGuard-free rating for local and external categories on page 282
- Expanded CASB SaaS application database on page 284

File filter warning action for HTTP downloads

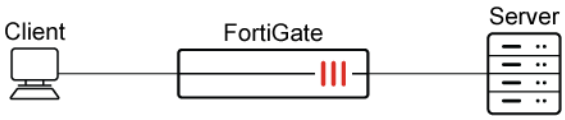
This information is also available in the FortiOS 8.0 Administration Guide:

- Configuring a file filter profile with the warning action

A warning action has been added to the file filter profile, allowing FortiGate to show a user authorization page before permitting downloads of matched file types over HTTP/HTTPS in proxy inspection mode. After the user acknowledges the warning, the content is allowed and access is logged. This behavior applies only to incoming HTTP traffic (eg. HTTP GET); outgoing HTTP traffic (eg. HTTP POST) and non-HTTP protocols are log-only, with no warning or blocking when applied to the same profile.

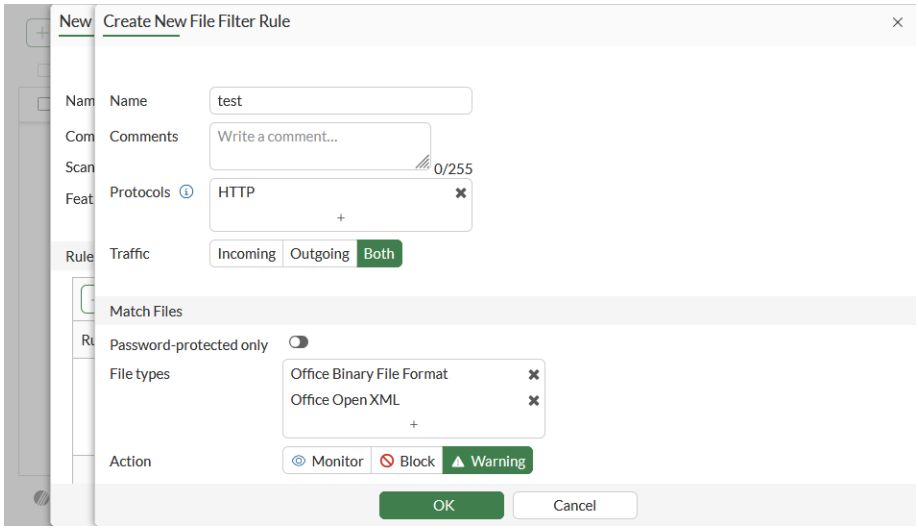
```
config file-filter profile
    edit <name>
        set feature-set proxy
        config rules
            edit <name>
                set action warning
            next
        end
    next
end
```

Example

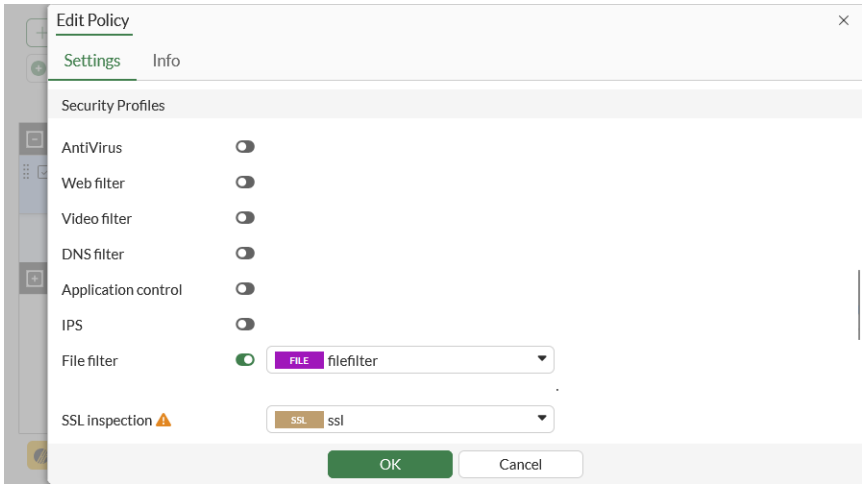


To configure and use a file filter profile with the warning action in the GUI:

1. Go to *Security Profiles* > *File Filter* and click *Create New*.
2. Set *Feature set* to *Proxy-based*.
3. In the *Rules* table, click *Create New*.
4. Set *Action* to *Warning*.
5. Configure the remaining settings as needed, then click *OK*.



6. Configure the remaining filter profile settings as needed, then click *OK*.
7. Go to *Policy & Objects* > *Firewall Policy* and create a new policy or edit an existing one.
8. In the *Security Profiles* section, enable *File filter* and select the file filter profile.



9. Configure the remaining settings as required, then click *OK*.

To configure and use a file filter profile with the warning action in the CLI:

1. Configure a file filter profile with the warning action:

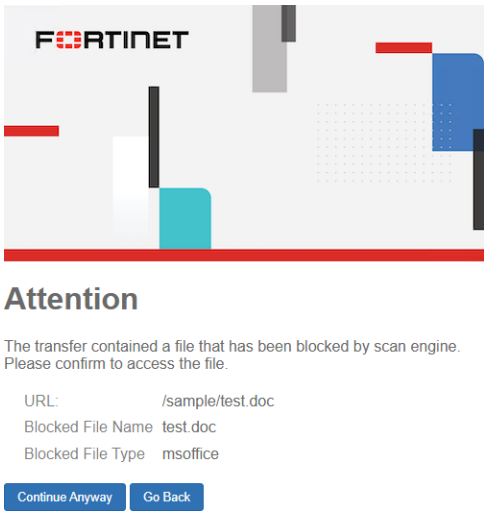
```
config file-filter profile
  edit "filefilter"
        set feature-set proxy
        config rules
            edit "test"
                set protocol http
                set action warning
                set file-type "msoffice" "msofficex"
            next
        end
    next
end
```

2. Apply the file filter profile to a firewall policy:

```
config firewall policy
    edit 1
        set name "filefilter"
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "ssl"
        set file-filter-profile "filefilter"
        set logtraffic all
        set nat enable
    next
end
```

To test the file filter:

1. On a Windows client, access an absolute URL (http://<iP address>/sample/test.doc) to download file *test.doc*.
   A replacement message is shown that asks the user to authorize or cancel the file download, and a blocking UTM file filter log is generated.

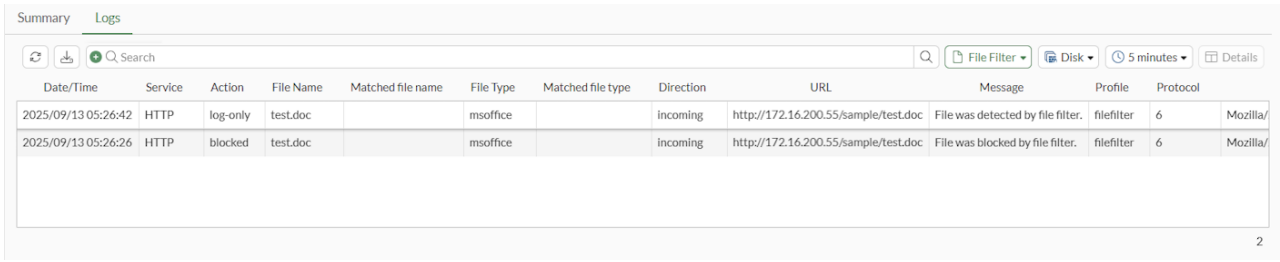



```
2: date=2025-09-13 time=05:26:26 eventtime=1757697986602674439 tz="+1200" logid="1900064000"
type="utm" subtype="file-filter" eventtype="file-filter" level="warning" vd="vdom1" policyid=1
poluuid="61c636d2-5b59-51f0-b555-76579a0b21fa" policytype="policy" sessionid=1583
srcip=10.1.100.195 srcport=51426 srccountry="Reserved" srcintf="port1" srcintfrole="undefined"
srcuuid="1027adc4-5b59-51f0-7577-e6bcd1ba3541" dstip=172.16.200.55 dstport=80
dstcountry="Reserved" dstintf="port3" dstintfrole="undefined" dstuuid="1027adc4-5b59-51f0-
7577-e6bcd1ba3541" proto=6 service="HTTP" profile="filefilter" direction="incoming"
action="blocked" url="http://172.16.200.55/sample/test.doc" hostname="172.16.200.55"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/140.0.0.0 Safari/537.36" httpmethod="GET" rulename="test" filename="test.doc"
filesize=19456 filetype="msoffice" msg="File was blocked by file filter."
```

2. Click *Continue Anyway* to download the file. A log-only file filter log is generated:

```
1: date=2025-09-13 time=05:26:42 eventtime=1757698001607832698 tz="+1200" logid="1900064001"
type="utm" subtype="file-filter" eventtype="file-filter" level="notice" vd="vdom1" policyid=1
poluuid="61c636d2-5b59-51f0-b555-76579a0b21fa" policytype="policy" sessionid=1602
srcip=10.1.100.195 srcport=51432 srccountry="Reserved" srcintf="port1" srcintfrole="undefined"
srcuuid="1027adc4-5b59-51f0-7577-e6bcd1ba3541" dstip=172.16.200.55 dstport=80
dstcountry="Reserved" dstintf="port3" dstintfrole="undefined" dstuuid="1027adc4-5b59-51f0-
7577-e6bcd1ba3541" proto=6 service="HTTP" profile="filefilter" direction="incoming"
action="log-only" url="http://172.16.200.55/sample/test.doc" hostname="172.16.200.55"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/140.0.0.0 Safari/537.36" httpmethod="GET"
referralurl="http://172.16.200.55/sample/test.doc" rulename="test" filename="test.doc"
filesize=19456 filetype="msoffice" msg="File was detected by file filter."
```

3. Go to *Log & Report* > *Security Events* and view the *File Filter* logs.



Local URL filtering with custom FortiGuard categories

A new `type` option has been added for local URL filter tables to enhance URL filtering flexibility. This allows the creation of custom FortiGuard categories using the local URL filter table, applicable in both policy-based and profile-based NGFW modes.

```
config webfilter urlfilter
    edit <id>
        set type {profile | category}
    next
end
```

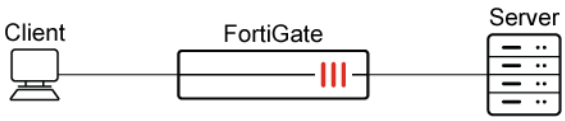
Option	Description
`category`	URL list for local web category override.
`profile`	URL list for webfilter profile.

A new `urlfilter-table` option has been added to the FortiGuard local category configuration, enabling users to associate a list of URLs, defined in a local URL filter table with type category, with a FortiGuard local category.

```
config webfilter ftgd-local-cat
    edit <desc>
        set urlfilter-table <urlfilter-table>
    next
end
```

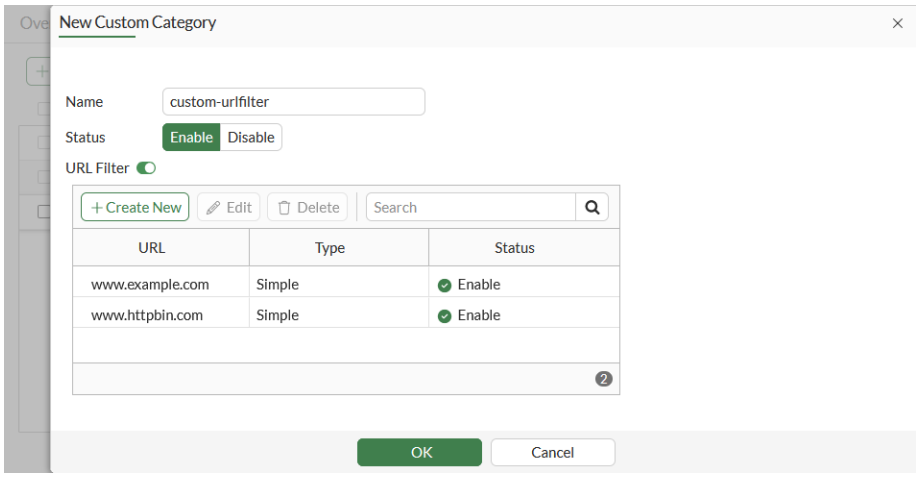
After a local URL filter table is configured with type category it cannot be configure for regular URL filtering. In policy-based NGFW mode, these local categories can be referenced directly in the url-category field of a security policy. In profile-based NGFW mode, the local categories can be used to define category-based filtering actions in a web filter profile.

Examples



To configure a custom URL filter in the GUI:

1. Go to *Security Profiles* > *Web Rating Overrides*.
2. Select the *Custom Categories* tab and click *Create New*.
3. Enter a name for the custom category, such as *custom-urlfilter*.
4. Enable URL Filter and click *Create New* in the table.
5. Enter a URL, such as *www.example.com*, then click *OK*.
6. Click *Create New* in the table again and add another URL, such as *www.httpbin.com*, then click *OK*.



7. Click *OK*.

To configure a custom URL filter in the CLI:

1. Configure the URL filter list:

```
config webfilter urlfilter
    edit 1
        set name "test"
        set comment ''
        set type category
        config entries
            edit 1
                set url "www.example.com"
                set type simple
                set status enable
                set comment ''
            next
            edit 2
                set url "www.httpbin.org"
                set type simple
                set status enable
                set comment ''
            next
        end
    next
end
```

2. Configure FortiGuard Web Filter local categories, using the `test` URL filter list:

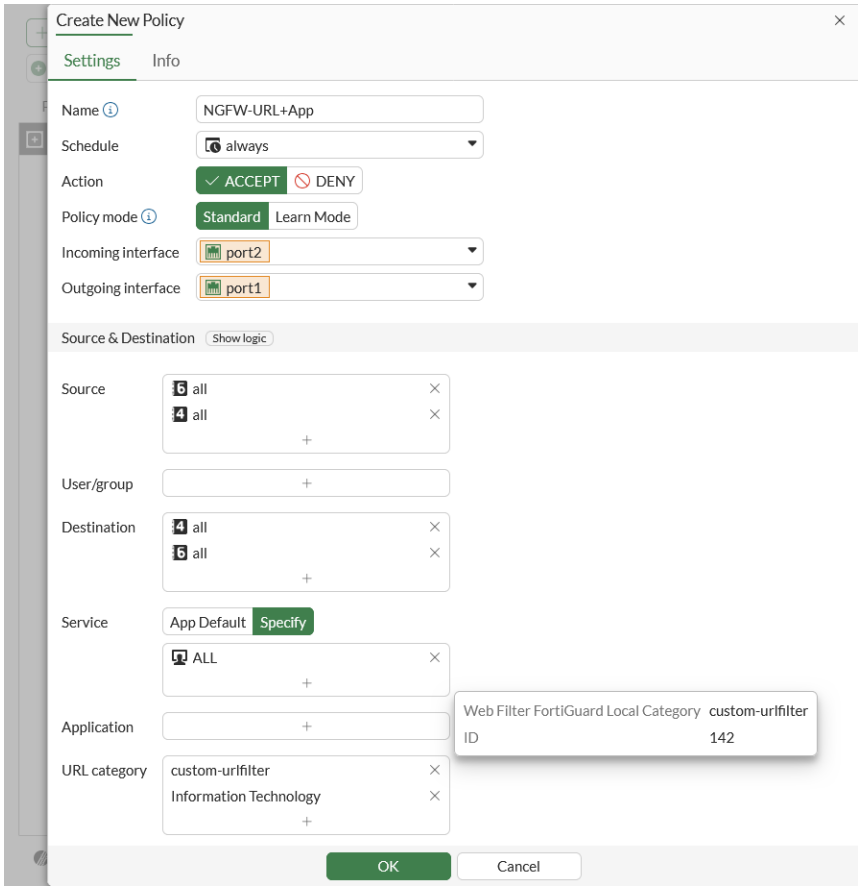
```
config webfilter ftgd-local-cat
    edit "custom-urlfilter"
        set id 142
        set urlfilter-table 1
    next
end
```

Case 1: Policy-based NGFW mode

A security policy is configured that uses the URL filter table based FortiGuard local category.

To configure the security policy in the GUI:

1. Go to *Policy & Objects* > *Security Policy* and click *Create New*.
2. Configure the policy as needed, using the *custom-urlfilter* in the *URL category*.



3. Click *OK*.

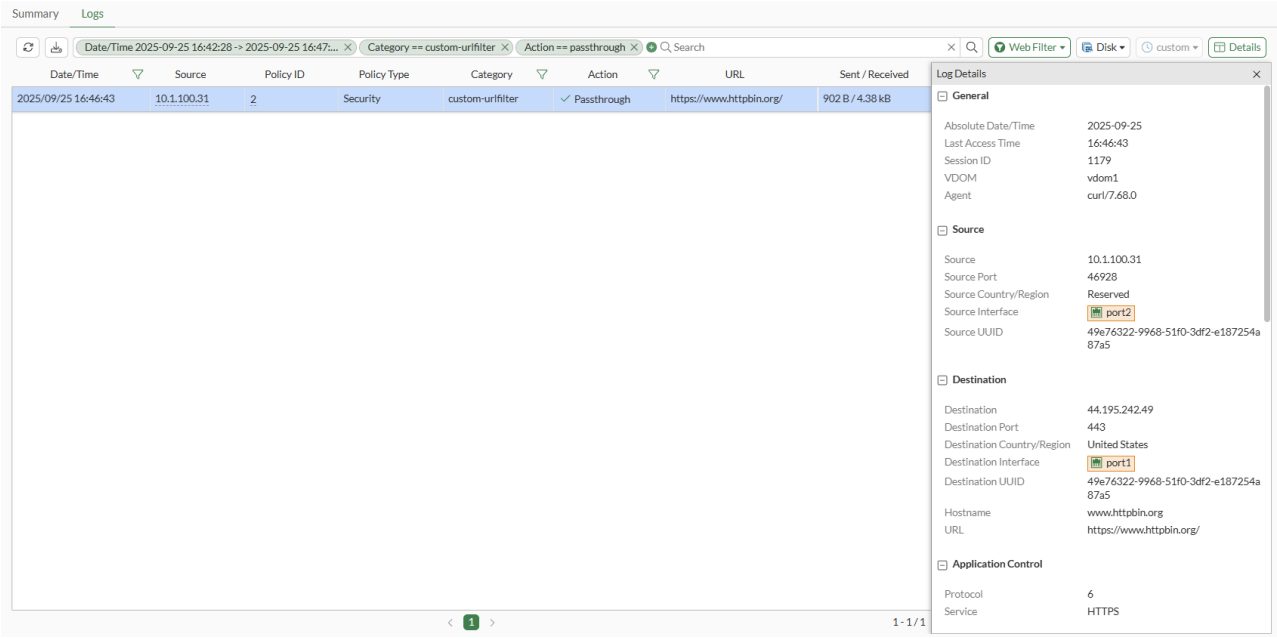
To configure the security policy in the CLI:

```
config firewall security-policy
    edit 1
        set name "NGFW-URL+App"
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set enforce-default-app-port disable
        set service "ALL"
```

```
        set action accept
        set schedule "always"
        set logtraffic all
        set url-category 52 142
    next
end
```

To test the filter:

1. On the client, access *www.httpbin.org*. It matches the security policy and access is allowed.
2. Check the logs:



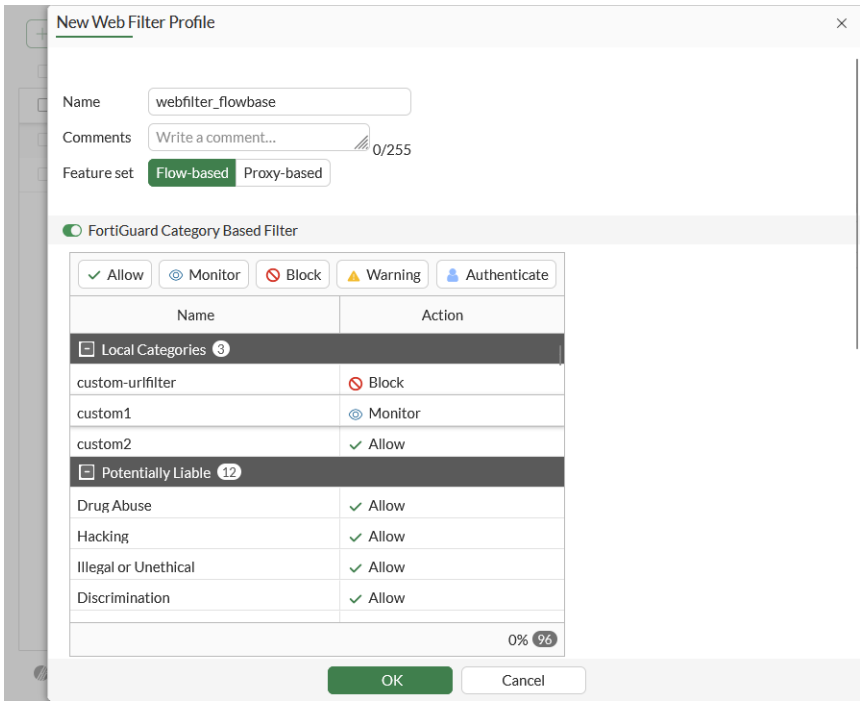

```
1: date=2025-09-25 time=16:46:43 eventtime=1758844003654750138 tz="-0700" logid="0317013312"
type="utm" subtype="webfilter" eventtype="ftgd_allow" level="notice" vd="vdom1" policyid=2
poluuid="13b292a2-9a41-51f0-d0bd-4c0ee324dd45" policytype="security-policy" sessionid=1179
srcip=10.1.100.31 srcport=46928 srccountry="Reserved" srcintf="port2" srcintfrole="undefined"
srcuuid="49e76322-9968-51f0-3df2-e187254a87a5" dstip=44.195.242.49 dstport=443
dstcountry="United States" dstintf="port1" dstintfrole="undefined" dstuuid="49e76322-9968-
51f0-3df2-e187254a87a5" proto=6 httpmethod="GET" service="HTTPS" hostname="www.httpbin.org"
agent="curl/7.68.0" action="passthrough" reqtype="direct" url="https://www.httpbin.org/"
sentbyte=902 rcvdbyte=4376 direction="outgoing" msg="URL belongs to an allowed category in
policy" cat=142 catdesc="custom-urlfilter" ratemethod="domain"
```

Case 2: Profile-based NGFW mode

A web filter profile is configured that uses the URL filter table based FortiGuard local category.

To configure a web filter profile in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*.
2. In the *FortiGuard Category Based Filter* table, in the *Local Categories* category, set the *Action* for *custom-urlfilter* to *Block*.
3. Configure the remaining settings as needed.



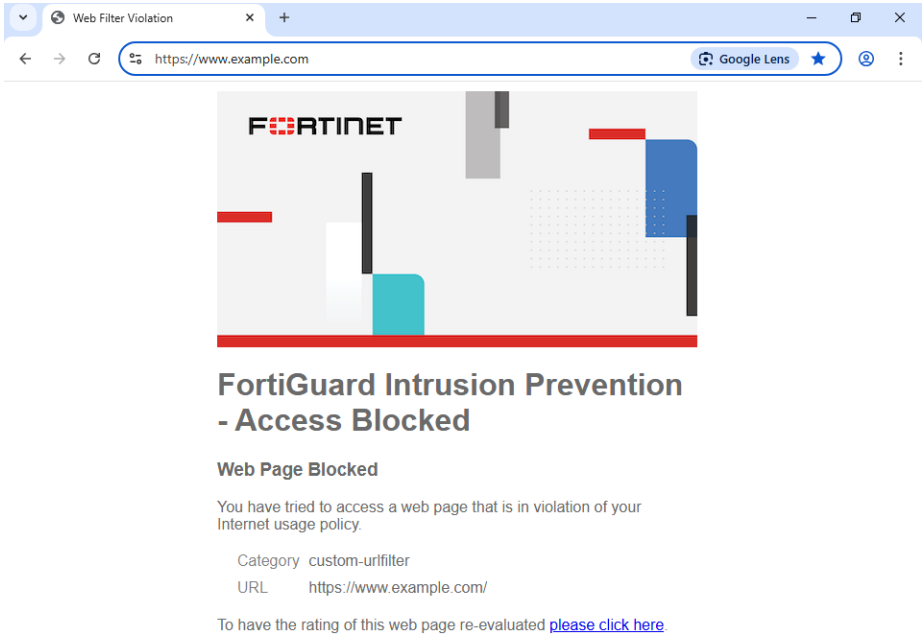
4. Click *OK*.

To configure a web filter profile in the CLI:

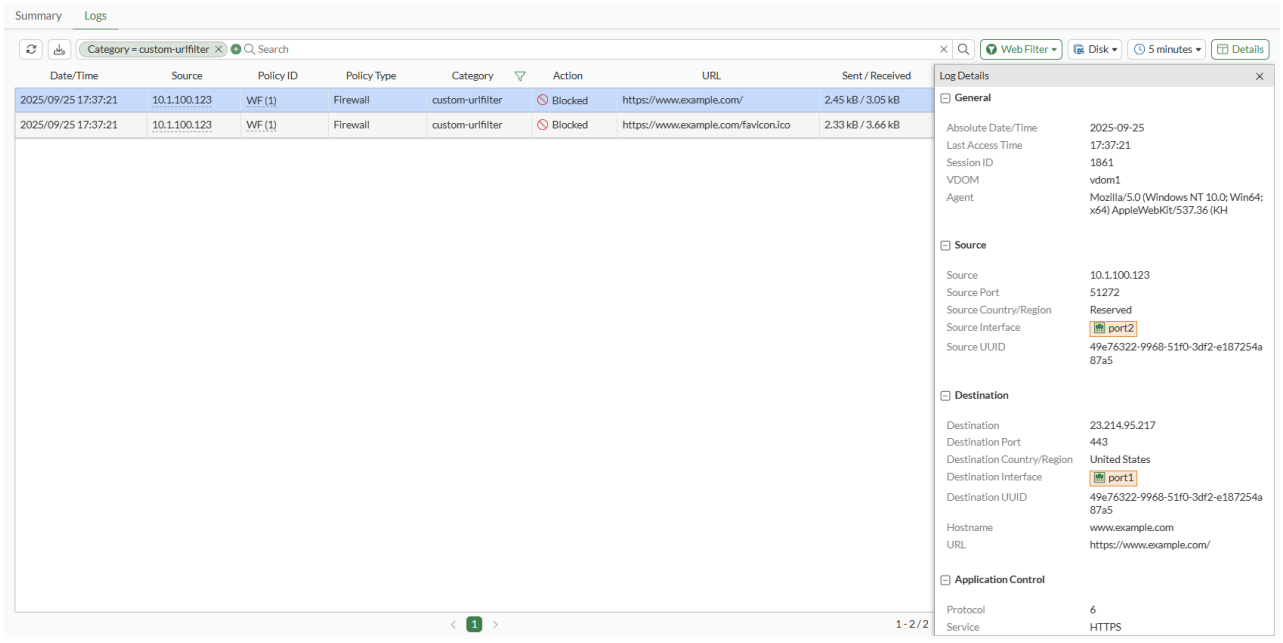
```
config webfilter profile
    edit "webfilter_flowbase"
        config ftgd-wf
            unset options
            config filters
                ...
                edit 31
                    set category 142
                    set action block
                    set log disable
                next
                ...
            end
        end
    next
end
```

To test the filter:

1. On the client, access *www.example.com*. It matches the FortiGuard local category, so access is blocked and a replacement message is shown.



2. Check the logs:




```
1: date=2025-09-25 time=17:37:21 eventtime=1758847041142331041 tz="-0700" logid="0316013056"
type="utm" subtype="webfilter" eventtype="ftgd_blk" level="warning" vd="vdom1" policyid=1
poluuid="fb4996a2-9a6d-51f0-5dfd-5049690d8232" policytype="policy" sessionid=1861
srcip=10.1.100.123 srcport=51272 srccountry="Reserved" srcintf="port2" srcintfrole="undefined"
srcuuid="49e76322-9968-51f0-3df2-e187254a87a5" dstip=23.214.95.217 dstport=443
dstcountry="United States" dstintf="port1" dstintfrole="undefined" dstuuid="49e76322-9968-
```

```
51f0-3df2-e187254a87a5" proto=6 httpmethod="GET" service="HTTPS" hostname="www.example.com"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH" profile="webfilter_
flowbase" action="blocked" reqtype="direct" url="https://www.example.com/" sentbyte=2448
rcvdbyte=3054 direction="outgoing" msg="URL belongs to a denied category in policy" cat=142
catdesc="custom-urlfilter" ratemethod="domain"
```

New classification framework for application signatures and FortiView support

Administrators can now mark applications or categories of applications as sanctioned or unsanctioned. By default, FortiOS automatically labels applications without explicit classification as unclassified, although you can change the implicit rule to treat unclassified applications as sanctioned or unsanctioned. This change provides clearer visibility in logs and improves monitoring and management of application usage.

A classification application filter can also be defined in application control profiles to allow or block sanctioned or unsanctioned applications. Application control logs display which apps are allowed and blocked.

On FortiView, you can view the number of unsanctioned and sanctioned applications, the number of applications in each risk level, and the number of applications in each application category. You can also change application status between sanctioned and unsanctioned and block or allow applications by creating override rules.

CLI syntax

A new CLI command is available to configure applications or categories of applications as sanctioned or unsanctioned:

```
config application unsanctioned-apps
    edit 1
        set type {app | category}
        set *app <ID>
        set *category <ID>
        set status {sanctioned | unsanctioned}
   next
end
```

Option	Description
`set type {app | category}`	Set type: • `app`: application • `category`: category of applications
`set *app <ID>`	Enter the application ID. Available when `type` is set to `app`.
`set *category <ID>`	Enter the category ID. Available when `type` is set to `category`.
`set status {sanctioned | unsanctioned}`	Set status: • `sanctioned`: mark as sanctioned. • `unsanctioned`: mark as unsanctioned.

A new CLI command is available to configure the default classification setting for applications:

```
config application classification-settings
    set default-app-classification {unclassified | unsanctioned | sanctioned}
end
```

Option	Description
`set default-app-classification {unclassified | unsanctioned | sanctioned}`	Set default classification setting for applications (default = unclassified): • `unclassified` • `unsanctioned` • `sanctioned`

A new CLI parameter is available to define an application classification filter:

```
config application list
    edit <name>
        config entries
            edit <ID>
                set classification [none | sanctioned | unsanctioned | unclassified]
                set action pass
            next
        end
    next
end
```

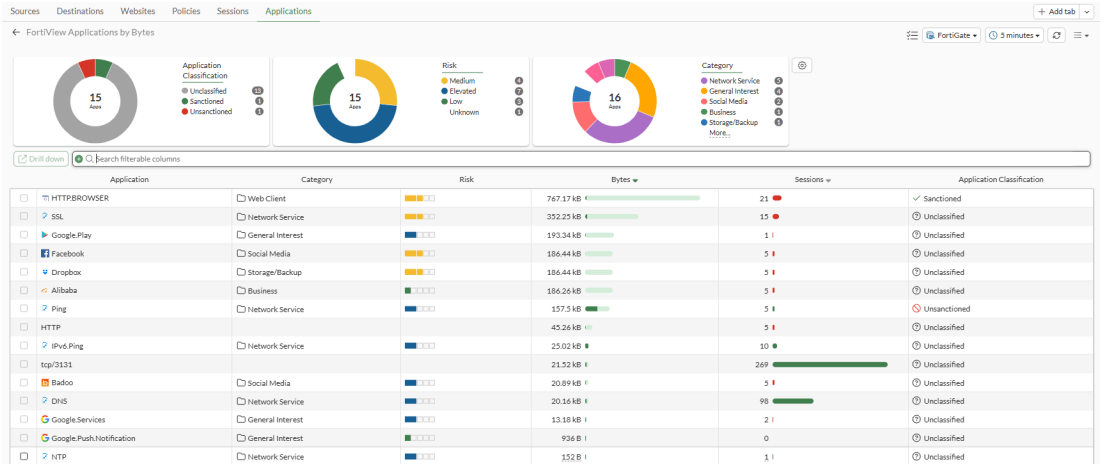
Option	Description
`classification [none | sanctioned | unsanctioned | unclassified]`	Specify an application classification filter (default = none): • `none`: No classification filter. • `sanctioned`: Pass or allow sanctioned applications. • `unsanctioned`: Pass or allow unsanctioned applications. • `unclassified`: Pass or allow unclassified applications.

GUI

The following FortiView monitors provide a way to view classified (sanctioned or unsanctioned) applications:

- FortiView > Applications
- FortiView > Proxy Applications
- FortiView > AI Applications
- FortiView > AI Use Cases

Following is an example of the *FortiView > Applications* monitor:



The FortiView monitors now use pie charts instead of graphs to show:

- *Applications classification*: Number of sanctioned, unsanctioned, and unclassified applications.
- *Risk*: Number of applications in each risk level.
- *Category*: Number of applications in each application category. (Does not apply to AI monitors.)

From FortiView, you can change applications status to unsanctioned or sanctioned, and you can block or allow applications by creating overriding rules. See Example 3: using FortiView on page 266 for details.

Example

This section includes the following examples to demonstrate the feature:

- Example 1: classifying applications on page 258
- Example 2: creating override filters in application control profiles on page 263
- Example 3: using FortiView on page 266

Example 1: classifying applications

In this example, an administrator marks the Facebook application, Social Media category, and implicit rule default classification as follows:

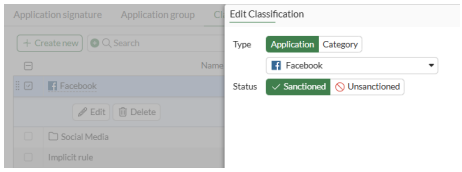
- Facebook application is marked as sanctioned.
- Social Media category is marked as unsanctioned.
- In the implicit rule, the default classification is set to unclassified.

After a user visits Facebook, Twitter, and YouTube websites, the signatures are triggered, and the application classification status is displayed in the FortiOS logs.

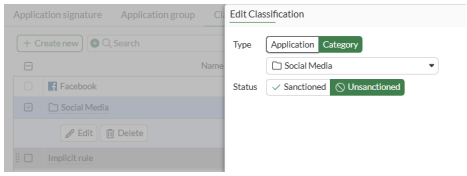
To classify applications in the GUI:

1. Go to *Security Profiles* > *Application Signatures* > *Classification*.
2. Create an application classification:
   a. Click *Create new*.
   b. Set *Type* to *Application*, and select *Facebook* application from the dropdown list.

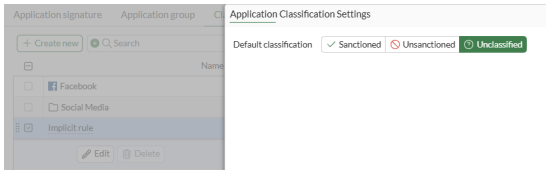
c. Set *Status* to *Sanctioned*.
d. Click *OK*.



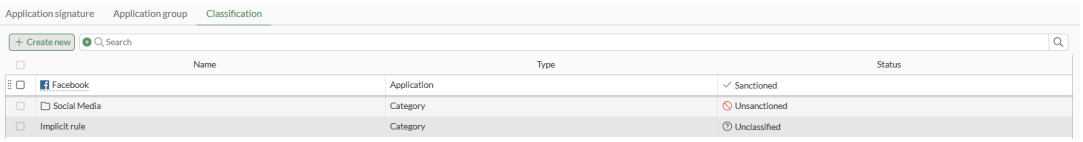
3. Create an application category classification:
   a. Click *Create new*.
   b. Set *Type* to *Category*, and select *Social Media* from the dropdown list.
   c. Set *Status* to *Unsanctioned*.
   d. Click *OK*.



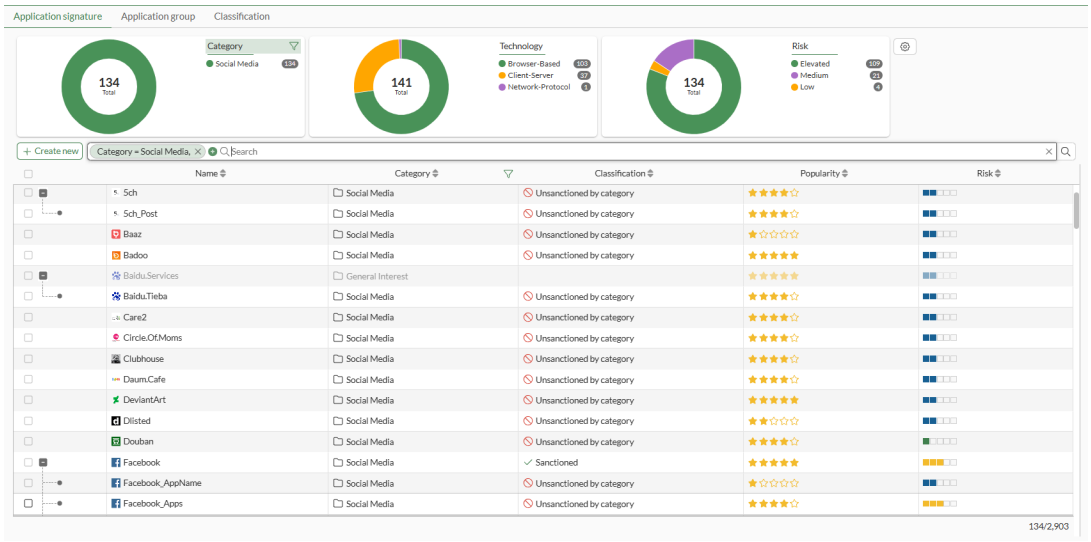
4. Edit Implicit rule entry:
   a. Select *Implicit rule*, and click *Edit*.
   b. Set *Default classification* to *Unclassified*.
   c. Click *OK*.



On the *Classification* page, the application (Facebook), category (Social Media), and default implicit rule entries are displayed in a prioritized list. The order determines which classification is applied first. Items at the top of the table take precedence, and administrators can drag and reorder entries to adjust the priority. The *Status* column displays the classification status.



5. Go to *Security Profiles* > *Application Signatures* > *Application signature*. The *Classification* column displays *Sanctioned* for Facebook and *Unsanctioned by category* for signatures in the *Social Media* category.



6. Go to *Security Profiles* > *Application Control*, and edit the default application profile to monitor all applications with action set to pass.
7. Go to *Policy & Objects* > *Firewall Policy*, and apply the default application profile to a firewall policy with a deep inspection ssl-ssh profile.

To classify applications in the CLI:

1. Apply sanctioned status to Facebook application (app 15832) and unsanctioned status to social media category (category 23):

```
config application unsanctioned-apps
    edit 1
        set type app
        set app 15832
        set status sanctioned
    next
    edit 2
        set type category
        set category 23
        set status unsanctioned
    next
end
```

2. Edit the default application profile to set the action to pass:

```
config application list
    edit "default"
        set comment "Monitor all applications."
        config entries
            edit 1
                set action pass
            next
        end
    next
end
```

3. Apply the default application profile to a firewall policy with a deep inspection ssl-ssh profile:

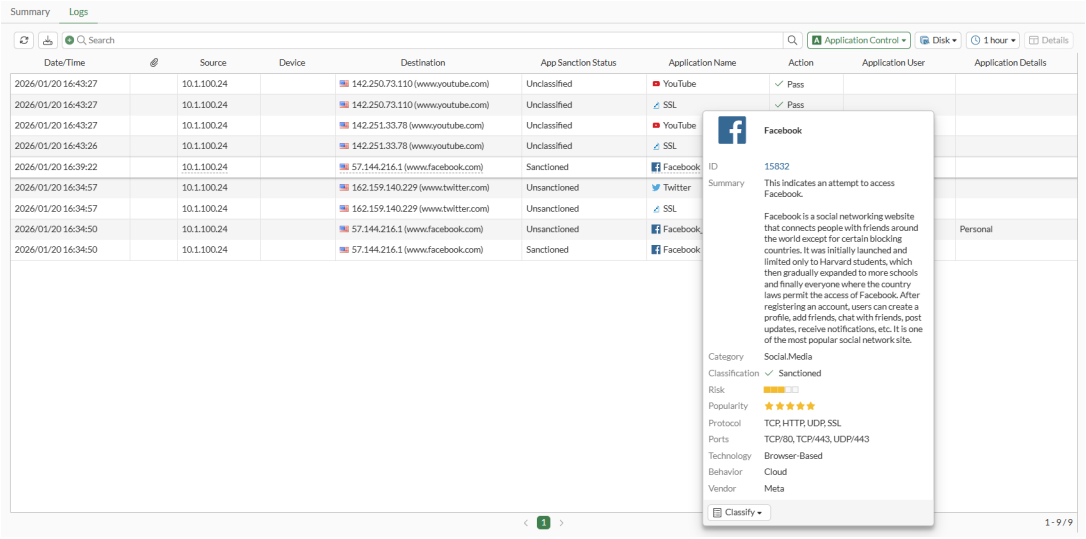
```
config firewall policy
    edit 1
        set uuid e47d62aa-f62f-51f0-91d4-78a59eb63616
        set srcintf "port18"
        set dstintf "port17"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "new-deep-inspection"
        set application-list "default"
        set nat enable
    next
end
```

To verify:

1. After a user has visited Facebook, Twitter, and YouTube websites to trigger the signatures, view the logs.
2. Go to *Log & Report* > *Security Events* > *Application Control Logs*.

   The *App Sanction Status* column displays the classification status. The classification status is also displayed in the signature tooltip.

   Logs do not show `appstatus` for `unclassified` type in the CLI.



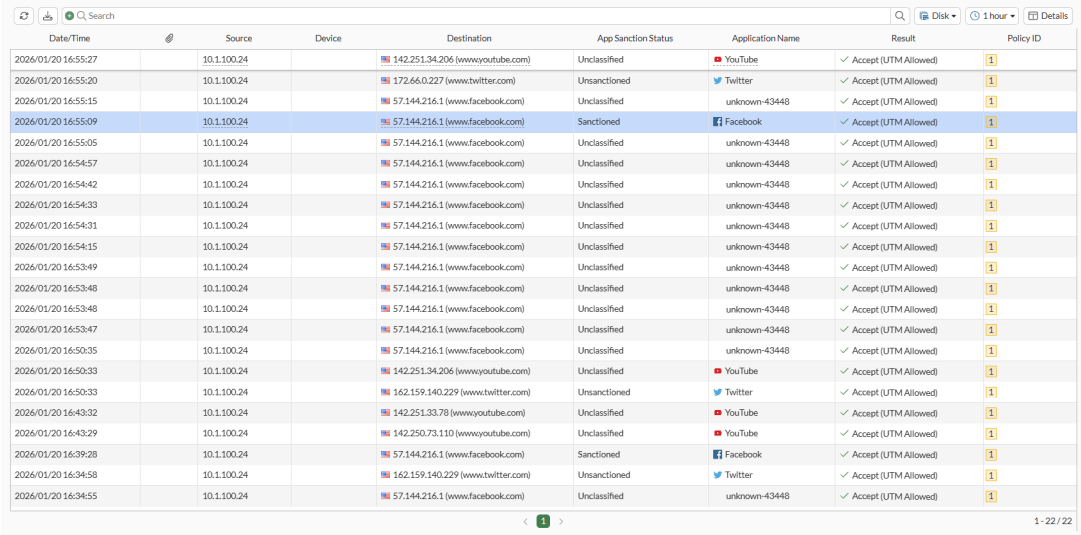
Sample application logs:

```
7: date=2026-01-20 time=16:55:10 eventtime=1768956909927343511 tz="-0800" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="vd1" appid=15832
srcip=10.1.100.24 srccountry="Reserved" dstip=57.144.216.1 dstcountry="United States"
```

```
srcport=56180 dstport=443 srcintf="port18" srcintfrole="undefined" dstintf="port17"
dstintfrole="undefined" proto=6 service="SSL" direction="outgoing" policyid=1
poluuid="e47d62aa-f62f-51f0-91d4-78a59eb63616" policytype="policy" sessionid=2015
applist="default" action="pass" appcat="Social.Media" app="Facebook" appstatus="sanctioned"
hostname="www.facebook.com" incidentserialno=93323270 url="/" msg="Social.Media: Facebook"
apprisk="medium" scertcname="*.facebook.com" scertissuer="DigiCert Global G2 TLS RSA SHA256
2020 CA1"

4: date=2026-01-20 time=16:55:19 eventtime=1768956919154052108 tz="-0800" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="vd1" appid=16001
srcip=10.1.100.24 srccountry="Reserved" dstip=172.66.0.227 dstcountry="United States"
srcport=48354 dstport=443 srcintf="port18" srcintfrole="undefined" dstintf="port17"
dstintfrole="undefined" proto=6 service="SSL" direction="outgoing" policyid=1
poluuid="e47d62aa-f62f-51f0-91d4-78a59eb63616" policytype="policy" sessionid=2018
applist="default" action="pass" appcat="Social.Media" app="Twitter" appstatus="unsanctioned"
hostname="www.twitter.com" incidentserialno=93323275 url="/" msg="Social.Media: Twitter"
apprisk="elevated" scertcname="twitter.com" scertissuer="E7"

1: date=2026-01-20 time=16:55:30 eventtime=1768956930752241455 tz="-0800" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="vd1" appid=31077
srcip=10.1.100.24 srccountry="Reserved" dstip=74.125.195.93 dstcountry="United States"
srcport=58128 dstport=443 srcintf="port18" srcintfrole="undefined" dstintf="port17"
dstintfrole="undefined" proto=6 service="SSL" direction="outgoing" policyid=1
poluuid="e47d62aa-f62f-51f0-91d4-78a59eb63616" policytype="policy" sessionid=2025
applist="default" action="pass" appcat="Video/Audio" app="YouTube" hostname="www.youtube.com"
incidentserialno=93323282 url="/" msg="Video/Audio: YouTube" apprisk="elevated"
scertcname="*.google.com" scertissuer="WR2"
```

```
ration=5 sentbyte=204 rcvdbyte=112 sentpkt=4 rcvdpkt=2 utmaction="allow" countapp=2
utmref=65534-524
```

In application and traffic logs, the *App Sanction Status* column displays *Sanctioned* for Facebook, *Unsanctioned* for Twitter, and *Unclassified* for YouTube.



Date/Time	Source	Device	Destination	App Sanction Status	Application Name	Result	Policy ID
2026/01/20 16:55:27	10.1.100.24		142.251.34.206 (www.youtube.com)	Unclassified	YouTube	Accept (UTM Allowed)	1
2026/01/20 16:55:20	10.1.100.24		172.66.0.227 (www.twitter.com)	Unsanctioned	Twitter	Accept (UTM Allowed)	1
2026/01/20 16:55:15	10.1.100.24		57.144.216.1 (www.facebook.com)	Unclassified	unknown-43448	Accept (UTM Allowed)	1
2026/01/20 16:55:09	10.1.100.24		57.144.216.1 (www.facebook.com)	Sanctioned	Facebook	Accept (UTM Allowed)	1
2026/01/20 16:55:05	10.1.100.24		57.144.216.1 (www.facebook.com)	Unclassified	unknown-43448	Accept (UTM Allowed)	1
2026/01/20 16:54:57	10.1.100.24		57.144.216.1 (www.facebook.com)	Unclassified	unknown-43448	Accept (UTM Allowed)	1
2026/01/20 16:54:42	10.1.100.24		57.144.216.1 (www.facebook.com)	Unclassified	unknown-43448	Accept (UTM Allowed)	1
2026/01/20 16:54:33	10.1.100.24		57.144.216.1 (www.facebook.com)	Unclassified	unknown-43448	Accept (UTM Allowed)	1
2026/01/20 16:54:31	10.1.100.24		57.144.216.1 (www.facebook.com)	Unclassified	unknown-43448	Accept (UTM Allowed)	1
2026/01/20 16:54:15	10.1.100.24		57.144.216.1 (www.facebook.com)	Unclassified	unknown-43448	Accept (UTM Allowed)	1
2026/01/20 16:53:49	10.1.100.24		57.144.216.1 (www.facebook.com)	Unclassified	unknown-43448	Accept (UTM Allowed)	1
2026/01/20 16:53:48	10.1.100.24		57.144.216.1 (www.facebook.com)	Unclassified	unknown-43448	Accept (UTM Allowed)	1
2026/01/20 16:53:48	10.1.100.24		57.144.216.1 (www.facebook.com)	Unclassified	unknown-43448	Accept (UTM Allowed)	1
2026/01/20 16:53:47	10.1.100.24		57.144.216.1 (www.facebook.com)	Unclassified	unknown-43448	Accept (UTM Allowed)	1
2026/01/20 16:50:35	10.1.100.24		57.144.216.1 (www.facebook.com)	Unclassified	unknown-43448	Accept (UTM Allowed)	1
2026/01/20 16:50:33	10.1.100.24		142.251.34.206 (www.youtube.com)	Unclassified	YouTube	Accept (UTM Allowed)	1
2026/01/20 16:50:33	10.1.100.24		162.159.140.229 (www.twitter.com)	Unsanctioned	Twitter	Accept (UTM Allowed)	1
2026/01/20 16:43:32	10.1.100.24		142.251.33.78 (www.youtube.com)	Unclassified	YouTube	Accept (UTM Allowed)	1
2026/01/20 16:43:29	10.1.100.24		142.250.73.110 (www.youtube.com)	Unclassified	YouTube	Accept (UTM Allowed)	1
2026/01/20 16:39:28	10.1.100.24		57.144.216.1 (www.facebook.com)	Sanctioned	Facebook	Accept (UTM Allowed)	1
2026/01/20 16:34:58	10.1.100.24		162.159.140.229 (www.twitter.com)	Unsanctioned	Twitter	Accept (UTM Allowed)	1
2026/01/20 16:34:55	10.1.100.24		57.144.216.1 (www.facebook.com)	Unclassified	unknown-43448	Accept (UTM Allowed)	1

Sample traffic logs:

```
1: date=2026-01-20 time=16:55:09 eventtime=1768956908971028919 tz="-0800" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vd1" srcip=10.1.100.24 srcport=56176
srcintf="port18" srcintfrole="undefined" dstip=57.144.216.1 dstport=443 dstintf="port17"
dstintfrole="undefined" srccountry="Reserved" dstcountry="United States" sessionid=2012
proto=6 action="client-rst" policyid=1 policytype="policy" poluuid="e47d62aa-f62f-51f0-91d4-
78a59eb63616" service="HTTPS" trandisp="snat" transip=172.16.200.1 transport=56176 appid=15832
app="Facebook" appcat="Social.Media" apprisk="medium" appstatus="sanctioned" applist="default"
appact="detected" duration=5 sentbyte=296 rcvdbyte=112 sentpkt=6 rcvdpkt=2 utmaction="allow"
countapp=1 utmref=65534-444

1: date=2026-01-20 time=16:55:20 eventtime=1768956920271027543 tz="-0800" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vd1" srcip=10.1.100.24 srcport=48354
srcintf="port18" srcintfrole="undefined" dstip=172.66.0.227 dstport=443 dstintf="port17"
dstintfrole="undefined" srccountry="Reserved" dstcountry="United States" sessionid=2018
proto=6 action="close" policyid=1 policytype="policy" poluuid="e47d62aa-f62f-51f0-91d4-
78a59eb63616" service="HTTPS" trandisp="snat" transip=172.16.200.1 transport=48354 appid=16001
app="Twitter" appcat="Social.Media" apprisk="elevated" appstatus="unsanctioned"
applist="default" appact="detected" duration=1 sentbyte=216 rcvdbyte=164 sentpkt=4 rcvdpkt=3
utmaction="allow" countapp=2 utmref=65534-484

1: date=2026-01-20 time=16:55:35 eventtime=1768956936151029070 tz="-0800" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vd1" srcip=10.1.100.24 srcport=58128
srcintf="port18" srcintfrole="undefined" dstip=74.125.195.93 dstport=443 dstintf="port17"
dstintfrole="undefined" srccountry="Reserved" dstcountry="United States" sessionid=2025
proto=6 action="client-rst" policyid=1 policytype="policy" poluuid="e47d62aa-f62f-51f0-91d4-
78a59eb63616" service="HTTPS" trandisp="snat" transip=172.16.200.1 transport=58128 appid=31077
app="YouTube" appcat="Video/Audio" apprisk="elevated" applist="default" appact="detected" du
```

Example 2: creating override filters in application control profiles

Example 2 builds on example 1 by adding classification filters to an application control profile. The filters let administrators deny or allow specific sanctioned or unsanctioned applications.

In example 1, an administrator marked the Facebook application, Social Media category, and implicit rule default classification as follows:

- Facebook application was marked as sanctioned.
- Social Media category was marked as unsanctioned.
- In the implicit rule, the default classification was set to unclassified for all other applications.

In example 2, the administrator also creates an application control profile with the following classification filters:

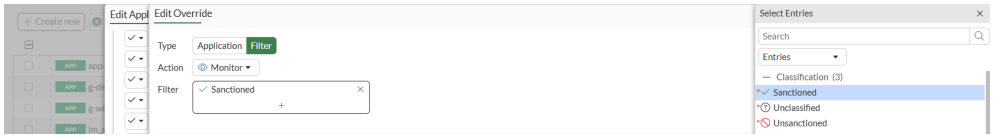
- One filter allows and monitors sanctioned applications.
- A second filter blocks unsanctioned applications.

After a user visits Facebook and Twitter, the signatures and override filters are triggered, and the results are displayed in the application control logs.

To add classification filters to application control profiles in the GUI:

1. Go to *Security Profiles* > *Application Control*, and click *Create New* to create a new profile.
2. Set *Name*.
3. Configure a filter override for sanctioned applications:

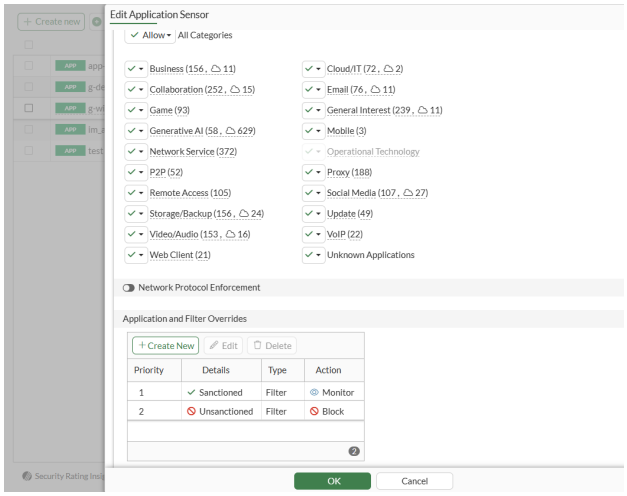
a. Under *Application and Filter Overrides*, click *Create New*.
b. Set *Type* to *Filter*.
c. Set *Action* to *Monitor*.
d. Click *Filter* and select *Sanctioned*.



e. Click *OK*.

4. Configure a filter override for unsanctioned applications:
   a. Under *Application and Filter Overrides*, click *Create New*.
   b. Set *Type* to *Filter*.
   c. Set *Action* to *Block*.
   d. Click *Filter* and select *Unsanctioned*.
   e. Click *OK*.

   In the *Application and Filter Overrides* table, the *Details* column displays *Sanctioned* and *Unsanctioned* for the two filters. The *Priority* column indicates the order in which overrides are evaluated, and administrators can drag and reorder the entries to set the desired precedence.



5. Click *OK* to save the profile.
6. Go to *Policy & Objects* > *Firewall Policy*, and apply the application control profile to a firewall policy with a deep inspection ssl-ssh profile.

To add classification filters to application control profiles in the CLI:

1. Create an application control profile with two override filters:

```
config application list
    edit "test"
        config entries
            edit 1
                set classification sanctioned
                set action pass
```

```
            next
            edit 2
                set classification unsanctioned
                set action block
            next
        end
    next
end
```

2. Apply the application profile to a firewall policy with a deep inspection ssl-ssh profile:

```
config firewall policy
    edit 1
        set uuid 9215ed56-fe23-51f0-5aff-cfa3dc62e797
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "new-deep-inspection"
        set application-list "test"
        set nat enable
    next
end
```

To verify:

1. After a user has visited Facebook and Twitter, view the application control logs.
   FortiGate passes the sanctioned Facebook traffic and blocks the unsanctioned Twitter traffic.

```
1: date=2026-01-30 time=17:38:52 eventtime=1769823532910175686 tz="-0800" logid="1059028705"
type="utm" subtype="app-ctrl" eventtype="signature" level="warning" vd="vd1" appid=16001
srcip=10.1.100.22 srccountry="Reserved" dstip=151.101.2.146 dstcountry="United States"
srcport=38002 dstport=443 srcintf="port2" srcintfrole="undefined" dstintf="port1"
dstintfrole="undefined" proto=6 service="SSL" direction="outgoing" policyid=1
poluuid="9215ed56-fe23-51f0-5aff-cfa3dc62e797" policytype="policy" sessionid=1695
applist="test" action="block" appcat="Social.Media" app="Twitter" appstatus="unsanctioned"
hostname="www.twitter.com" incidentserialno=46 url="/" msg="Social.Media: Twitter"
apprisk="elevated"

2: date=2026-01-30 time=17:38:47 eventtime=1769823527228616258 tz="-0800" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="vd1" appid=15832
srcip=10.1.100.22 srccountry="Reserved" dstip=57.144.204.1 dstcountry="United States"
srcport=39594 dstport=443 srcintf="port2" srcintfrole="undefined" dstintf="port1"
dstintfrole="undefined" proto=6 service="SSL" direction="outgoing" policyid=1
poluuid="9215ed56-fe23-51f0-5aff-cfa3dc62e797" policytype="policy" sessionid=1694
applist="test" action="pass" appcat="Social.Media" app="Facebook" appstatus="sanctioned"
hostname="www.facebook.com" incidentserialno=44 url="/" msg="Social.Media: Facebook"
```

```
apprisk="medium" scertcname="*.facebook.com" scertissuer="DigiCert Global G2 TLS RSA SHA256
2020 CA1"
```

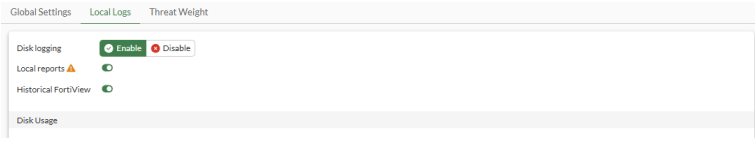
Example 3: using FortiView

This example demonstrates how to configure logging of classified applications, and then use FortiView to:

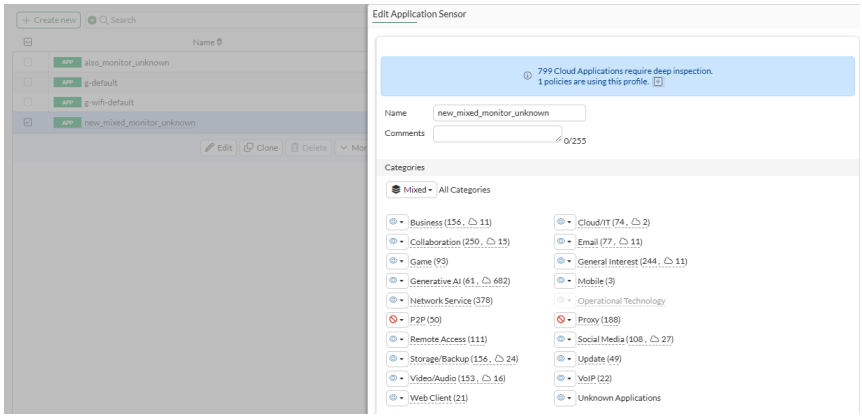
- Classify an application as unsanctioned
- Block an application

To configure logging of sanctioned and unsanctioned applications in the GUI:

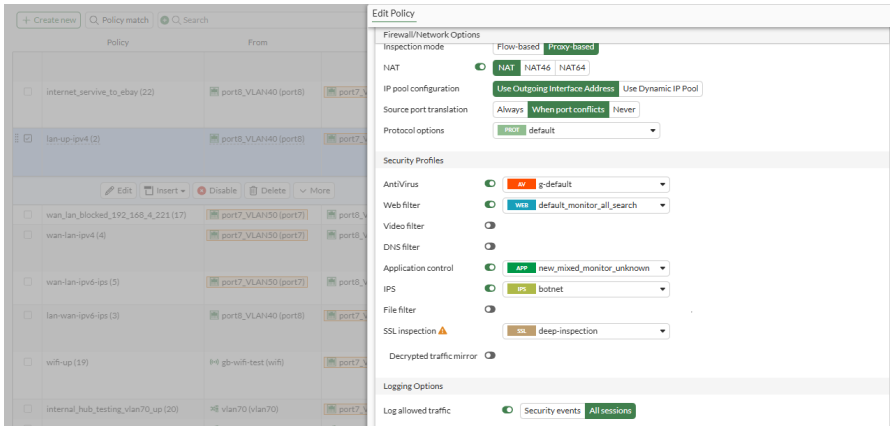
1. Go to *Log & Report* > *Log Settings* > *Local Logs*, and enable *Disk logging* and *Historical FortiView*.



2. Create an application control sensor.



3. Create a firewall policy with the application control sensor selected and logging of allowed traffic enabled.



To configure logging of sanctioned and unsanctioned applications in the CLI:

1. Create an application control profile:

```
config application list
    edit "new_mixed_monitor_unknown"
        set uuid cb295a4a-18d3-51f1-afdb-982be464621b
        set other-application-log enable
        set unknown-application-log enable
        config entries
            edit 1
                set category 2 6
            next
        end
    next
end
```

2. Create a firewall policy with the application control sensor selected and logging of allowed traffic enabled:

```
config firewall policy
    edit 2
        set name "lan-up-ipv4"
        set uuid cf8a295c-18d3-51f1-3278-1d5441886be9
        set srcintf "port8"
        set dstintf "port7"
        set action accept
        set srcaddr "lan_ipv4"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set av-profile "g-default"
        set webfilter-profile "default_monitor_all_search"
        set ips-sensor "botnet"
        set application-list "new_mixed_monitor_unknown"
        set logtraffic all
        set nat enable
    next
end
```

To classify an application as unsanctioned in FortiView:

1. Go to *Dashboard* > *FortiView* > *Applications*, and select a disk data source and a historical time period:
   a. Click *Settings*, and set *Data source* to disk logging.
   b. Set *Time period* to *1 hour*, *24 hours*, or *7 days*.
   c. Click *OK*.
2. Add the *Application Classification* column to the table and view the classification status.

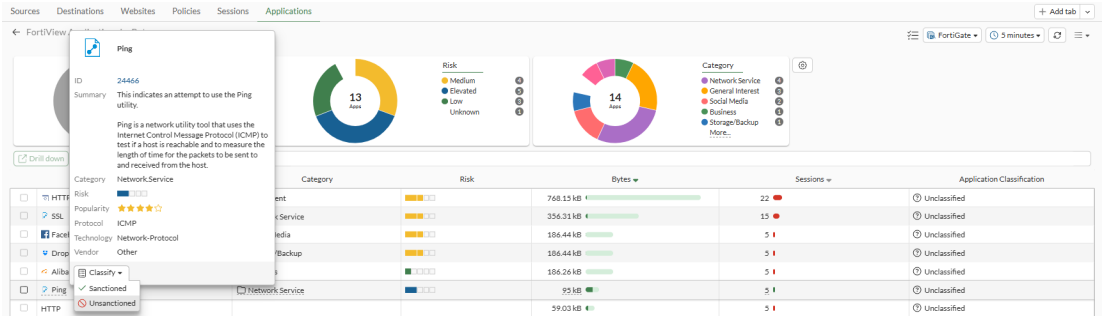
a. Hover the mouse over the first column heading, and click the *Configure Table (gear)* icon.
b. Move the *Application Classification* column to the *Active Columns* list, and click *Apply*. The *Application Classification* column is displayed.

   By default all applications display *Unclassified* status.

3. Classify an application as unsanctioned:

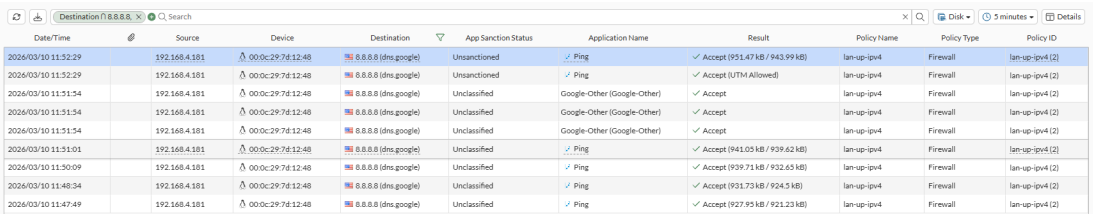
   In this example, Ping is classified as unsanctioned.

   a. Hover over the Ping application. The *Classify* button is displayed.
   b. Click *Classify* and choose *Unsanctioned*.



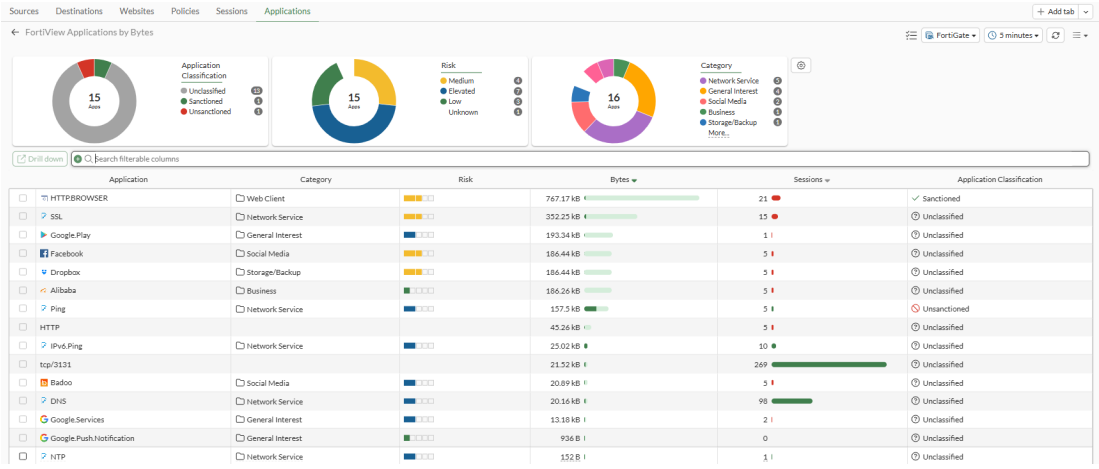
4. Start a new session, using the unsanctioned Ping application.
5. Go to *Log & Report* > *Forward Traffic*, add the *App Sanction Status* column, and view the status for the Ping application sessions before an after it was labeled an unsanctioned application.

   a. Hover the mouse over the first column heading, and click the *Configure Table (gear)* icon.
   b. Move the *App Sanction Status* column to the *Active Columns* list, and click *Apply*. The *App Sanction Status* column is displayed.
   c. View the *App Sanction Status* column for the Ping application. The new session displays *Unsanctioned* status, and the old sessions display *Unclassified* status.

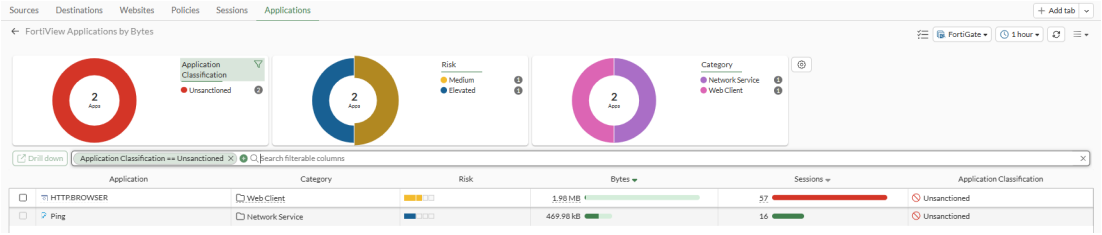


6. Go to *Dashboard* > *FortiView* > *Applications*.

   a. Choose *5 min* time frame.
   b. In the *Application Classification* column, view the Ping application status is *Unsanctioned*.

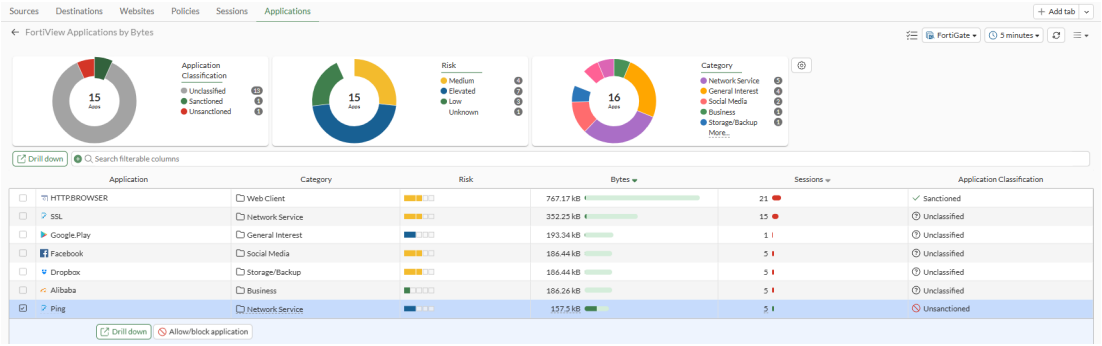


7. In the pie chart, click the *Unsanctioned* filter to display only unsanctioned applications in the table.



To block an application in FortiView:

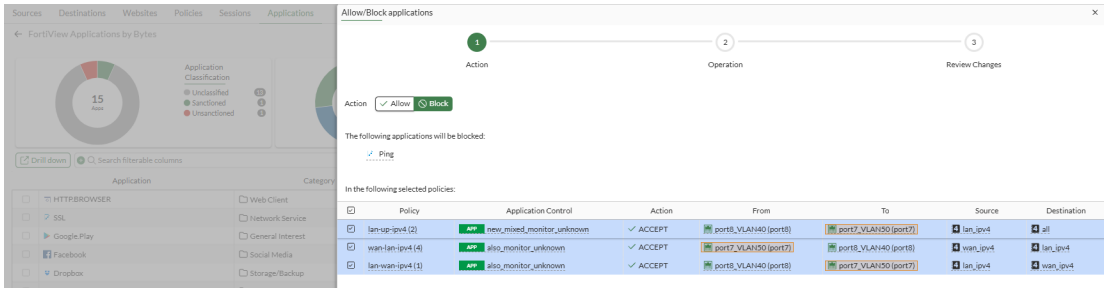
1. Go to *Dashboard* > *FortiView* > *Applications*.
2. Select the Ping application and click the *Allow/block application* button.



The *Allow/Block applications* wizard appears, starting on the *Action* step.

3. Complete the *Action* step:

a. Set *Action* to *Block*.



b. View the list of affected policies.

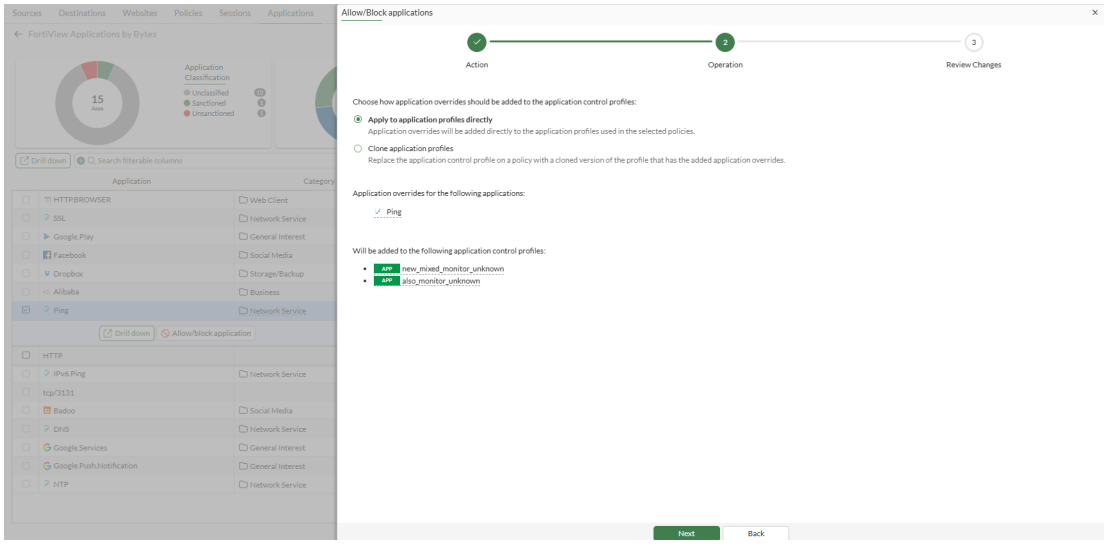
c. Click *Next*. The wizard moves to the *Operation* step.

4. Complete the *Operation* step:

The following options are available:

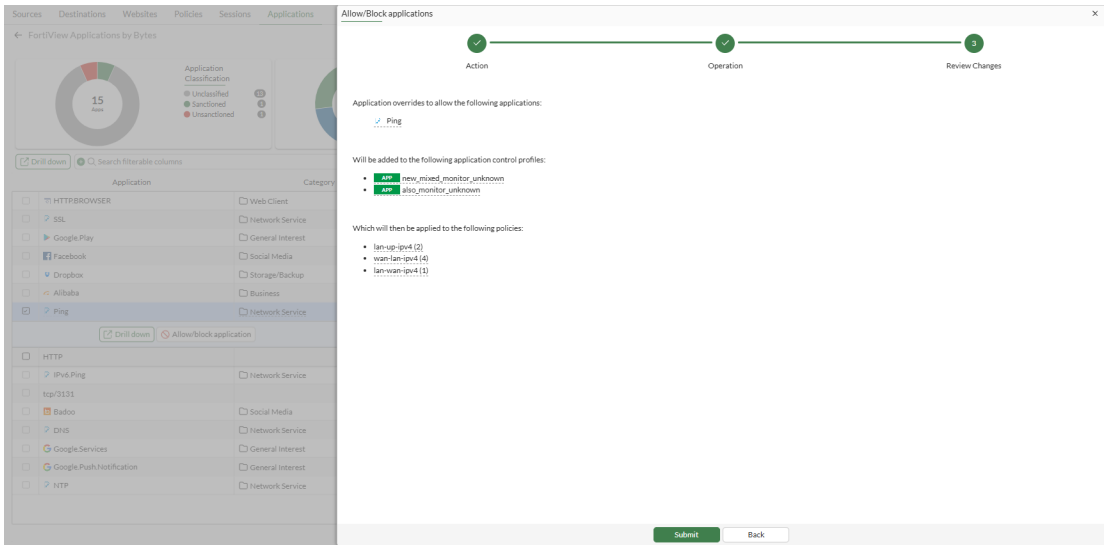
- *Apply to application profile directly*: Application overrides will be added directly to the application profiles used in the selected policies.
- *Clone application profiles*: Replace the application control profile on a policy with a cloned version of the profile that has the added application overrides.

a. Select *Apply to application profile directly*.

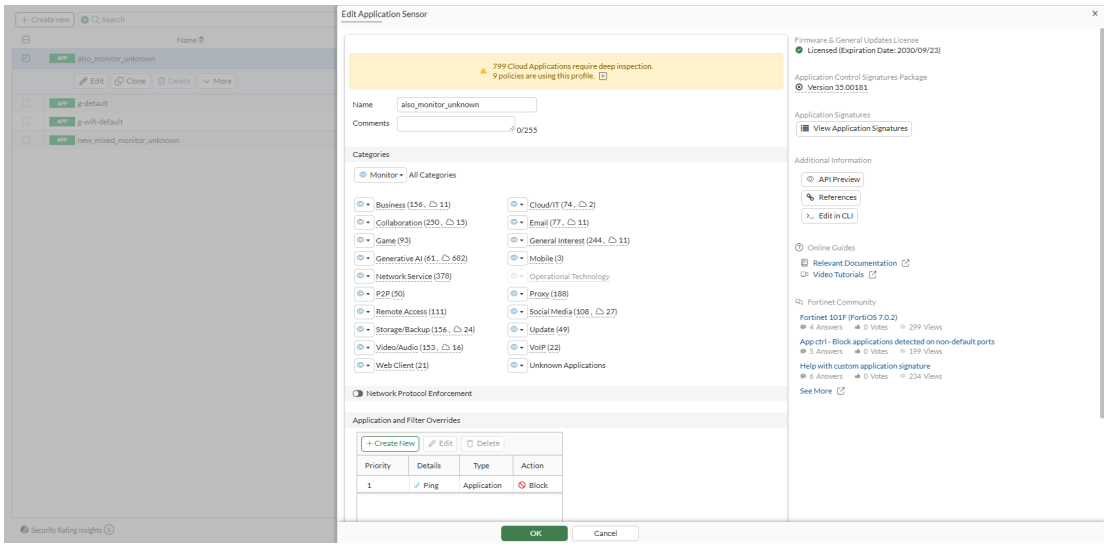


b. Click *Next*. The wizard moves to the *Review Changes* step.

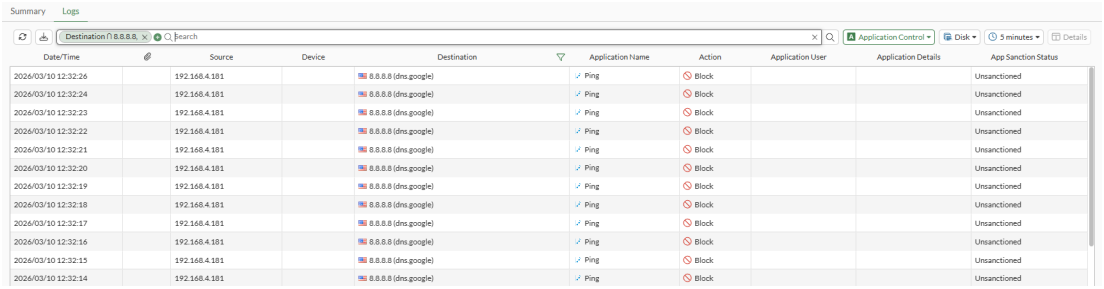
5. On the *Review Changes* step, review the changes and click *Submit* to save the changes.



6. Go to *Security Profiles* > *Application Control*, and view the blocked application in the *Application and Filter Overrides* list.
   a. On the *Application Control* page, select the sensor, and click *Edit*.
   b. Check that the blocked application is in the *Application and Filter Overrides* list with correct action.



   c. Click *Cancel*.
7. Try to use the application and check that it is blocked.
   a. Go to *Log & Report* > *Security Events* > *Application Control*.
   b. Check that the application was blocked by the application control override rule.



Support for hybrid PQC SSL deep inspection in proxy mode

This information is also available in the FortiOS 8.0 Administration Guide:
- Hybrid PQC support in proxy mode SSL deep inspection

FortiGate now supports TLS 1.3 hybrid Post-Quantum Cryptography (PQC) key exchanges in SSL deep inspection (proxy mode), enabling secure traffic inspection. This ensures compatibility with modern browsers and PQC-enabled servers that use hybrid PQC algorithms, such as X25519MLKEM768.

Now, both pure PQC and hybrid PQC are supported in SSL deep inspection:

- Hybrid PQC algorithms:
  - X25519MLKEM768 (0x11EC)
  - SecP256r1MLKEM768 (0x11EB)
  - SecP384r1MLKEM1024 (0x11ED)
- Pure PQC algorithms:
  - MLKEM512 (0x1302)
  - MLKEM768 (0x1302)
  - MLKEM1024 (0x1302)

FortiGate can now perform SSL deep inspection and the client can access PQC algorithm web servers in proxy mode.

These changes also add support for DNS over TLS, HTTPS (1.0, 2.0, 3.0), POP3, IMAPS, and SMTPS with deep inspection to negotiate hybrid+ML-KEM. Hybrid+ML-KEM and ML-KEM are also supported when these protocols run in a virtual-server or explicit-proxy modes.

Although SIP over TLS and the urlfilter TLS connection to FortiGuard use the same TLS implementation as HTTPS (1.0, 2.0, 3.0), POP3, IMAPS, and SMTPS, they do not yet support hybrid+ML-KEM.

Example

This example demonstrates the following scenarios:

- How clients can connect to a web server that uses PQC encryption
- How clients can access through a DNS connection to Cloudflare a web server using hybrid PQC key exchange

First configure the FortiGate for testing, and use the same configuration to test both scenarios.

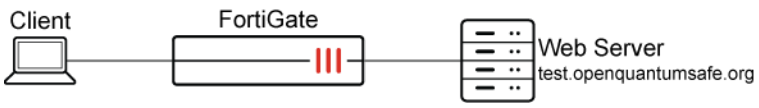
To configure FortiGate for testing:

1. On FortiGate, clone a deep-inspection profile and add port 6000-6005 to HTTPS:
   a. Go to *Security Profiles* > *SSL/SSH Inspection*.
   b. Select a deep-inspection profile, and click *Clone*.
   c. Leave *Name* set to *Clone of deep-inspection*.
   d. Under *Protocol Port Mapping*, enable *HTTPS*, and enter *443,6000,6001,6002,6003,6004,6005*.
   e. Click *OK* to save the changes to *Clone of deep-inspection*.
2. Add Port Range 6000–6005 to the HTTPS Service:
   a. Go to *Policy & Objects* > *Service*.
   b. Locate the service named HTTPS and click *Edit*.
   c. In the *Edit Service* window, scroll to the *Protocol Options* section.
   d. Under *Destination Port*, click + to add a new port entry.
   e. Set the second entry to:
      - *Protocol*: TCP
      - *Low*: 6000
      - *High*: 6005
   f. Confirm the configuration and click *OK* to save the service.
3. On FortiGate, configure a policy and select the *Clone of deep-inspection* profile.

```
config firewall policy
   edit 1
       set srcintf "port2"
       set dstintf "port1"
       set action accept
       set srcaddr "all"
       set dstaddr "all"
       set schedule "always"
       set service "ALL"
       set utm-status enable
       set inspection-mode proxy
       set ssl-ssh-profile "Clone of deep-inspection"
       set av-profile "av"
       set logtraffic all
       set nat enable
  next
end
```

4. Test the following scenarios:
   - Connect to a web server that uses PQC encryption on page 273
   - Send DNS to a Cloudflare DNS server that uses PQC encryption on page 274

Connect to a web server that uses PQC encryption



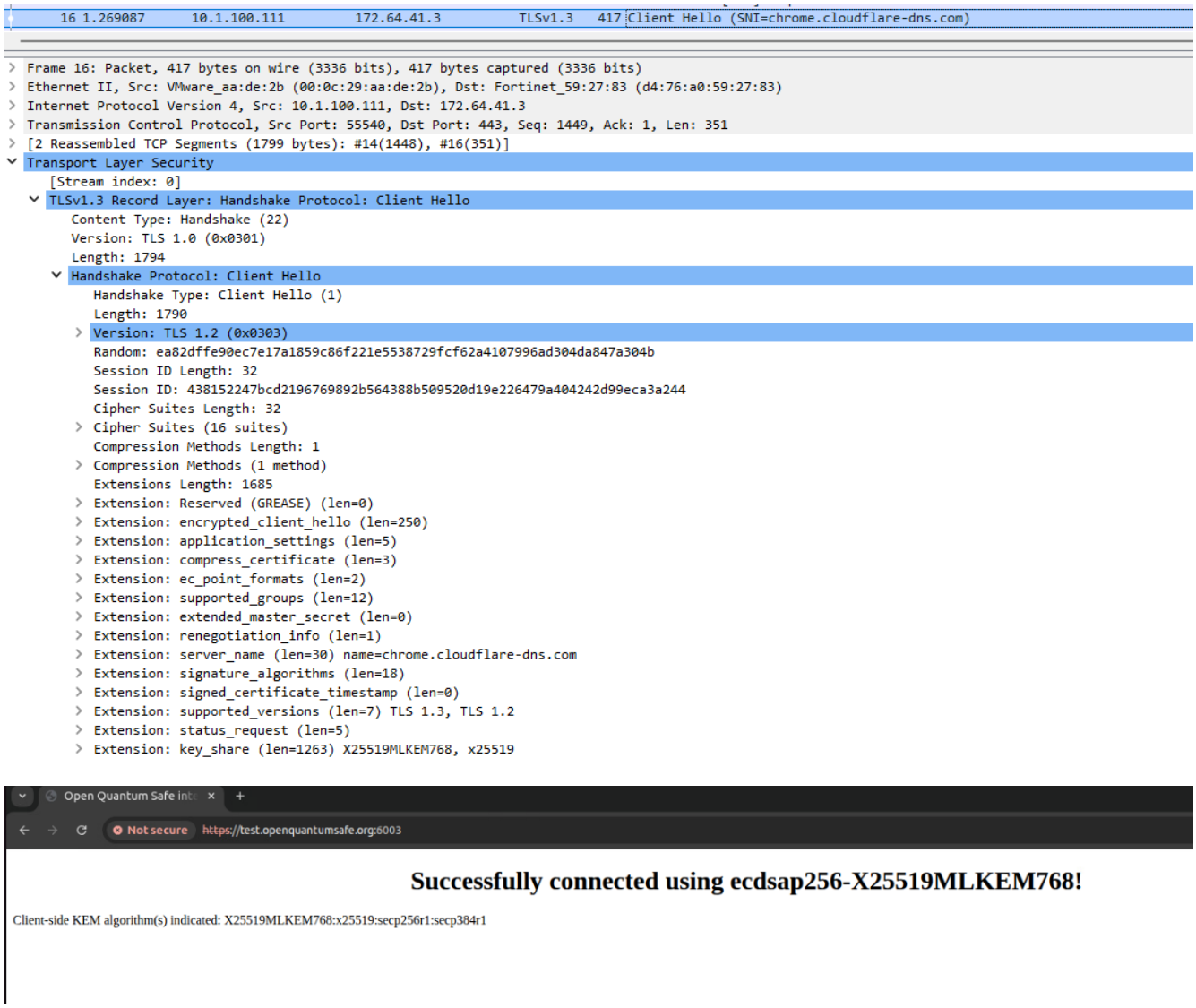
`test.openquantumsafe.org:6000` supports the X25519MLKEM768 key exchange algorithm. Previously, hybrid PQC algorithm was downgraded to non-ML-KEM encryption, and the site would be unreachable. Now, the client can access the web server using hybrid PQC key exchange.

Send DNS to a Cloudflare DNS server that uses PQC encryption

To test the example:

1. In a browser, enable secure DNS connection with Cloudflare (1.1.1.1) to force DNS over TLS. The client can access the web server using hybrid PQC key exchange.

2. Packet capture shows that the Client Hello sent to the Cloudflare DNS server uses X25519MLKEM768.



Enhanced UTM protection with WebSocket traffic inspection

Support for WebSocket traffic inspection is added , allowing UTM modules, including DLP, antivirus, IPS, and File filter, to detect and block sensitive data, malware, and restricted files sent over WebSocket. With the growing adoption of WebSocket-based applications, this provides essential security coverage that was previously unavailable.

This allows UTM to scan message transmissions inside chatbots that use WebSocket, such as Microsoft Copilot. Files that are transferred over WebSocket in other usage scenarios can also now be scanned.

WebSocket protocol options are added in `profile-protocol-options`, allowing users to specify WebSocket protocol settings, such as oversize control, uncompressed-nest-limit, bzip2 scan, and others.

This feature is only supported in proxy mode.

Example configuration

In this example, WebSocket profile protocol is enabled, and DLP, antivirus, and file filter profiles are configured and used in a proxy mode firewall policy.

For comparisons of the results with and without WebSocket traffic inspection, see:

- DLP scan over Microsoft Copilot chatbot on page 278
- Antivirus blocks viruses sent over WebSocket on page 280
- File filter blocks matching file type sent over WebSocket on page 281

To enable WebSocket profile protocol and configure profiles:

1. Enable the WebSocket profile protocol option:

```
config firewall profile-protocol-options
    edit "protocol"
        config websocket
            set status enable
        end
    next
end
```

2. Configure the DLP dictionary and sensors:

```
config dlp dictionary
    edit "ssn"
        config entries
            edit 1
                set type "g-ssn-us"
                set repeat enable
            next
        end
    next
end
```

```
config dlp sensor
    edit "genai"
        config entries
            edit 1
                set dictionary "fortinet"
            next
            edit 2
                set dictionary "ssn"
            next
            edit 3
                set dictionary "credit"
            next
```

```
        end
    next
end
```

3. Configure a DLP profile:

```
config dlp profile
    edit "genai"
        set feature-set proxy
        config rule
            edit 1
                set proto smtp pop3 imap http-get http-post ftp nntp mapi ssh cifs websocket
                set filter-by sensor
                set sensor "genai"
                set action block
            next
        end
    next
end
```

4. Configure an antivirus profile:

```
config antivirus profile
    edit "av"
        set feature-set proxy
        config websocket
            set av-scan block
        end
    next
end
```

5. Configure a file filter profile:

```
config file-filter profile
    edit "filefilter"
        set feature-set proxy
        config rules
            edit "1"
                set protocol http ftp smtp imap pop3 mapi cifs ssh websocket
                set action block
                set file-type "msoffice" "pdf"
            next
        end
    next
end
```

6. Configure a policy that uses the profiles:

```
config firewall policy
    edit 1
        set name "test"
        set srcintf "port1"
```

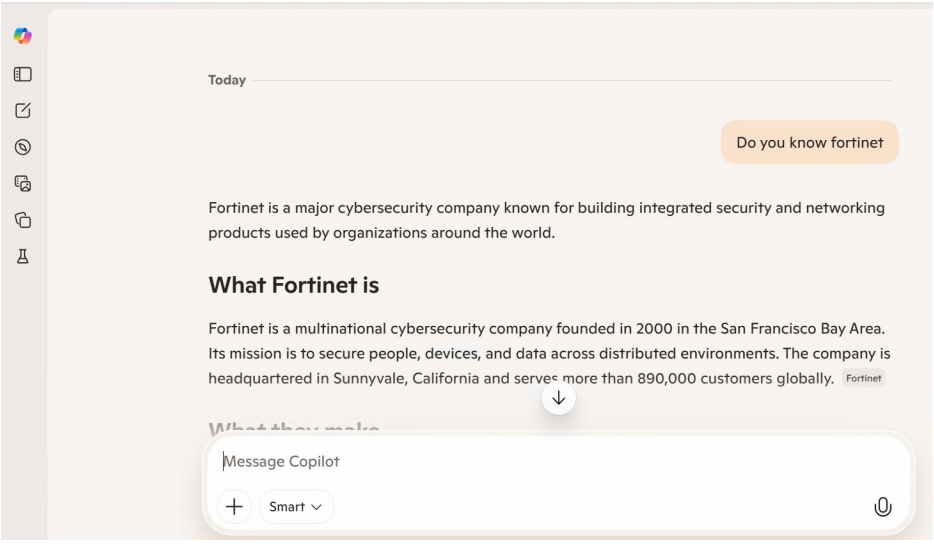
```
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "ssl"
        set av-profile "av"
        set dlp-profile "genai"
        set file-filter-profile "filefilter"
        set nat enable
    next
end
```

DLP scan over Microsoft Copilot chatbot

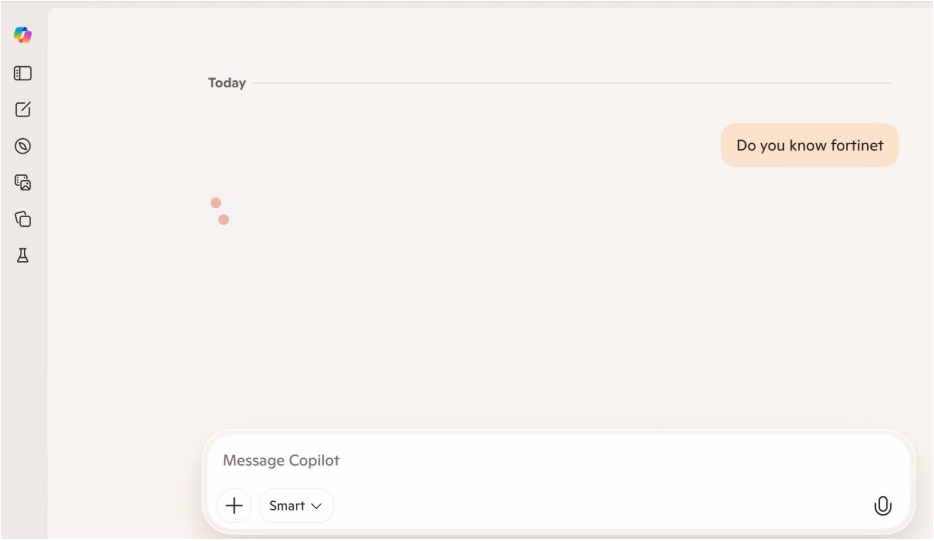
In previous versions, user configured keywords and sensitive data entered in the Microsoft Copilot chatbot were not blocked or logged by DLP.

In this version, DLP scan works on Copilot and blocks and logs traffic based on the user configuration. When keywords or sensitive data are matched, the Copilot user does not receive a response because the message is not sent successfully.

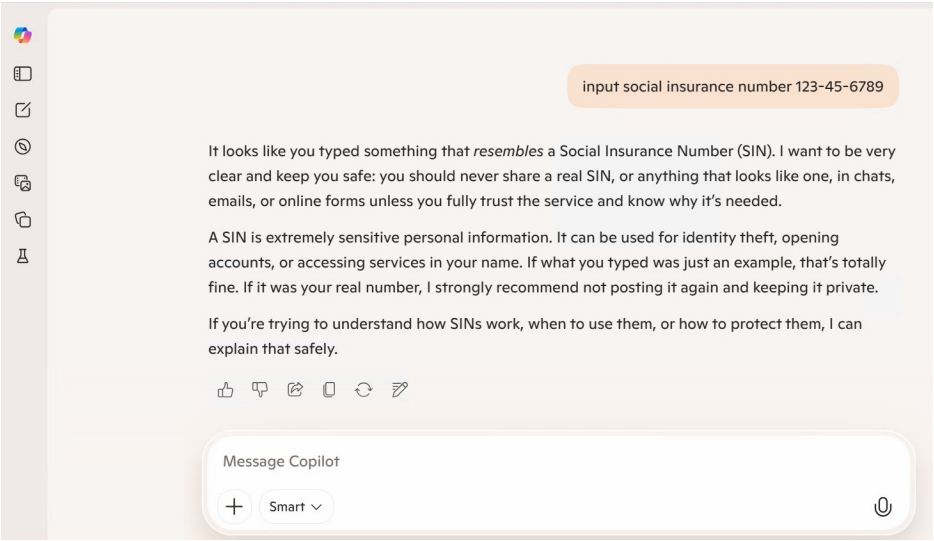
Keyword not blocked



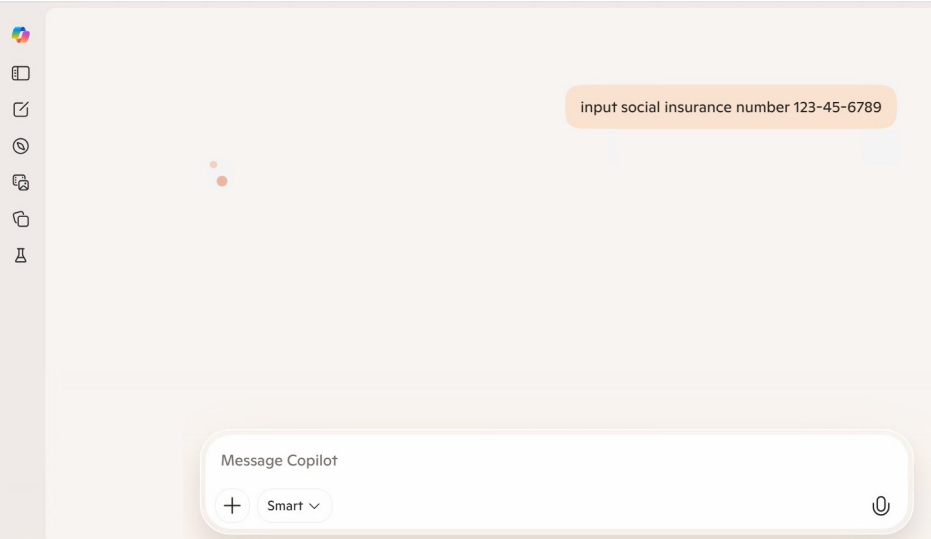
Keyword blocked



Sensitive data not blocked



Sensitive data blocked



DLP UTM log:

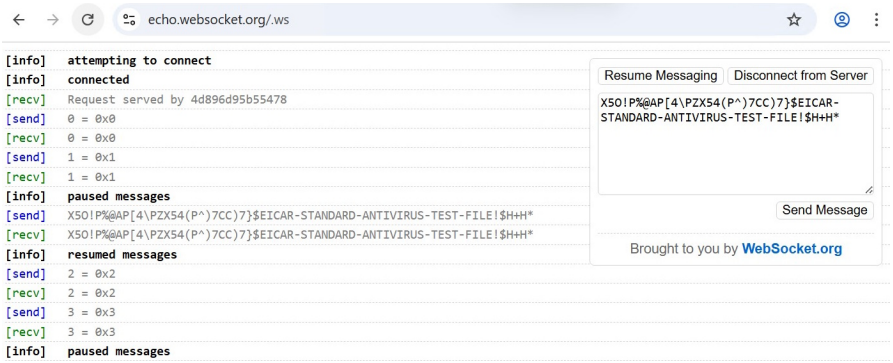
```
2: date=2026-02-11 time=15:18:31 eventtime=1770851910062236983 tz="-0800" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="vdom1" ruleid=1 dlpextra="Sensor
'genai' matching any: ('fortinet'=1) >= 1; match." filtertype="sensor" filtercat="file"
severity="medium" policyid=1 poluuid="53f43e62-fae3-51f0-a25e-3d6e8df10c42" policytype="policy"
sessionid=4776 epoch=730271499 eventid=0 srcip=10.1.100.195 srcport=51283 srccountry="Reserved"
srcintf="port1" srcintfrole="undefined" srcuuid="34bf29b2-fae3-51f0-b9a0-dc1e6936a985"
dstip=104.18.22.222 dstport=443 dstcountry="United States" dstintf="port3" dstintfrole="undefined"
dstuuid="34bf29b2-fae3-51f0-b9a0-dc1e6936a985" proto=6 service="WEBSOCKET" subservice="WEBSOCKET"
filetype="genscript" direction="outgoing" action="block" profile="genai"

1: date=2026-02-11 time=15:19:26 eventtime=1770851966412491870 tz="-0800" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="vdom1" ruleid=1 dlpextra="Sensor
'genai' matching any: ('fortinet'=0) >= 1; ('ssn'=1) >= 1; match." filtertype="sensor"
filtercat="file" severity="medium" policyid=1 poluuid="53f43e62-fae3-51f0-a25e-3d6e8df10c42"
policytype="policy" sessionid=4787 epoch=730271642 eventid=0 srcip=10.1.100.195 srcport=51298
srccountry="Reserved" srcintf="port1" srcintfrole="undefined" srcuuid="34bf29b2-fae3-51f0-b9a0-
dc1e6936a985" dstip=104.18.22.222 dstport=443 dstcountry="United States" dstintf="port3"
dstintfrole="undefined" dstuuid="34bf29b2-fae3-51f0-b9a0-dc1e6936a985" proto=6 service="WEBSOCKET"
subservice="WEBSOCKET" filetype="unknown" direction="outgoing" action="block" profile="genai"
```

Antivirus blocks viruses sent over WebSocket

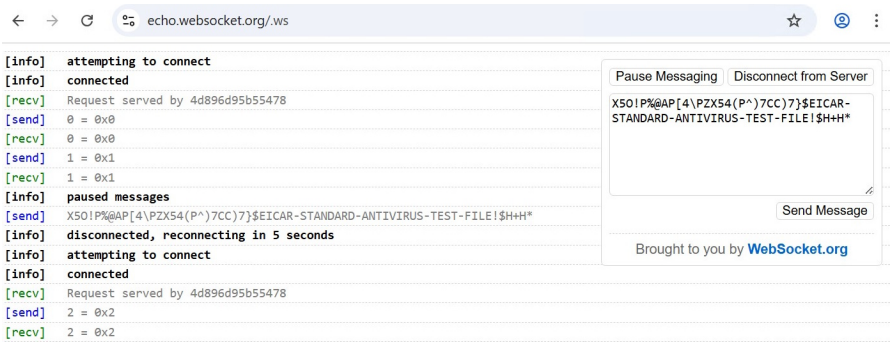
In previous versions, virus content cannot be blocked when sent over WebSocket.

In this version, virus content can be blocked when sent over WebSocket.

Virus content not blocked



Virus content blocked



AV UTM log:

```
1: date=2026-02-11 time=15:32:10 eventtime=1770852730506964187 tz="-0800" logid="0211008192"
type="utm" subtype="virus" eventtype="infected" level="warning" vd="vdom1" policyid=1
poluuid="53f43e62-fae3-51f0-a25e-3d6e8df10c42" policytype="policy" msg="File is infected."
action="blocked" service="WEBSOCKET" subservice="WEBSOCKET" sessionid=5208 srcip=10.1.100.195
dstip=66.241.124.119 srcport=51497 dstport=443 srccountry="Reserved" dstcountry="United States"
srcintf="port1" srcintfrole="undefined" dstintf="port3" dstintfrole="undefined" srcuuid="34bf29b2-
fae3-51f0-b9a0-dc1e6936a985" dstuuid="34bf29b2-fae3-51f0-b9a0-dc1e6936a985" proto=6
direction="outgoing" quarskip="Quarantine-disabled" virus="EICAR_TEST_FILE" viruscat="Virus"
dtype="av-engine" itype="infected" ref="https://fortiguard.com/encyclopedia/virus/2172"
virusid=2172 profile="av"
analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

File filter blocks matching file type sent over WebSocket

In previous versions, matching files cannot be blocked when sent over WebSocket.

In this version, matching files can be blocked when sent over WebSocket.

Matching files not blocked

```
qa@pc06:~$ python3 ws_client_final_no_chunk.py -u wss://172.16.200.180/ws -f sample/test.doc
connect to wss://172.16.200.180/ws
sending file as ONE binary message
file: test.doc
size: 19456 bytes
Received: {"ok": true, "mode": "raw_single_message", "received": 19456, "sha256": "793920b2ba0ff8efb862e51ba0b50da25adcad4091e223932aec7700a46d3238", "saved_as": "uploads/raw_upload-20260212-18
246.bin"}
qa@pc06:~$
```

Matching files blocked

```
qa@pc06:~$ python3 ws_client_final_no_chunk.py -u wss://172.16.200.180/ws -f sample/test.doc
connect to wss://172.16.200.180/ws
sending file as ONE binary message
file: test.doc
size: 19456 bytes
Connection closed: received 1008 (policy violation) traffic blocked by websocket scan; then sent 1008 (policy violation) traffic blocked by websocket scan
qa@pc06:~$
```

File filter UTM log:

```
1: date=2026-02-11 time=15:48:38 eventtime=1770853718415697143 tz="-0800" logid="1900064000"
type="utm" subtype="file-filter" eventtype="file-filter" level="warning" vd="vdom1" policyid=1
poluuid="53f43e62-fae3-51f0-a25e-3d6e8df10c42" policytype="policy" sessionid=5548
srcip=10.1.100.194 srcport=53666 srccountry="Reserved" srcintf="port1" srcintfrole="undefined"
srcuuid="34bf29b2-fae3-51f0-b9a0-dc1e6936a985" dstip=172.16.200.180 dstport=443
dstcountry="Reserved" dstintf="port3" dstintfrole="undefined" dstuuid="34bf29b2-fae3-51f0-b9a0-
dc1e6936a985" proto=6 service="WEBSOCKET" subservice="WEBSOCKET" profile="filefilter"
direction="outgoing" action="blocked" rulename="1" filesize=19456 filetype="msoffice" msg="File
was blocked by file filter."
```

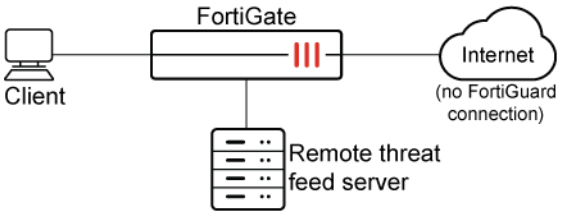
FortiGuard-free rating for local and external categories

Support is added for handling category detection, in flow inspection mode, without contacting FortiGuard servers when only local or external threat-feed categories are used in web filter profiles (NGFW profile mode), URL-category policies (NGFW policy mode), or SSL exemptions. This eliminates unnecessary rating traffic and improves performance.

Example

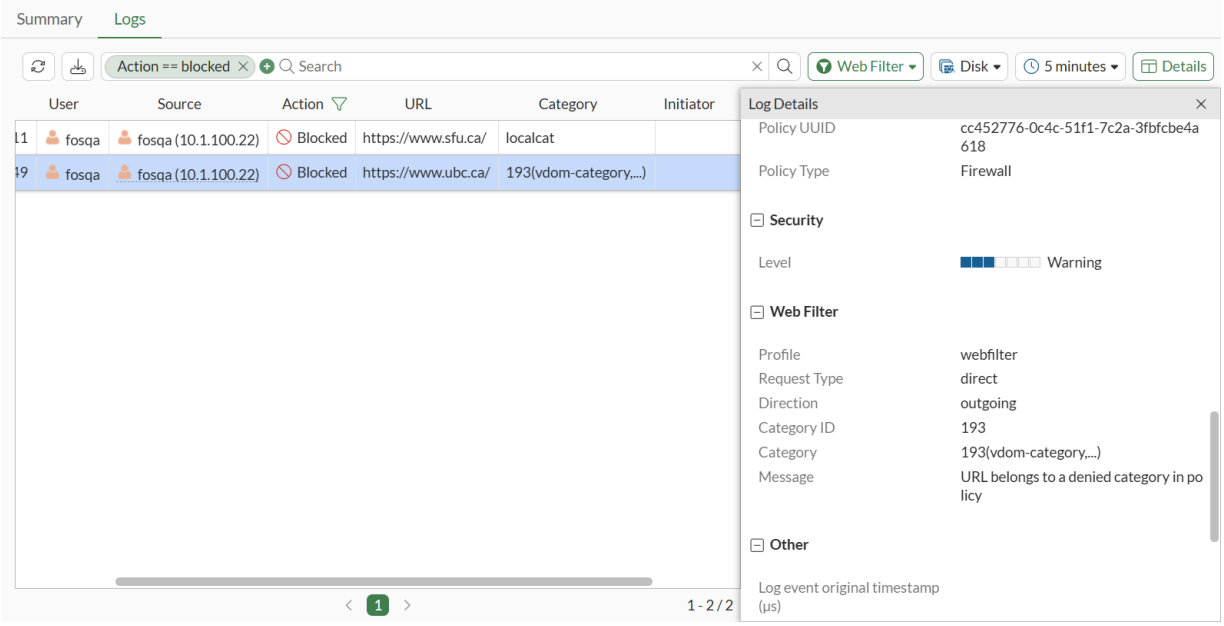
In this example, the FortiGate does not have a valid web filter license and no web filter rating server is usable. *www.sfu.ca* is the local category example defined locally and *www.ubc.ca* is the remote category example synchronized from the remote threat feed server.

For more detailed information about configuring a FortiGuard external feed, see FortiGuard category external feed.

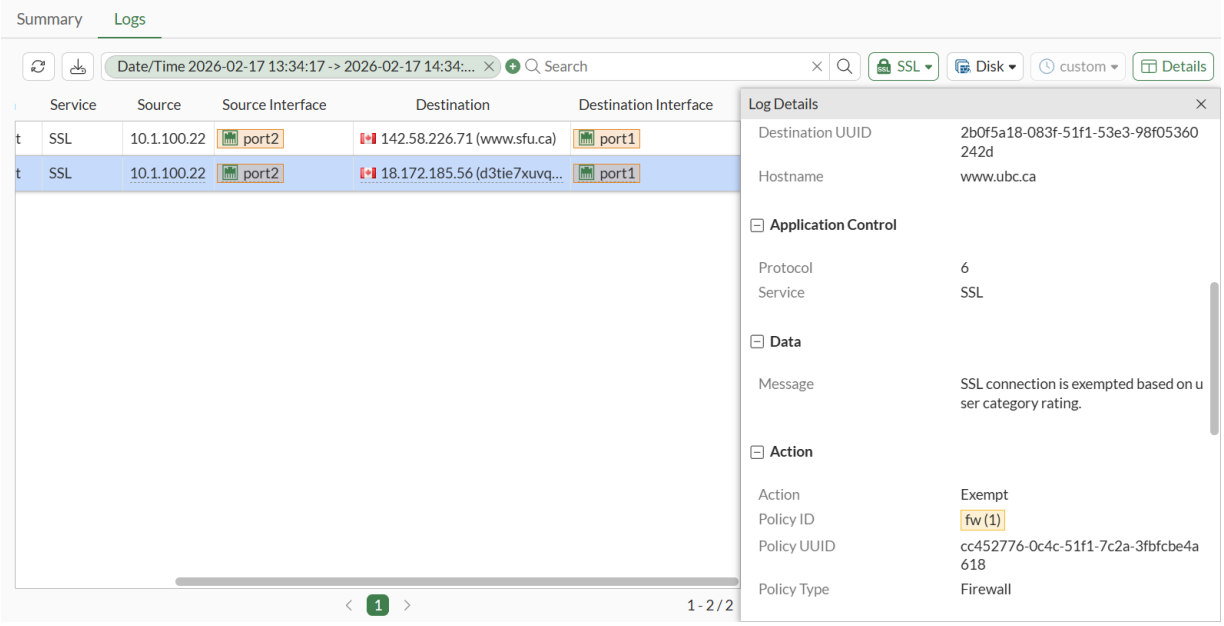


When there is no FortiGuard server connection, the IPS engine will still work in the following cases. The defined action is taken, and UTM logs are generated.

1. Firewall policy flow-mode:
   - Web filter from remote/local category:

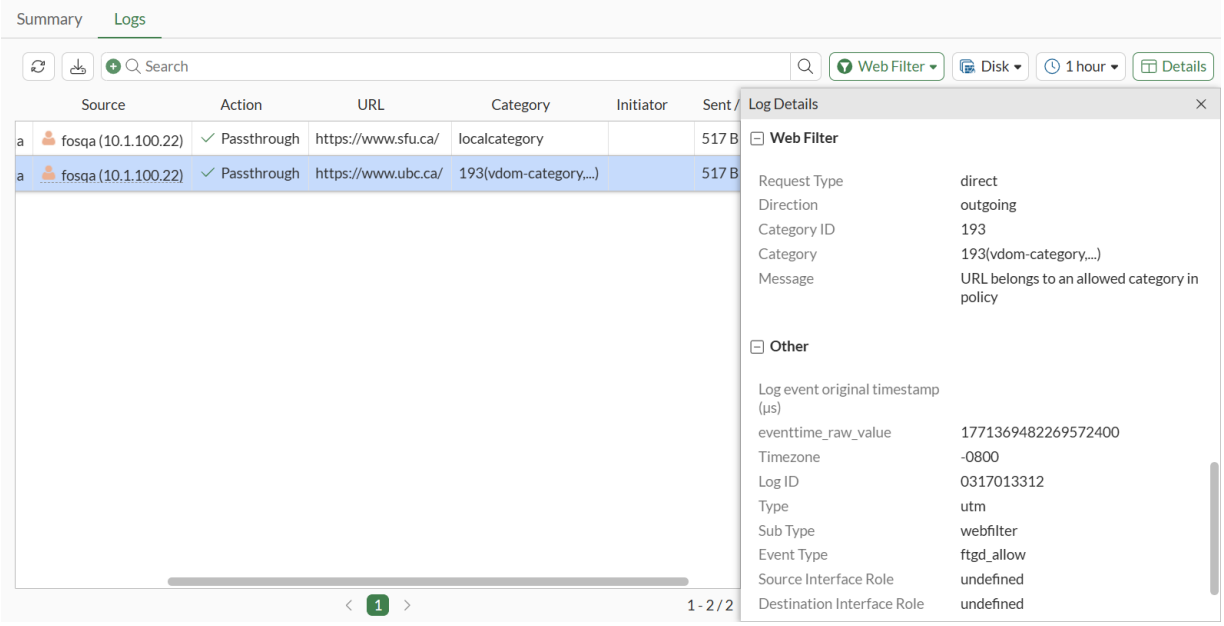


   - SSL exemption from remote/local category:

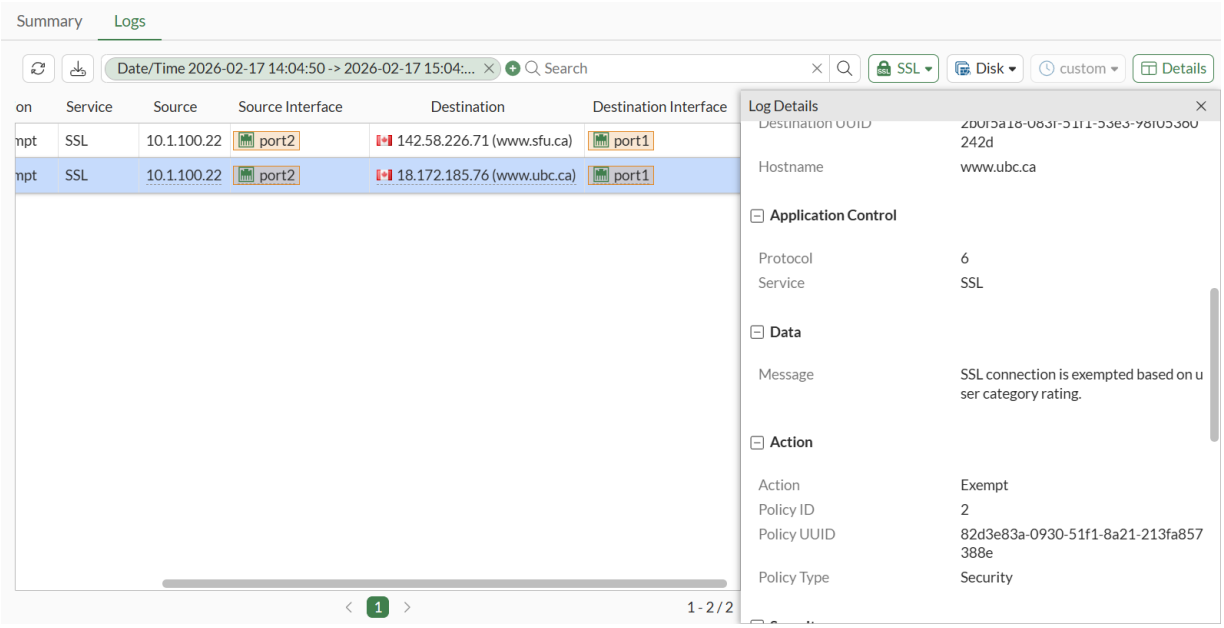


2. NGFW mode security policy:
   See NGFW policy for more information about NGFW policies.

- URL category from remote/local category:



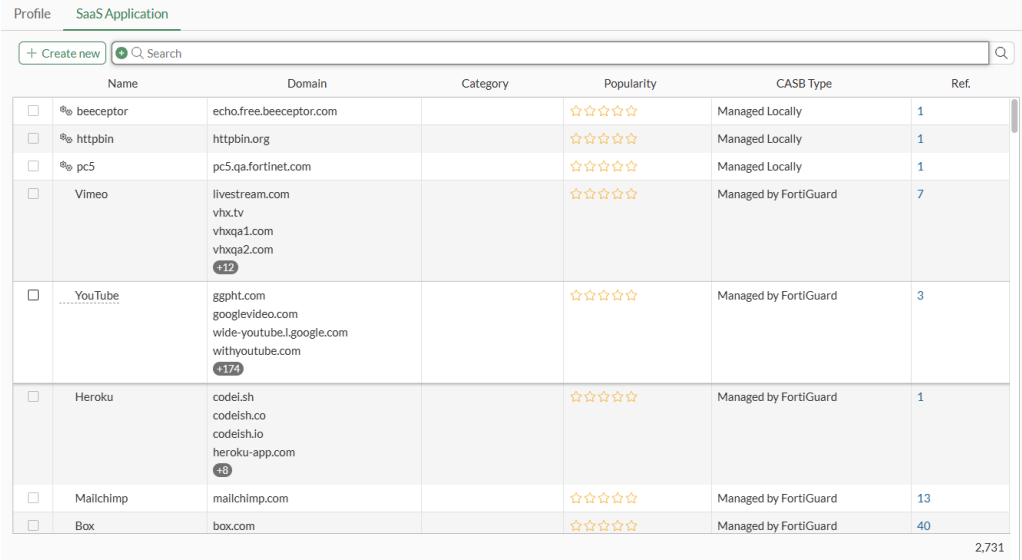
- SSL exemption from remote/local category:



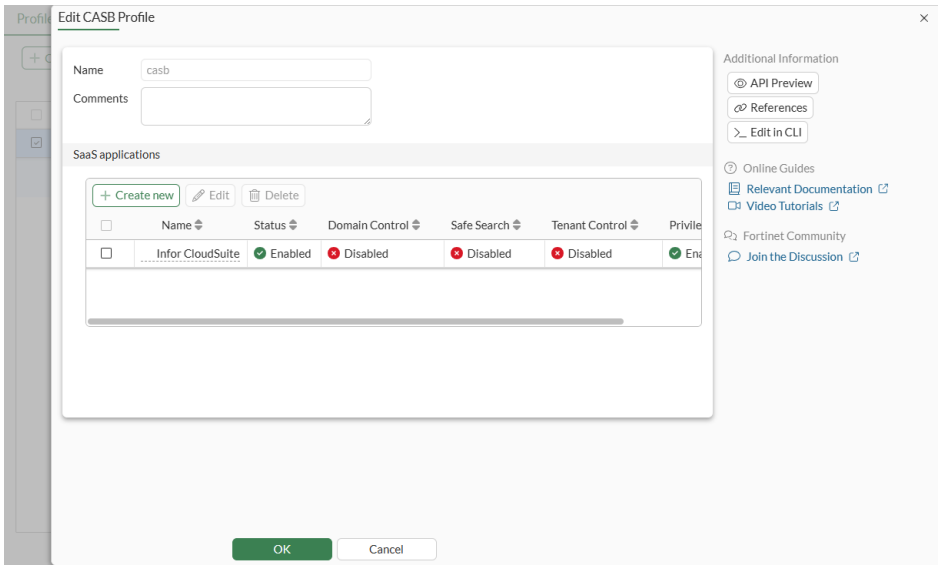
Expanded CASB SaaS application database

The number of recognized SaaS applications in the CASB database is expanded to more than 2700 entries. This strengthens inline CASB filtering by providing broader and more accurate application coverage and is available to customers with the Firmware & General Updates license

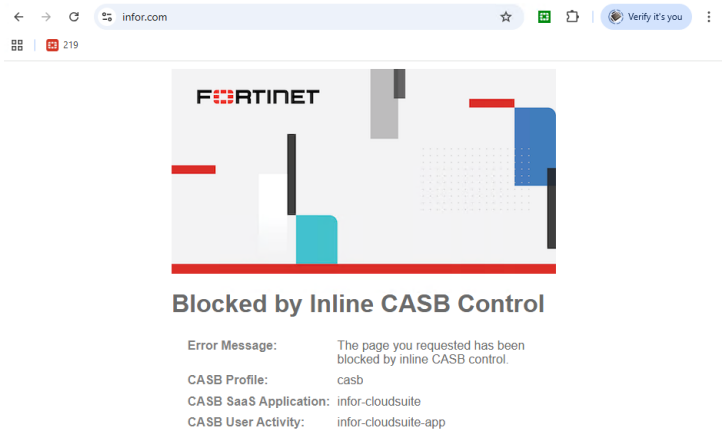
Go to *Security Profile* > *Inline-CASB* and select the *SaaS Application* tab to see the list of SaaS applications.



Go to *Security Profile* > *Inline-CASB* and create a new profile that blocks the *Infor ClouSuite* application.



Try to access the domain from a client behind the FortiGate, and a CASB blocking page is returned.



The CASB-DB is now delivered exclusively from the FortiGuard Distribution Server (FDS) for customers that have the license .

Protecting LLM and GenAI

This section includes information about new features related to protecting LLM and generative AI:

- Application control support for generative AI on page 286
- Agentic AI protocol support in FortiOS on page 286

Application control support for generative AI

Application control support has been added for generative AI to enhance the management and categorization of generative AI signatures. This offers improved visibility and insights into AI-related activities. This feature includes:

- Support for the new AIAP database type for generative AI rules.
- UTM application control generative AI log fields available in the CLI and GUI:

CLI log field	GUI log field
`aiuser`	AI User
`model`	Model
`dcgeo`	Data Center's Geographical Location
`usecase`	Use Case
`prompt`	Prompt
`cloudgenai`	Generate AI Application

- A new *Generative AI* application category in *Security Profiles* > *Application Signatures*.
- *Generative AI* application signatures.
- Two new FortiView widgets for FortiGate data:
  - *AI Applications*
  - *AI Use Cases*

For more information about this feature, see Application control support for generative AI.

Agentic AI protocol support in FortiOS

Application control can now be used to detect agentic AI protocols, such as Model Context Protocol (MCP) and Agent-to-Agent Protocol (A2A). New AI-specific log fields and FortiView enhancements provide deeper insights into AI-driven activity, improving oversight and security for emerging AI workloads.

Example

This example describes how to configure FortiGate to detect the MCP and A2A protocols using application control. It also describes how to review the results in FortiGate logs and FortiView. It includes the following sections:

- Configuring FortiGate to detect agentic AI protocols on page 287
- Agentic AI protocol support in FortiOS on page 286
- Agentic AI protocol support in FortiOS on page 286

See also Limitations on page 292.

Configuring FortiGate to detect agentic AI protocols

On FortiGate, create an application control sensor, and apply the default application control profile in a firewall policy with deep-inspection enabled to monitor all applications.

Some signatures require deep inspection and some do not. You can use the GUI to identify whether a signature requires deep inspection.

After you configure FortiGate, you can monitor Agentic AI protocols:

- Agentic AI protocol support in FortiOS on page 286
- Agentic AI protocol support in FortiOS on page 286

Extended UTM log should be enabled to capture extra information in the log, such as AI Method, AI Arguments, AI Function, AI Message, AI URI, and so on. Extended UTM logging requires deep inspection.

```
config application list
    edit <profile name>
        set extended-log enable
    next
end
```

To create an application control sensor for generative AI in the GUI:

1. Go to *Security Profiles* > *Application Control*.
2. Click *Create New*.
3. Enter a *Name*, such as GenAI.
4. Set the *Generative AI* category to *Monitor*.
5. Configure other fields as needed.
6. Click *OK*.

To create an application control sensor for generative AI in the CLI:

```
config application list
    edit "GenAI"
        config entries
            edit 1
```

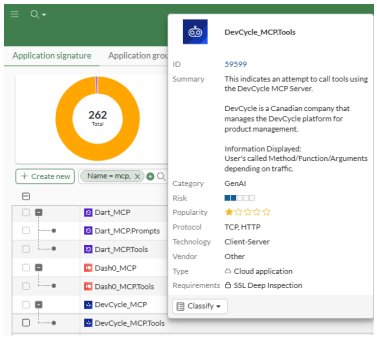
```
                set category 36
                set action pass
            next
        end
    next
end
```

Apply the default appctrl profile in firewall policy with deep-inspection to monitor all applications:

```
config firewall policy
    edit 1
        set uuid 13a56522-b12b-51f0-1d35-f6ecfe4e2568
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "new-deep-inspection"
        set application-list "g-default"
        set nat enable
    next
end
```

To identify which signatures require deep inspection:

1. Go to *Security Profiles* > *Application Signatures*, and locate the Generative AI signature, such as *DevCycle_MCP.Tools*.
2. Hover over the signature to display the popup.



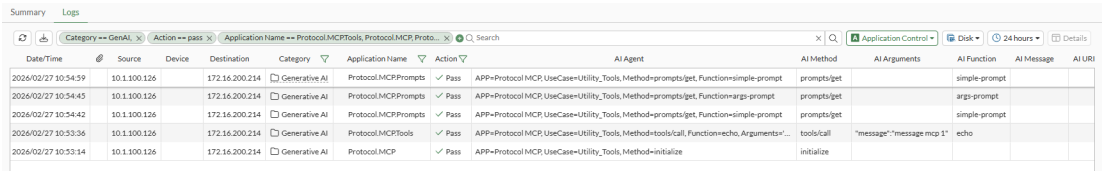
3. View the *Requirements* field to see whether *SSL Deep Inspection* is required.

Viewing MCP results in logs and FortiView

When a device on the network uses MCP tools and prompts, related MCP signatures (Protocol.MCP, Protocol.MCP.Tools, and Protocol.MCP.Prompts) are triggered on the FortiGate, and you can view the results in logs and FortiView.

To view logs in the GUI:

1. Go to *Log & Report > Security Events > Application Control Logs*.
   AI Method, AI Arguments, AI Function, AI Agent are displayed in the application logs:



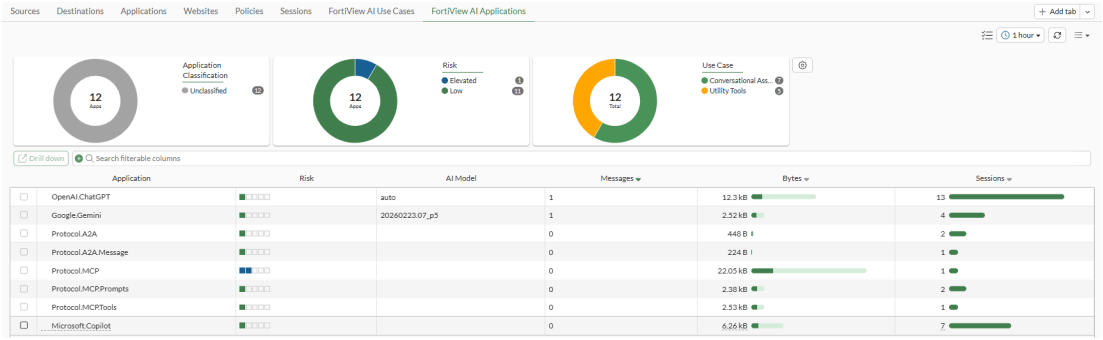
To view logs in the CLI:

The `aimethod`, `aifunc`, `aiargs`, and `cloudaiagent` fields are shown in application logs.

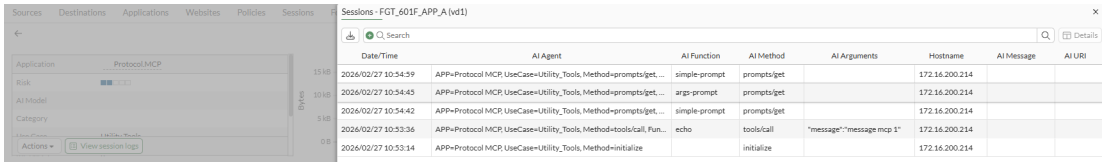
```
120: date=2026-02-27 time=10:54:42 eventtime=1772218482338739800 tz="-0800" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="vd1" appid=59249
srcip=10.1.100.126 srccountry="Reserved" dstip=172.16.200.214 dstcountry="Reserved" srcport=60047
dstport=3333 srcintf="port2" srcintfrole="undefined" dstintf="port1" dstintfrole="undefined"
proto=6 service="HTTP" direction="outgoing" policyid=1 poluuid="13a56522-b12b-51f0-1d35-
f6ecfe4e2568" policytype="policy" sessionid=454 applist="g-default" action="pass" appcat="GenAI"
app="Protocol.MCP.Prompts" hostname="172.16.200.214" incidentserialno=36700313 url="/mcp"
agent="node" httpmethod="POST" msg="GenAI: Protocol.MCP.Prompts" usecase="Utility_Tools"
aimethod="prompts/get" aifunc="simple-prompt" cloudaiagent="APP=Protocol MCP, UseCase=Utility_
Tools, Method=prompts/get, Function=simple-prompt" apprisk="low"

121: date=2026-02-27 time=10:53:36 eventtime=1772218415610539888 tz="-0800" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="vd1" appid=59072
srcip=10.1.100.126 srccountry="Reserved" dstip=172.16.200.214 dstcountry="Reserved" srcport=60037
dstport=3333 srcintf="port2" srcintfrole="undefined" dstintf="port1" dstintfrole="undefined"
proto=6 service="HTTP" direction="outgoing" policyid=1 poluuid="13a56522-b12b-51f0-1d35-
f6ecfe4e2568" policytype="policy" sessionid=439 applist="g-default" action="pass" appcat="GenAI"
app="Protocol.MCP.Tools" hostname="172.16.200.214" incidentserialno=36700297 url="/mcp"
agent="node" httpmethod="POST" msg="GenAI: Protocol.MCP.Tools" usecase="Utility_Tools"
aimethod="tools/call" aifunc="echo" aiargs="\"message\":\"message mcp 1\""
cloudaiagent="APP=Protocol MCP, UseCase=Utility_Tools, Method=tools/call, Function=echo,
Arguments='\"message\":\"message mcp 1\"'" apprisk="low"

122: date=2026-02-27 time=10:53:14 eventtime=1772218393538655508 tz="-0800" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="vd1" appid=59070
srcip=10.1.100.126 srccountry="Reserved" dstip=172.16.200.214 dstcountry="Reserved" srcport=60025
dstport=3333 srcintf="port2" srcintfrole="undefined" dstintf="port1" dstintfrole="undefined"
proto=6 service="HTTP" direction="outgoing" policyid=1 poluuid="13a56522-b12b-51f0-1d35-
f6ecfe4e2568" policytype="policy" sessionid=418 applist="g-default" action="pass" appcat="GenAI"
app="Protocol.MCP" hostname="172.16.200.214" incidentserialno=36700275 url="/mcp" agent="node"
httpmethod="POST" msg="GenAI: Protocol.MCP" usecase="Utility_Tools" aimethod="initialize"
cloudaiagent="APP=Protocol MCP, UseCase=Utility_Tools, Method=initialize" apprisk="elevated"
```

To view FortiView:

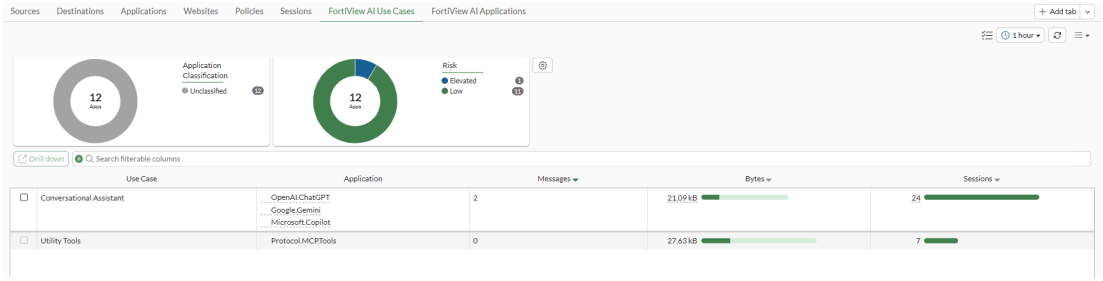
1. Go to *Dashboard > FortiView > FortiView AI Applications* to monitor MCP activities.
   A new *Use Case* chart is available at the top of the page.



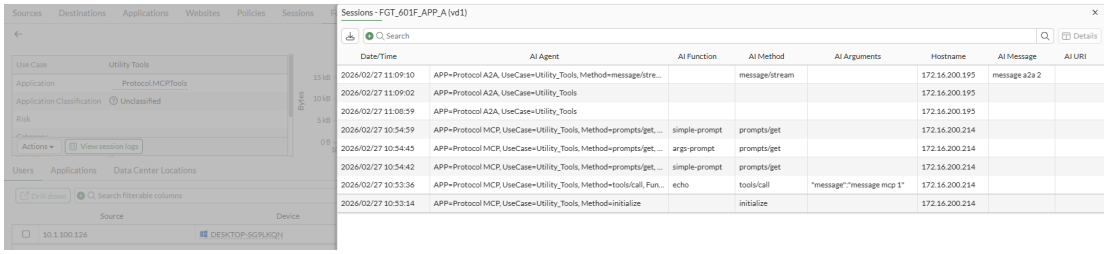
2. Drill down on an entry, such as *Protocol.MCP*. The *Sessions* pane is displayed, and it includes the following columns of information: *AI Agent*, *AI Function*, *AI Method*, *AI Arguments*, *Hostname*, and *AI URI*:



3. Click the *FortiView AI Use Cases* tab to view MCP activities. The type of use case is *Utility Tools*.



4. Drill down on a *Utility Tools* entry. The *Sessions* pane is displayed, and it includes the following columns of information: *AI Agent*, *AI Function*, *AI Method*, *AI Arguments*, *Hostname*, and *AI URI* for MCP protocol.

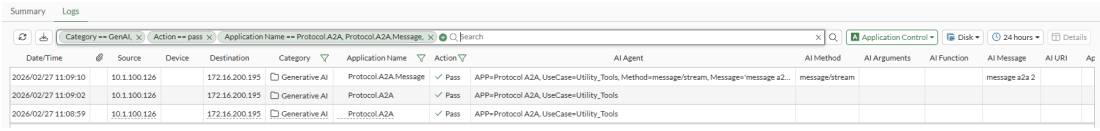


Viewing A2A results in logs and FortiView

When a device on the network uses A2A messages, related A2A signatures (Protocol.A2A and Protocol.A2A.Message) are triggered on the FortiGate, and you can view the results in logs and FortiView.

To view logs in the GUI:

1. Go to *Log & Report* > *Security Events* > *Application Control Logs*.
   AI Method, AI Message, AI Agent are shown in application logs:



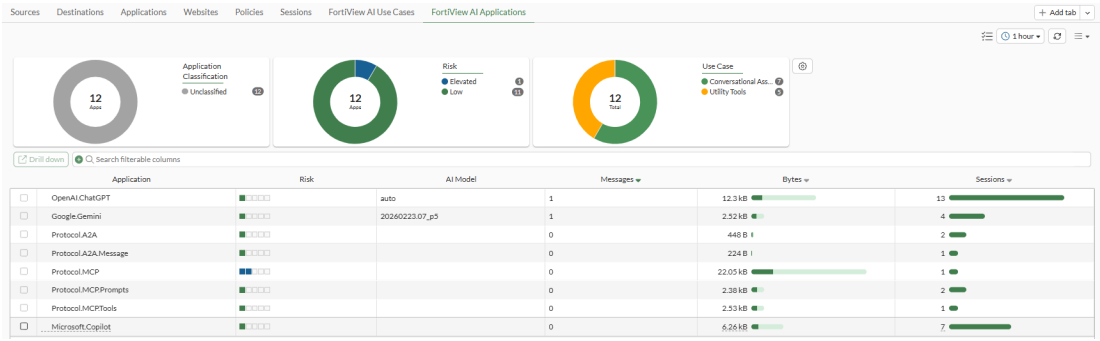
To view logs in the CLI:

The `aimethod`, `aimessage`, and `cloudaiagent` fields are shown in application logs.

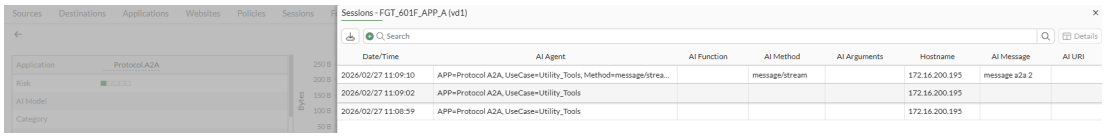
```
115: date=2026-02-27 time=11:09:10 eventtime=1772219349846174869 tz="-0800" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="vd1" appid=59162
srcip=10.1.100.126 srccountry="Reserved" dstip=172.16.200.195 dstcountry="Reserved" srcport=60174
dstport=10101 srcintf="port2" srcintfrole="undefined" dstintf="port1" dstintfrole="undefined"
proto=6 service="HTTP" direction="outgoing" policyid=1 poluuid="13a56522-b12b-51f0-1d35-
f6ecfe4e2568" policytype="policy" sessionid=828 applist="g-default" action="pass" appcat="GenAI"
app="Protocol.A2A.Message" hostname="172.16.200.195" incidentserialno=36700549
url="/v1/message:stream" agent="python-httpx/0.28.1" httpmethod="POST" msg="GenAI:
Protocol.A2A.Message" usecase="Utility_Tools" aimethod="message/stream" aimessage="message a2a 2"
cloudaiagent="APP=Protocol A2A, UseCase=Utility_Tools, Method=message/stream, Message='message a2a
2'" apprisk="low"

116: date=2026-02-27 time=11:09:02 eventtime=1772219341608208341 tz="-0800" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="vd1" appid=59157
srcip=10.1.100.126 srccountry="Reserved" dstip=172.16.200.195 dstcountry="Reserved" srcport=60173
dstport=10101 srcintf="port2" srcintfrole="undefined" dstintf="port1" dstintfrole="undefined"
proto=6 service="HTTP" direction="outgoing" policyid=1 poluuid="13a56522-b12b-51f0-1d35-
f6ecfe4e2568" policytype="policy" sessionid=827 applist="g-default" action="pass" appcat="GenAI"
app="Protocol.A2A" hostname="172.16.200.195" incidentserialno=36700547 url="/.well-known/agent-
card.json" agent="python-httpx/0.28.1" httpmethod="GET" msg="GenAI: Protocol.A2A"
usecase="Utility_Tools" cloudaiagent="APP=Protocol A2A, UseCase=Utility_Tools" apprisk="low"
```

To view FortiView:

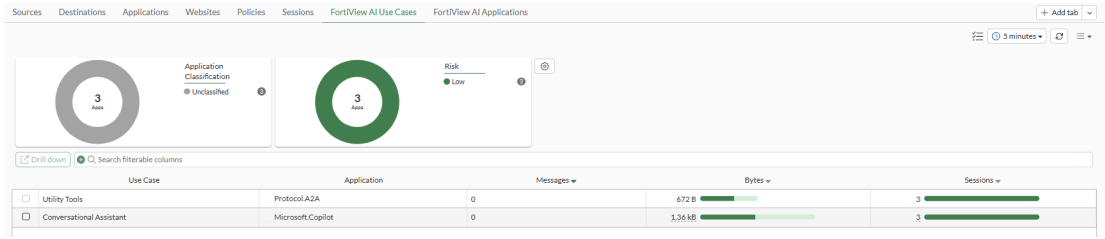
1. Go to *Dashboard* > *FortiView* > *FortiView AI Applications* to monitor A2A activities.
   A new *Use Case* chart is available at the top of the page.



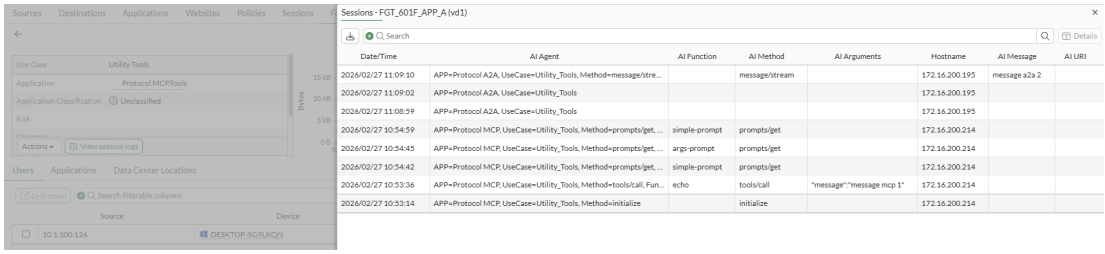
2. Drill down on an entry, such as *Protocol.Z2A*. The *Sessions* pane is displayed, and it includes the following columns of information: *AI Agent*, *AI Function*, *AI Method*, *AI Arguments*, *Hostname*, and *AI URI*:



3. Click the *FortiView AI Use Cases* tab to view A2A activities. The type of use case is *Utility Tools*.



4. Drill down on a *Utility Tools* entry. The *Sessions* pane is displayed, and it includes the following columns of information: *AI Agent*, *AI Function*, *AI Method*, *AI Arguments*, *Hostname*, and *AI URI* for A2A protocol.



Limitations

- For the AIAP database to be updated, ensure that at least one firewall policy is utilizing an application control profile. Database updates require FMWR contract.

  To review the generative AI database:

  ```
  # diagnose autoupdate versions | grep -A 6 GenAI
  GenAI Application Definitions
  ---------
  Version: 33.00033 signed
  Contract Expiry Date: Thu Jan  3 2030
  Last Updated using scheduled update on Tue Jun 24 20:59:43 2025
  Last Update Attempt: Tue Jun 24 23:10:03 2025
  Result: No Updates
  ```

- Proxy inline-ips does not support MCP and A2A protocol.

  Inline IPS must be disabled when using a firewall policy with an inspection-mode proxy, as well as an explicit proxy policy.

  ```
  config ips settings
      set proxy-inline-ips disable
  end
  ```

- NGFW security policy does not support MCP and A2A protocol.

VPN

This section includes information about VPN related new features:

- IPsec and Agentless VPN on page 293

IPsec and Agentless VPN

This section includes information about IPsec and Agentless related new features:

- New Cloud SDN Orchestration VPN wizard for AWS on page 293
- Support matching firewall policies and policy routes based on source IP geography of dial-up IPsec remote users on page 307
- Send security posture tags from FortiClient to FortiOS directly as JWT on page 311
- TLS 1.3 based VPN over TCP on page 316
- Native VPN remote access support in VPN wizard on page 323
- Default to IPAM addressing for remote access VPNs in VPN Wizard on page 341
- Display interface and tunnel alias in various diagnose and get commands on page 346
- Support SM3/SM4 cryptographic algorithms for IKEv1/ IKEv2 on page 350
- Show only recommended IKE proposals by default on page 356

New Cloud SDN Orchestration VPN wizard for AWS

This information is also available in the FortiOS 8.0 Administration Guide:

- Cloud SDN Orchestration VPN wizard for AWS

A new Cloud SDN Orchestration VPN wizard is added to simplify configuration of a VPN tunnel between a FortiGate and a virtual private gateway or transit gateway on AWS. When a FortiGate has an SDN connector with the proper permissions established with AWS, the VPN wizard creates the FortiGate VPN configurations and pushes the necessary customer gateway and VPN tunnel configurations to AWS under the configured VPC. This reduces the chance of mis-configurations and the number of steps needed to configure a VPN tunnel.

A new `config sys sdn-vpn` command is available:

```
config sys sdn-vpn
    edit <name>
        set *sdn <string>
        set *remote-type {vgw | tgw}
        set *routing-type {static | dynamic}
        set *vgw-id <string>
```

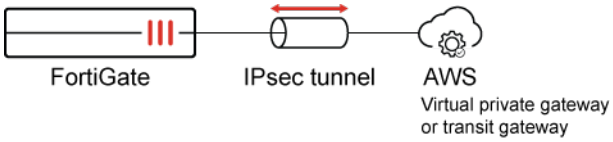
```
        set nat-traversal {enable | disable}
        set *tunnel-interface <string>
        set *internal-interface <string>
        set *local-cidr <class_ip&net_netmask>
        set *remote-cidr <class_ip&net_netmask>
        set cgw-name <string>
        set psksecret <passwd>
    next
end
```

Option	Description
`config sys sdn-vpn`	Configure public cloud VPN service.
`edit <name>`	Public cloud VPN name.
`*sdn <string>`	SDN connector name.
`*remote-type {vgw | tgw}`	Type of remote device: • `vgw`: virtual private gateway • `tgw`: transit gateway
`*routing-type {static | dynamic}`	Type of routing: • `static`: static routing • `dynamic`: dynamic routing
`*vgw-id <string>`	Virtual private gateway ID.
`nat-traversal {enable | disable}`	Enable/disable use for NAT traversal. Please enable if your FortiGate device is behind a NAT/PAT device.
`*tunnel-interface <string>`	Tunnel interface with public IP.
`*internal-interface <string>`	Internal interface with local subnet.
`*local-cidr <class_ip&net_netmask>`	Local subnet address and subnet mask.
`*remote-cidr <class_ip&net_netmask>`	Remote subnet address and subnet mask.
`cgw-name <string>`	AWS customer gateway name to be created.
`psksecret <passwd>`	Pre-shared secret for PSK authentication. Auto-generated if not specified.

New diagnose commands are available:

```
# diagnose debug app sdnvpnd -1

# diagnose test application awsd 5
5. list the settings created by sdn-vpn widgets
```

Example



This example demonstrates how to use the Cloud SDN Orchestration VPN wizard on FortiGate to create VPN configurations and push the necessary customer gateway and VPN tunnel configurations to AWS under the configured VPC.

Following are the requirements for using the wizard:

- On AWS, create IAM and certificates for use with SDN connector.
- On FortiGate, create a regional SDN connector for AWS and establish a connection to AWS. The wizard uses the AWS SDN connector. See AWS SDN connector using access keys for information about creating an AWS SDN connector.
- On AWS, ensure the FortiGate SDN connector for AWS has the following permissions required by the wizard:

```
- ec2:DescribeVpcs
- ec2:DescribeSubnets
- ec2:DescribeRouteTables
- ec2:CreateRoute
- ec2:DeleteRoute
- ec2:DescribeCustomerGateways
- ec2:CreateCustomerGateway
- ec2:DeleteCustomerGateway
- ec2:DescribeVpnGateways
- ec2:EnableVgwRoutePropagation
- ec2:DescribeTransitGateways
- ec2:CreateTransitGatewayRouteTable
- ec2:DeleteTransitGatewayRouteTable
- ec2:EnableTransitGatewayRouteTablePropagation
- ec2:DisassociateTransitGatewayRouteTable
- ec2:AssociateTransitGatewayRouteTable
- ec2:DescribeTransitGatewayRouteTables
- ec2:CreateTransitGatewayRoute
- ec2:DescribeTransitGatewayAttachments
- ec2:DescribeTransitGatewayVpcAttachments
- ec2:DescribeVpnConnections
- ec2:CreateVpnConnection
- ec2:DeleteVpnConnection
- ec2:CreateVpnConnectionRoute
- ec2:DeleteVpnConne
```

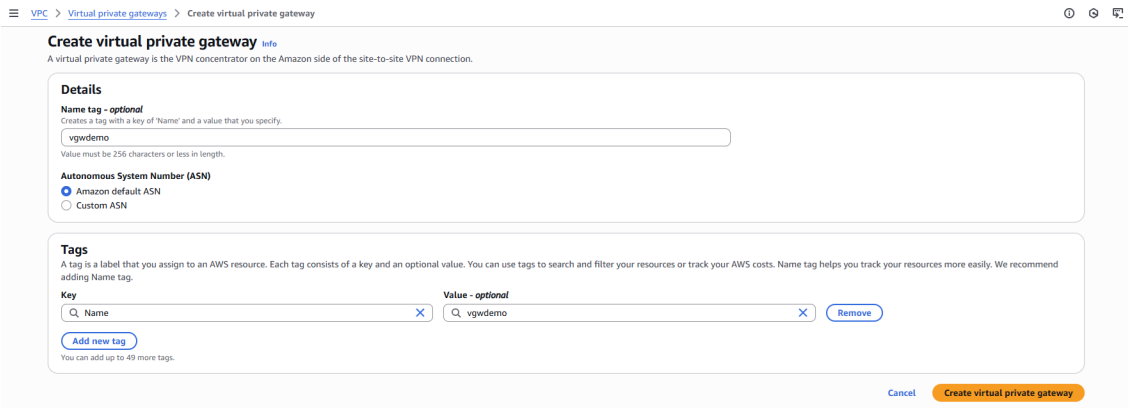
- On AWS, provision a VPC with subnets.
- On AWS, provision one of the following and attach it to the VPC:
  - Virtual private gateway (VPG)
    This example uses a VPG.
  - Transit gateway(TGW)
    When you attach a TGW to the VPC, ensure you set *Type* to *VPN*.

- Use FortiGate local admin privileges or higher

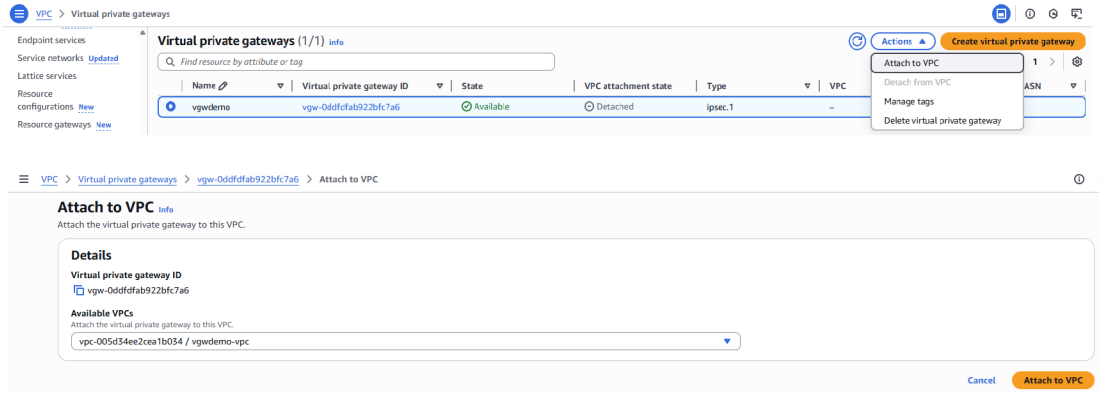
Security Groups are manually edited after the wizard is finished to allow access from on-premise network.

To create VPN tunnels using the Cloud SDN Orchestration VPN wizard:

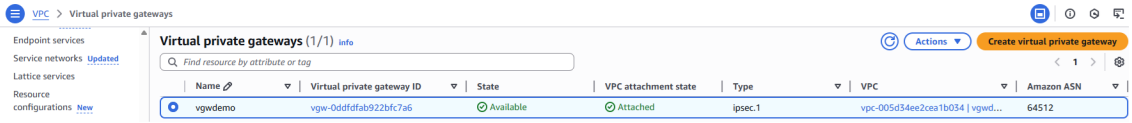
1. On AWS, create a new virtual private gateway and associate it with the target AWS VPC:
   a. Go to an existing VPC or create a new VPC.
   b. In the *VPC* side menu, go to *Virtual private network (VPN) > Virtual private gateways*. Click *Create virtual private gateway*.



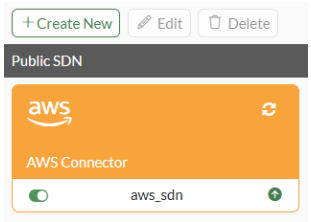
   c. After the VPG is created, select it and click *Actions*. Select *Attach to VPC*.



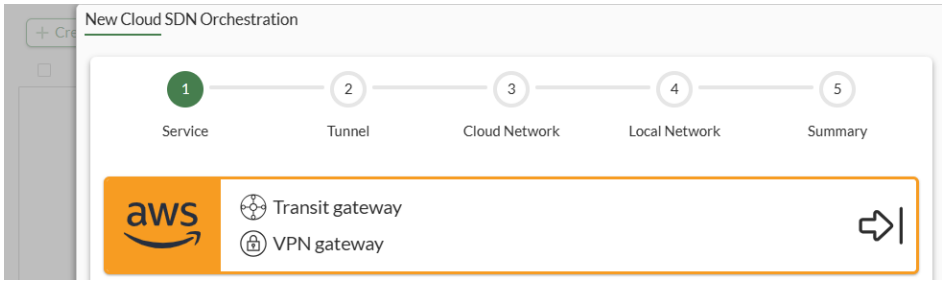
   d. Choose the VPC to attach this VPG to. Click *Attach to VPC*.
   The new virtual private gateway is ready to for FortiGate configurations.



2. On FortiGate, go to *Security Fabric > External Connectors*, and click the *AWS Connector* to check that *Status* is *Up*.
   This step assumes that the AWS Connector has already been configured.

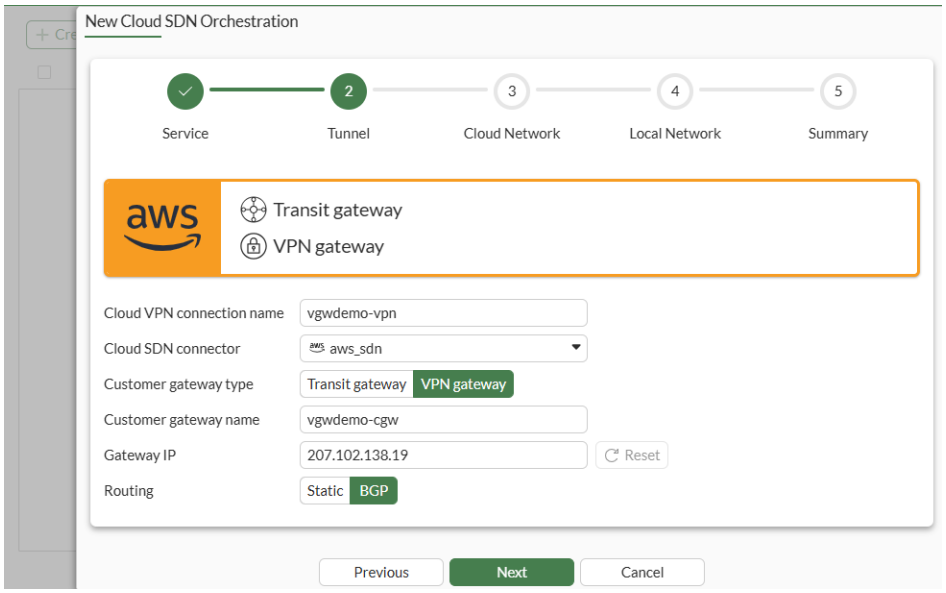


3. On FortiGate, use the Cloud SDN Orchestration wizard to create VPN tunnels:
   a. Go to *VPN > Cloud SDN Orchestration*, and click *Create new*. The *New Cloud SDN Orchestration* wizard is displayed on step 1 for *Service*.



   b. Click the AWS icon to confirm orchestration for AWS and to display step 2 for *Tunnel*.
   c. On the *Tunnel* page, complete the options, and click *Next*.

Option	Description
Cloud VPN connection name	Enter a name for the configuration.
Cloud SDN connector	Select the previously created AWS SDN connector.
Customer gateway type	Select the type of gateway created on AWS (*Transit gateway* or *VPN gateway*).
Customer gateway name	Enter the name of the gateway created on AWS.
Gateway IP	The IP address is automatically populated.
Routing	Select the type of routing to use for the tunnels (*Static* or *BGP*).



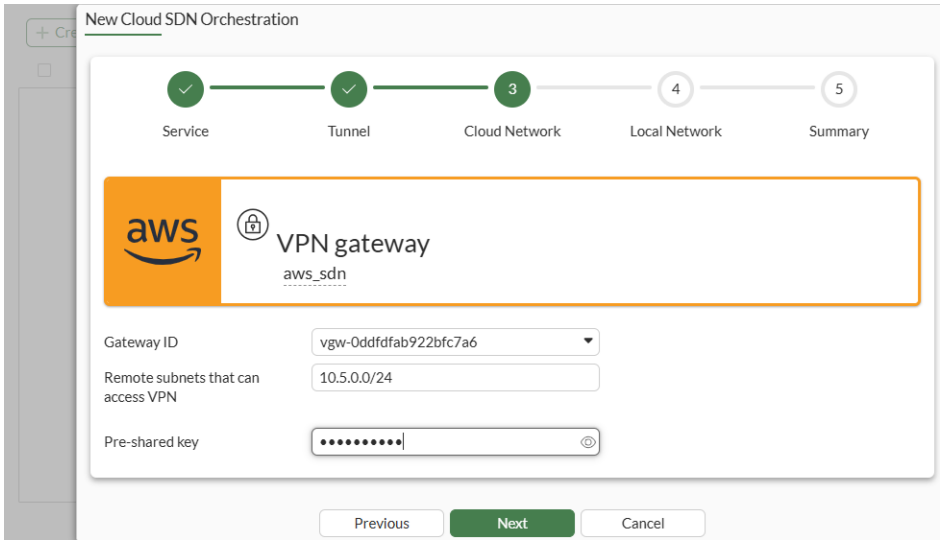
If permissions for the AWS SDN connector cannot be validated, the following error may display, and you my be unable to advance to the next wizard steps until you correct the permissions:

Failed to validate permission, ensure the SDN connector credentials are correct and you have the required permission from the cloud service.

The wizard moves to step 3 for *Cloud Network*.

d. On the *Cloud Network* page, complete the options, and click *Next*.

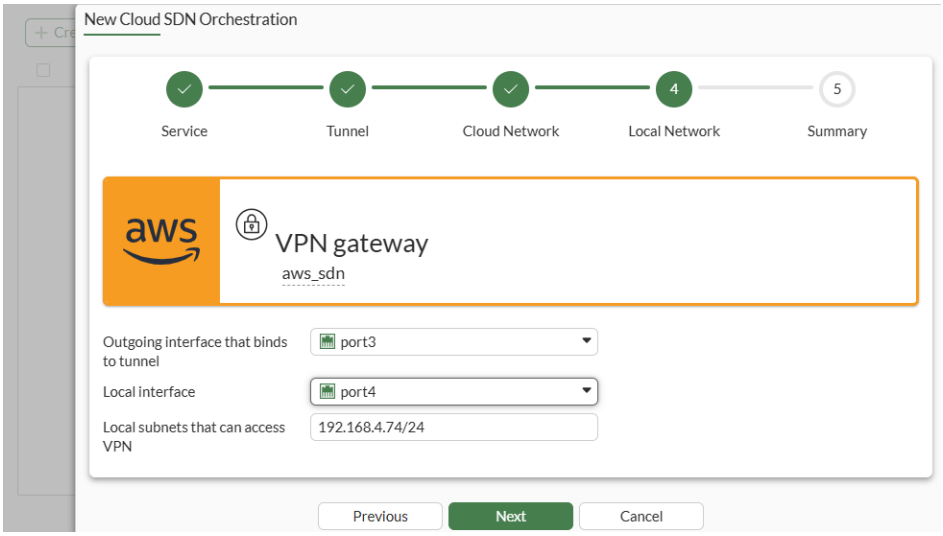
Option	Description
Gateway ID	Select the gateway ID created in AWS.
Remote subnets that can access VPN	Enter the AWS subnets that can access VPN.
Pre-shared key	Enter the pre-shared key (PSK) for PSK authentication. Auto-generated if not specified.



The wizard moves to step 4 for *Local Network*.

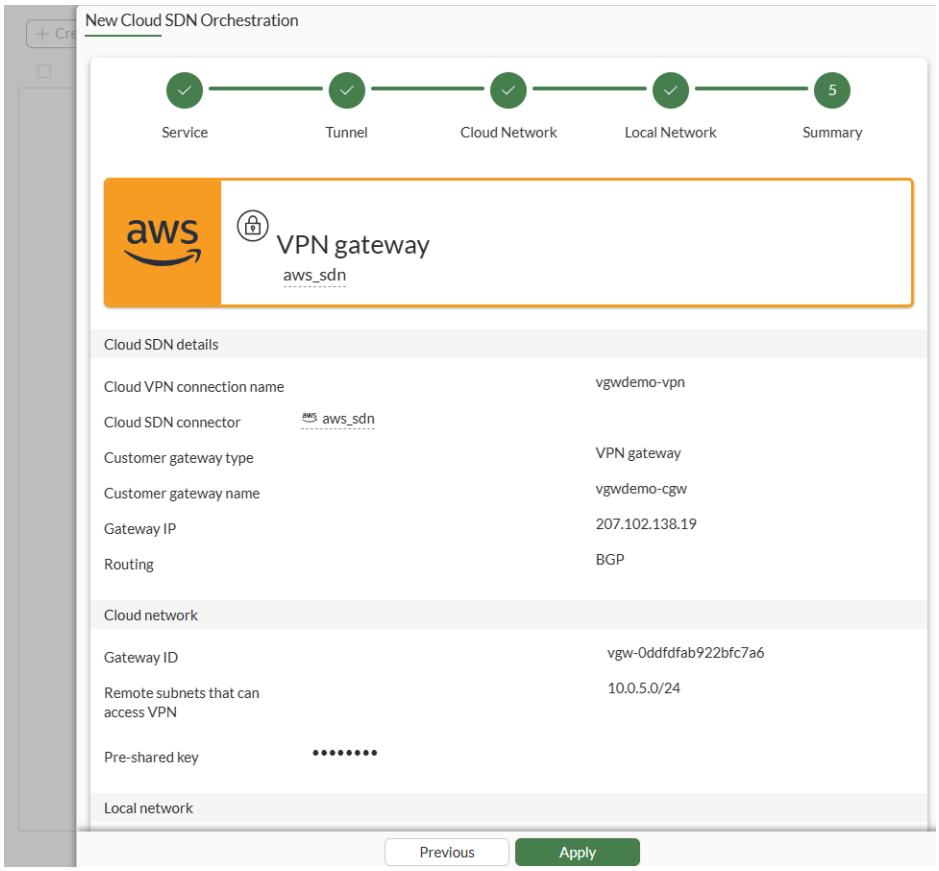
e. On the *Local Network* page, complete the options, and click *Next*.

Option	Description
Outgoing interface that binds to tunnel	Select the FortiGate outbound interface for VPN.
Local interface	Select the FortiGate inbound interface for VPN.
Local subnets that can access VPN	Enter the FortiGate subnets that can access VPN.



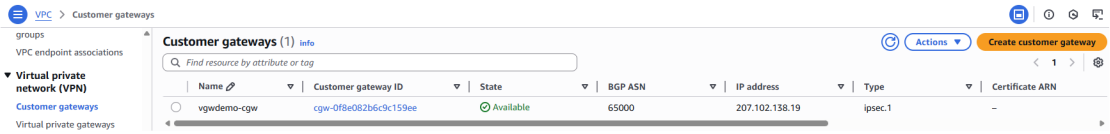
The wizard moves to step 5 for *Summary*.

f. On the *Summary* page, review the settings, and click *Apply*.

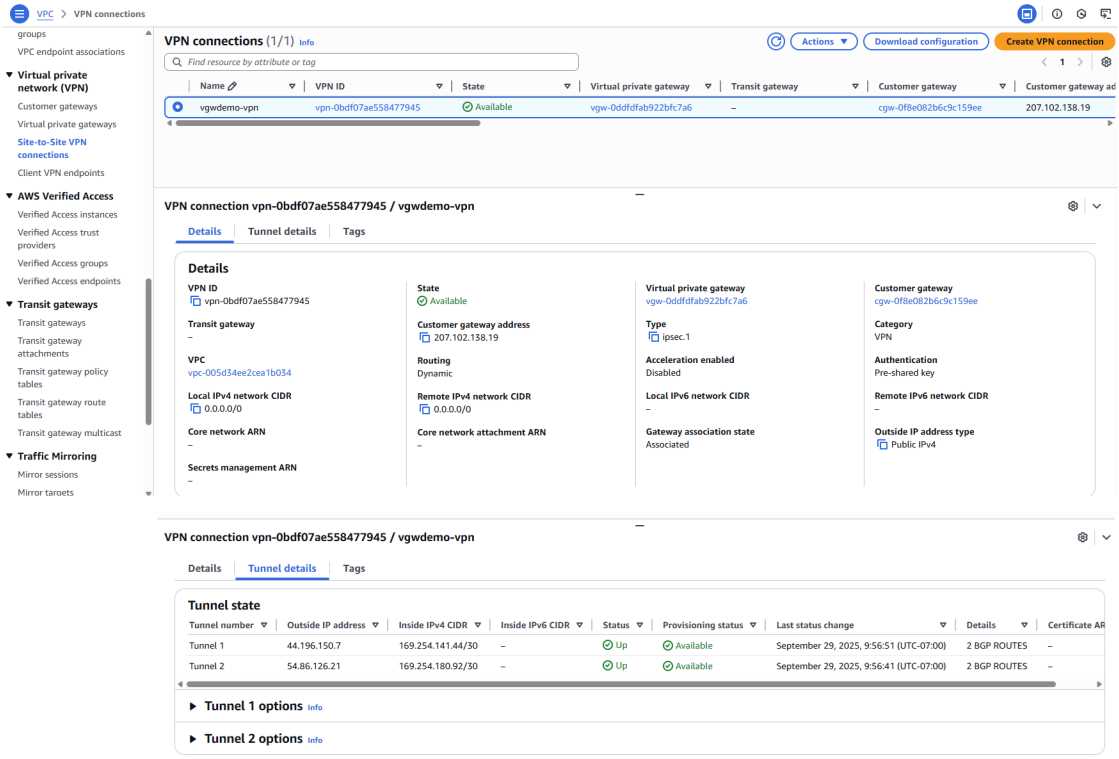


The wizard creates the VPN tunnels and pushes the Customer Gateway and VPN tunnel configurations to AWS under the configured VPC.

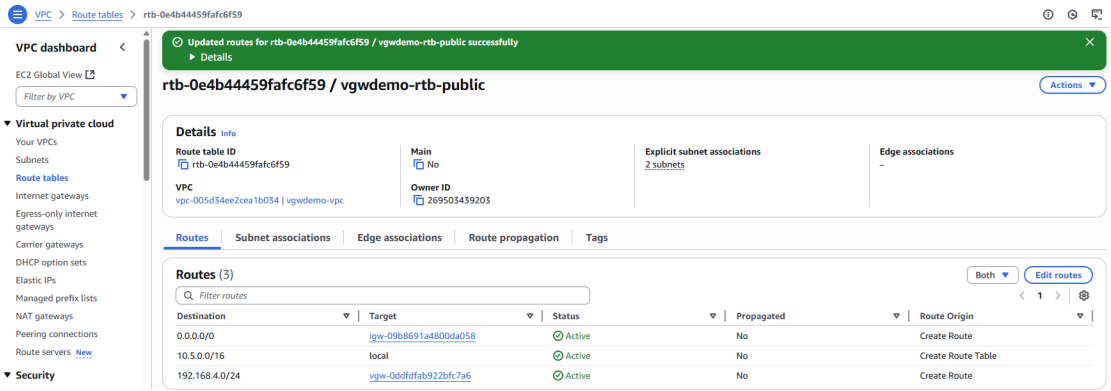
4. On AWS console, ensure the FortiGate wizard successfully created the *Customer gateway* and *VPN connection*.
   a. Go to *Virtual private network (VPN) > Customer gateways*.



   b. Under *Virtual private network (VPN)*, click *Site-to-Site VPN connections*.



5. On AWS, go to *Virtual private cloud > Route tables*, and add route entry for the routing table of AWS VPC target subnet, and make the traffic to local FGW 192.168.4.0/24 use the VGW as the gateway.



To verify the VPN tunnel configurations in the GUI:

1. Go to *VPN > Cloud SDN Orchestration*. Select the configuration, and click *View*:

To verify the VPN tunnel configurations in the CLI:

1. Show the `system sdn-vpn` settings:

```
show system sdn-vpn
config system sdn-vpn
    edit "vgwdemo-vpn"
        set sdn "aws_sdn"
        set vgw-id "vgw-0ddfdfab922bfc7a6"
        set cgw-gateway 207.102.138.19
        set tunnel-interface "port3"
        set internal-interface "port4"
        set local-cidr 192.168.4.0 255.255.255.0
        set remote-cidr 10.5.0.0 255.255.255.0
        set cgw-name "vgwdemo-cgw"
        set psksecret xxx
    next
end
```

2. Show the `vpn ipsec phase1-interface` settings:

```
show vpn ipsec phase1-interface
config vpn ipsec phase1-interface
    edit "vgwdemo-vpn-0"
        set interface "port3"
        set ike-version 2
        set keylife 28800
        set peertype any
        set net-device disable
        set proposal aes128-sha1
        set comments "Automatically generated by site-to-site VPN wizard."
        set dhgrp 2
        set wizard-type cloud-sdn-orchestration
        set transport auto
        set remote-gw 44.196.150.7
        set psksecret xxx
    next
    edit "vgwdemo-vpn-1"
        set interface "port3"
        set ike-version 2
        set keylife 28800
        set peertype any
        set net-device disable
        set proposal aes128-sha1
        set comments "Automatically generated by site-to-site VPN wizard."
        set dhgrp 2
        set wizard-type cloud-sdn-orchestration
        set transport auto
        set remote-gw 54.86.126.21
        set psksecret xxx
    next
end
```

3. Show the `vpn ipsec phase2-interface` settings:

```
show vpn ipsec phase2-interface
config vpn ipsec phase2-interface
    edit "vgwdemo-vpn-0"
        set phase1name "vgwdemo-vpn-0"
        set proposal aes128-sha1
        set dhgrp 2
        set auto-negotiate enable
        set comments "Automatically generated by site-to-site VPN wizard."
        set keylifeseconds 3600
    next
    edit "vgwdemo-vpn-0-inside"
        set phase1name "vgwdemo-vpn-0"
        set proposal aes128-sha1
        set dhgrp 2
        set auto-negotiate enable
        set comments "Automatically generated by site-to-site VPN wizard."
        set keylifeseconds 3600
    next
    edit "vgwdemo-vpn-1"
        set phase1name "vgwdemo-vpn-1"
        set proposal aes128-sha1
        set dhgrp 2
        set auto-negotiate enable
        set comments "Automatically generated by site-to-site VPN wizard."
        set keylifeseconds 3600
    next
    edit "vgwdemo-vpn-1-inside"
        set phase1name "vgwdemo-vpn-1"
        set proposal aes128-sha1
        set dhgrp 2
        set auto-negotiate enable
        set comments "Automatically generated by site-to-site VPN wizard."
        set keylifeseconds 3600
    next
end
```

4. Show the `system interface vgwdemo-vpn-0` settings:

```
show system interface vgwdemo-vpn-0
config system interface
    edit "vgwdemo-vpn-0"
        set vdom "root"
        set ip 169.254.141.46 255.255.255.255
        set allowaccess ping
        set type tunnel
        set tcp-mss 1379
        set remote-ip 169.254.141.45 255.255.255.252
        set description "Automatically generated by site-to-site VPN wizard."
        set snmp-index 16
        set interface "port3"
        set mtu-override enable
        set mtu 1427
```

```
    next
end
```

5. Show the `system interface vgwdemo-vpn-1` settings:

```
show system interface vgwdemo-vpn-1
config system interface
    edit "vgwdemo-vpn-1"
        set vdom "root"
        set ip 169.254.180.94 255.255.255.255
        set allowaccess ping
        set type tunnel
        set tcp-mss 1379
        set remote-ip 169.254.180.93 255.255.255.252
        set description "Automatically generated by site-to-site VPN wizard."
        set snmp-index 17
        set interface "port3"
        set mtu-override enable
        set mtu 1427
    next
end
```

6. Show the `config firewall policy` settings:

```
show firewall policy
config firewall policy
    edit 1
        set name "AWS_IPSEC_WIZ_vgwdemo-vpn-0_in"
        set uuid 0c945192-9d55-51f0-2ca8-e0de305845d8
        set srcintf "vgwdemo-vpn-0"
        set dstintf "port4"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set comments "Automatically generated by site-to-site VPN wizard."
    next
    edit 2
        set name "AWS_IPSEC_WIZ_vgwdemo-vpn-0_out"
        set uuid 0c95b67c-9d55-51f0-7f37-75b1fe7ad6fa
        set srcintf "port4"
        set dstintf "vgwdemo-vpn-0"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set comments "Automatically generated by site-to-site VPN wizard."
    next
    edit 3
        set name "AWS_IPSEC_WIZ_vgwdemo-vpn-1_in"
```

```
        set uuid 0c9d02f6-9d55-51f0-bb0d-cdd46a5f0145
        set srcintf "vgwdemo-vpn-1"
        set dstintf "port4"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set comments "Automatically generated by site-to-site VPN wizard."
    next
    edit 4
        set name "AWS_IPSEC_WIZ_vgwdemo-vpn-1_out"
        set uuid 0c9f31a2-9d55-51f0-a940-be65d2761bf9
        set srcintf "port4"
        set dstintf "vgwdemo-vpn-1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set comments "Automatically generated by site-to-site VPN wizard."
    next
end
```

7. Show the `config router bgp` settings:

```
show router bgp
config router bgp
    set as 65000
    set router-id 207.102.138.19
    config neighbor
        edit "169.254.141.45"
            set capability-default-originate enable
            set remote-as 64512
        next
        edit "169.254.180.93"
            set capability-default-originate enable
            set remote-as 64512
        next
    end
    config network
        edit 1
            set prefix 192.168.4.0 255.255.255.0
        next
    end
    config redistribute "connected"
    end
    config redistribute "rip"
    end
    config redistribute "ospf"
    end
```

```
    config redistribute "static"
    end
    config redistribute "isis"
    end
    config redistribute6 "connected"
    end
    config redistribute6 "rip"
    end
    config redistribute6 "ospf"
    end
    config redistribute6 "static"
    end
    config redistribute6 "isis"
    end
end
```

8. View the VPN tunnel list:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=vgwdemo-vpn-0 ver=2 serial=1 172.16.200.74:4500->44.196.150.7:4500 nexthop=172.16.200.254
tun_id=44.196.150.7 tun_id6=::44.196.150.7 status=up dst_mtu=1500 weight=1 country=US
bound_if=9 real_if=9 lgwy=static/1 tun=intf mode=auto/1 encap=none options[0x228]=npu frag-rfc
 run_state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=5 ilast=6 olast=6 ad=/0
stat: rxp=912 txp=914 rxb=55715 txb=56934
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=keepalive draft=0 interval=10 remote_port=4500
fec: egress=0 ingress=0
proxyid=vgwdemo-vpn-0 proto=0 sa=1 ref=3 serial=1 auto-negotiate
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=38203 type=00 soft=0 mtu=1422 expire=2295/0B replaywin=2048
       seqno=dd esn=0 replaywin_lastseq=000000dc qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=3329/3600
  dec: spi=ba41a824 esp=aes key=16 67e826b2dd5d119f39b8a069593cb858
       ah=sha1 key=20 f0f633f79b8148a76a0b5dfa6685101958c462f7
  enc: spi=cbe6508c esp=aes key=16 f5e5b7e8a4a1285731167088f3e64783
       ah=sha1 key=20 48590b92b6660c8005407580229d63d6ea91903f
  dec:pkts/bytes=220/13397, enc:pkts/bytes=220/30032
  npu_flag=00 npu_rgwy=0.0.0.0:65535 npu_lgwy=0.0.0.0:65535npu_selid=0
  dec_npuid=0 enc_npuid=0 dec_engid=-1 enc_engid=-1 dec_saidx=-1 enc_saidx=-1
------------------------------------------------------
name=vgwdemo-vpn-1 ver=2 serial=2 172.16.200.74:4500->54.86.126.21:4500 nexthop=172.16.200.254
tun_id=54.86.126.21 tun_id6=::54.86.126.21 status=up dst_mtu=1500 weight=1 country=US
bound_if=9 real_if=9 lgwy=static/1 tun=intf mode=auto/1 encap=none options[0x228]=npu frag-rfc
 run_state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=5 ilast=3 olast=3 ad=/0
stat: rxp=921 txp=925 rxb=56427 txb=57795
```

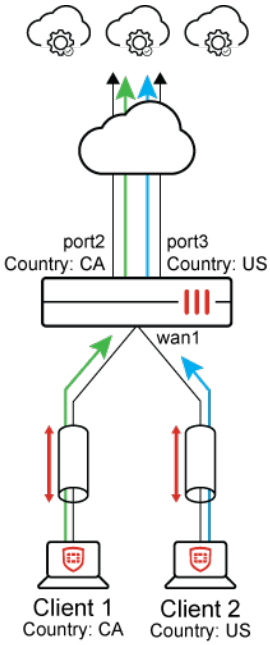
```
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=keepalive draft=0 interval=10 remote_port=4500
fec: egress=0 ingress=0
proxyid=vgwdemo-vpn-1 proto=0 sa=1 ref=3 serial=1 auto-negotiate
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=38203 type=00 soft=0 mtu=1422 expire=2320/0B replaywin=2048
       seqno=d8 esn=0 replaywin_lastseq=000000d7 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=3332/3600
  dec: spi=ba41a825 esp=aes key=16 e056c5bc5ef6ec11a4ad10142872db53
       ah=sha1 key=20 77e2ac173d9c80e4676cc6f922b2843a380e6fbd
  enc: spi=c44fe363 esp=aes key=16 9548351f37e355bbc60e8a41cc38bd05
       ah=sha1 key=20 327ce26e4d4fc94d765fec7a0fec0e8d7f87d35b
  dec:pkts/bytes=215/13099, enc:pkts/bytes=215/29344
  npu_flag=00 npu_rgwy=0.0.0.0:65535 npu_lgwy=0.0.0.0:65535npu_selid=0
  dec_npuid=0 enc_npuid=0 dec_engid=-1 enc_engid=-1 dec_saidx=-1 enc_saidx=-1
```

Support matching firewall policies and policy routes based on source IP geography of dial-up IPsec remote users

This information is also available in the FortiOS 8.0 Administration Guide:

- Geography-based tunneled internet browsing

A FortiGate VPN gateway can be configured to route egress traffic to the Internet through a specific interface (port2 or port3) based on the originating geolocation of the ingress client traffic (Canada or US). This can be achieved using FortiClient Secure Internet Access (SIA) configuration together with a policy route to route traffic to the proper egress interface.



Previously, using the supported configuration of Matching IPsec tunnel gateway based on address parameters, IPsec clients with public source IP addresses originating from a specific geolocation (country/region) were matched by specifying the source geography within the IPsec tunnel configuration itself, namely, in the Phase 1 configuration. This meant that for every geolocation to be matched, a new IPsec tunnel, consisting of Phase 1 and Phase 2 settings, needed to be configured.

FortiOS has added support for specifying source geography addresses in firewall policies and policy routes on a FortiGate configured for dial-up IPsec remote access. With this enhancement, a single IPsec tunnel can be used while matching client source IP addresses based on geolocation can be achieved by configuring source geography addresses in the firewall policies and policy routes.

Example

In this example, the client PC1 behind FGTA with NAT enabled is located in Canada. FGTD is configured as the VPN Gateway with:

- Firewall source address (geo-CA) configured with CA country
- VPN IPsec tunnel
- Firewall policy with geo-CA selected as the source address

This example shows how a firewall policy and policy route involving a dialup IPsec tunnel can both be matched based on the source IP geolocation of incoming dialup clients. FortiOS internally keeps track of the original geolocation of incoming dialup traffic and integrates with firewall policies and with policy routing.

To configure FGTD:

1. On FGTD, configure a firewall address with a geolocation country/region:

```
config firewall address
    edit "geo-CA"
        set uuid 3c2d2398-9d91-51f0-4c8d-58f43ee65c8a
        set type geography
        set country "CA"
    next
end
```

2. Configure an IPsec VPN tunnel:

```
config vpn ipsec phase1-interface
    edit "Dialup"
        set type dynamic
        set interface "wan1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
```

```
        set dhgrp 20 21
        set wizard-type dialup-fortigate
        set transport auto
        set psksecret ENC *
    next
end
config vpn ipsec phase2-interface
    edit "Dialup"
        set phase1name "Dialup"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

3. Configure a firewall policy and select `geo-CA` as the source address:

```
config firewall policy
    edit 1
        set name "vpn_Dialup_local"
        set uuid 9378cc0e-9d8e-51f0-4e0d-a90274db10fb
        set srcintf "port1"
        set dstintf "Dialup"
        set action accept
        set srcaddr "Dialup_local"
        set dstaddr "Dialup_remote"
        set schedule "always"
        set service "ALL"
        set comments "VPN: Dialup -- Created by VPN wizard"
    next
    edit 2
        set name "vpn_Dialup_remote"
        set uuid 937abc9e-9d8e-51f0-c288-04ad4bb6f72f
        set srcintf "Dialup"
        set dstintf "port1"
        set action accept
        set srcaddr "geo-CA"
        set dstaddr "192.168.5.0"
        set schedule "always"
        set service "ALL"
        set comments "VPN: Dialup -- Created by VPN wizard"
    next
end
```

4. Configure a policy route to ensure client traffic from Canada is routed out of port2:

```
config router policy
    edit 1
        set input-device "Dialup"
        set srcaddr "geo-CA"
        set dstaddr "all"
        set gateway 198.51.100.2
        set output-device "port2"
```

```
    next
end
```

5. Verify using the VPN IKE gateway list and the VPN tunnel list:
   - VPN IKE gateway list:

```
# diagnose vpn ike gateway list

vd: root/0
name: Dialup_0
version: 2
interface: port13 19
addr: 198.51.100.1:500 -> 192.0.2.1:500
tun_id: 192.0.2.1/::10.0.0.7
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 263s ago
peer-id: 192.0.2.1
peer-id-auth: no
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 1/2  established 1/2  time 0/0/0 ms

  id/spi: 5 7608e6dac7750e94/564ba227bdbd6dee
  direction: responder
  status: established 263-263s ago = 10ms
  proposal: aes128-sha256
  child: no
  SK_ei: 933120ad1055a39a-1c27f7ed4dfb258a
  SK_er: 32520d03666c4cbe-b79c65eff6b69c12
  SK_ai: 37da2c44ddb06cef-ab93d092d0c3ef7b-a182b40e35c98c60-2a6937a90a089707
  SK_ar: 4afc71586c84b9bd-e7e6a247bde21748-072dbdeb992b005c-c8466694bd177444
  PPK: no
  message-id sent/recv: 0/11
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/85866
  DPD sent/recv: 00000000/00000000
  peer-id: 192.0.2.1
```

   - VPN tunnel list:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=Dialup_0 ver=2 serial=7 198.51.100.1:0->192.0.2.1:0 nexthop=198.51.100.2 tun_
id=192.0.2.1 tun_id6=::10.0.0.7 status=up dst_mtu=1500 weight=1 country=CA
bound_if=19 real_if=19 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none options
[0x22a8]=npu rgwy-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0
```

```
parent=Dialup index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=105 olast=106 ad=/0
stat: rxp=159 txp=4 rxb=24588 txb=336
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=Dialup proto=0 sa=1 ref=2 serial=2 add-route
  src: 0:192.168.5.0-192.168.5.255:0
  dst: 0:10.1.100.0-10.1.100.255:0
  SA:  ref=6 options=6a6 type=00 soft=0 mtu=1438 expire=43075/0B replaywin=2048
       seqno=405 esn=0 replaywin_lastseq=00000013 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43186/43200
  dec: spi=9dc42e9e esp=aes key=16 e3c82f72ce497d27736b120cda74f23a
       ah=sha256 key=32 b9702181e33b8b4986b4d3b847bfe0307231ee423dc8651085dcfe58d88771c8
  enc: spi=c8e99d60 esp=aes key=16 c40474714b560d0855653089b4cea604
       ah=sha256 key=32 45bf85337226f86b9a7774657170b206db1df5b40a9eb31a3ec2bb79949c9b32
  dec:pkts/bytes=2/168, enc:pkts/bytes=4/624
  npu_flag=03 npu_rgwy=192.0.2.1:0 npu_lgwy=198.51.100.1:0npu_selid=8
  dec_npuid=1 enc_npuid=1 dec_engid=-1 enc_engid=-1 dec_saidx=11 enc_saidx=8
------------------------------------------------------
name=Dialup ver=2 serial=1 198.51.100.1:0->0.0.0.0:0 nexthop=198.51.100.2 tun_id=10.0.0.1
tun_id6=::10.0.0.1 status=up dst_mtu=0 weight=1
bound_if=19 real_if=0 lgwy=static/1 tun=intf mode=dialup/2 encap=none options[0x228]=npu
frag-rfc  role=primary accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=1 refcnt=3 ilast=42952401 olast=42952401 ad=/0
stat: rxp=734 txp=114 rxb=113640 txb=9576
dpd: mode=on-demand on=0 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
```

Send security posture tags from FortiClient to FortiOS directly as JWT

This information is also available in the FortiOS 8.0 Administration Guide:

- Enforcing security posture verification before dial-up VPN connection

To enhance the resiliency of security posture tag verification during IPSec VPN connections, FortiClient endpoints now send security posture tags in JSON Web Token (JWT) format directly to the FortiGate. The JWT tags are received during the VPN establishment process after authentication. They are also received periodically on tag changes and tag token expiration. This provides a backup mechanism for FortiGate to verify tags when FortiClient EMS is unreachable or slow to respond. FortiGate continues to prioritize querying FortiClient EMS for tag verification as the primary method.

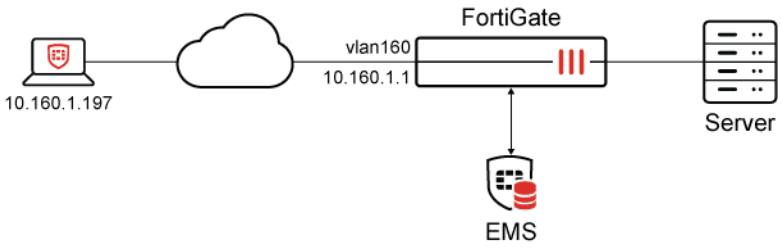
This feature requires FortiClient 7.4.5 and FortiClient EMS 7.4.4 or later.

See Improvements to tag compliance check for VPN connection for more background on FortiClient settings.

Example

In this example, FortiClient EMS is configured to use ZTNA token, which enables FortiClient to send security posture tags in JWT format.

A remote employee connects to the FortiGate VPN gateway while the FortiClient EMS connection is temporarily unavailable. FortiGate falls back to verifying the JWT security posture tag and allows the VPN connection to be established.



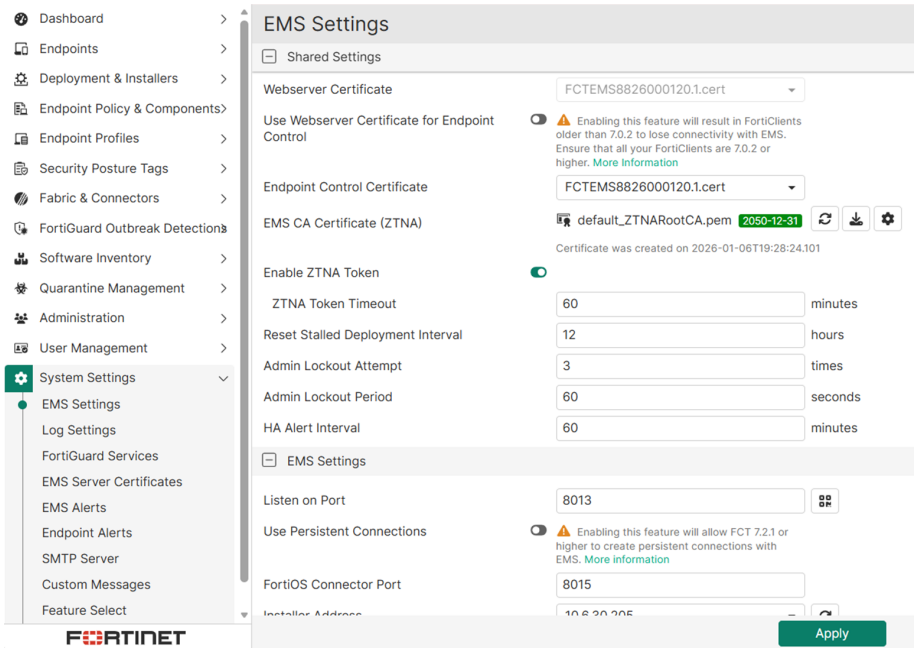
This example includes the following sections:

- Configuring FortiClient EMS on page 312
- Configuring FortiGate on page 313
- Verifying on page 314

Configuring FortiClient EMS

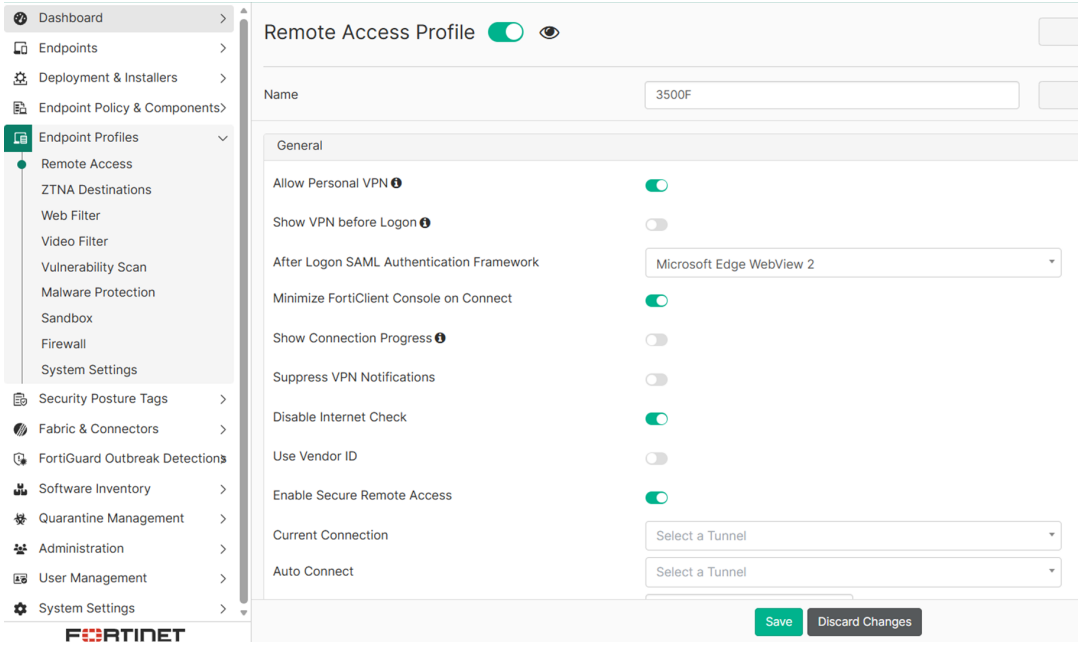
To configure FortiClient EMS:

1. On FortiClient EMS, go to *System Settings* > *EMS Settings*, and enable *Enable ZTNA Token*.
2. Set the *ZTNA Token Timeout* and interval time.



3. Go to *Endpoints Profiles* > *Remote Access*, and edit a Remote Access profile.

4. Under *General*, enable *Enable Secure Remote Access*.



5. Under *VPN Tunnels*, add or edit a tunnel.
6. In *Advanced Settings*, select the security posture tags.
7. Save the profile and push to FortiClient endpoints.

Configuring FortiGate

These steps assume FortiClient EMS fabric settings are already configured, and security posture tags have been synchronized to the FortiGate.

To configure the VPN tunnel on FortiGate:

1. On FortiGate, configure VPN phase1. Enable `remote-gw-match-ztna` and choose your ZTNA tags:

```
config vpn ipsec phase1-interface
    edit "FCT_EMS_TAG_ph2"
        set type dynamic
        set interface "vlan160"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dpd on-idle
        set dhgrp 20 21
        set eap enable
        set eap-identity send-request
        set authusrgrp "local_group"
        set ems-sn-check enable
```

```
        set remote-gw-match ztna
        set ipv4-start-ip 10.139.1.2
        set ipv4-end-ip 10.139.1.20
        set dns-mode auto
        set ipv4-split-include "all"
        set save-password enable
        set psksecret <password>
        set dpd-retryinterval 60
        set remote-gw-ztna-tags "EMS3_ZTNA_a"
    next
end
```

2. Configure VPN phase2:

```
config vpn ipsec phase2-interface
    edit "FCT_EMS_TAG_ph2"
        set phase1name "FCT_EMS_TAG_ph2"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

3. Configure a firewall policy:

```
config firewall policy
    edit 14
        set name "FCT_EMS_TAG_ph2"
        set srcintf "FCT_EMS_TAG_ph2"
        set dstintf "mgmt1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

Verifying

To verify:

1. Initiate a VPN connection from the remote FortiClient endpoint while the EMS connection failed.
2. During the VPN negotiation, use these commands to view the real-time debugs:

```
# diagnose debug application ike -1
# diagnose debug enable
ike V=root:0: udp comes 10.160.1.197:55846->10.160.1.1:4500,ifindex=1123,vrf=0,len=466....
…
ike V=root:0:FCT_EMS_TAG_ph2:FCT_EMS_TAG_ph2: created connection: 0x55e62c7d8a50 1123
10.160.1.1->10.160.1.197:55846.
```

```
,,,
ike V=root:0:FCT_EMS_TAG_ph2:40481: responder received AUTH msg
ike V=root:0:FCT_EMS_TAG_ph2:40481: processing notify type INITIAL_CONTACT
ike V=root:0:FCT_EMS_TAG_ph2:40481: peer identifier IPV4_ADDR 10.160.1.197
ike V=root:0:FCT_EMS_TAG_ph2:40481: eap-peer=yes
ike V=root:0:FCT_EMS_TAG_ph2:40481: re-validate gw ID
ike V=root:0:FCT_EMS_TAG_ph2:40481: gw validation OK
ike V=root:0:FCT_EMS_TAG_ph2:40481: responder preparing EAP identity request
…
ike V=root:0:FCT_EMS_TAG_ph2:40481: authentication succeeded
ike V=root:0:FCT_EMS_TAG_ph2:40481: processing notify type FORTICLIENT_CONNECT
ike V=root:0:FCT_EMS_TAG_ph2:40481: received FCT data len = 309, data = 'VER=1
FCTVER=7.4.5.1949
UID=F8A6AFF3531843B4972CBDBEEEB30E3C
IP=10.160.1.197
MAC=00-0c-29-2c-4d-d7;00-0c-29-2c-4d-e1;00-0c-29-2c-4d-eb;
HOST=DESKTOP-EMS-TAG-PH2
USER=tester
OSVER=Microsoft Windows 8.0 , 64-bit (build 9200)
REG_STATUS=0
EMSSN=FCTEMS8826000120
EMSID=00000000000000000000000000000000
FCTTAGS=1
```

The tag `FCTTAGS=1` indicates that FortiClient has JWT enabled and is sending security posture tags to FortiGate directly.

```
ike V=root:0:FCT_EMS_TAG_ph2:40481: received FCT-UID : F8A6AFF3531843B4972CBDBEEEB30E3C
…
ike V=root:0:FCT_EMS_TAG_ph2_0:40481: processing notify type ZTNA_TAG
```

This indicates FortiClient JWT has been received by the FortiGate:

```
...
ike V=root:0:FCT_EMS_TAG_ph2_0:40481: received ZTNA tags update token (len=791)
data=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbmRwb2ludCI6eyJ1aWQiOiJGOEE2QUZGMzUzMTg0M0I0OTcy
Q0JEQkVFRUIzMEUzQyIsInRhZ3MiOlt7Im5hbWUiOiJhIiwidHlwZSI6MSwibGV2bCI6bnVsbH0seyJuYW1lIjoiYWxsX3
JlZ2lzdGVyZWRfY2xpZW50cyIsInR5cGUiOjEsImxldmwiOm51bGx9LHsibmFtZSI6IkxvdyIsInR5cGUiOjMsImxldmwi
Om51bGx9XSwibndpZnMiOm51bGwsInp0bmFfY2VydF9zbiI6IkIwOUQ5NzE3ODEyMUI1ODhFRDMwNTUxMzc1Nzc1MjJDN0
I2ODlCNzkiLCJvd25lciI6bnVsbH0sImV4cCI6MTc2OTc2MTM4NywiaWF0IjoxNzY5NzU3Nzg3fQ.GT4B2wuDJcVKn1LCb
6UW1by0pXxvW5vdQD2DzvCmRxODoy-gk9-XKaXZtvlRyFVAVG6BGDOZR0UIW--
Eduhps4zCr0PdQhy2cFnhsKu7tygdxSEEHpm_ESXi6t3EcIbxPb5y4xQflb1ny4TscB-OVN68pXT_
BwgfuszBcYON2ZzgEiRbcJmfwzAkmjThfHik59EZBOOC_lQv1pkQ3-
IQHJVSq0QhBIPx8seYhSIMtto9vJwIEciaZ8DPGbHtJ1WlOtm7SKbkoIDer4xSDVhKO03McJ0hlxFQpjc7AufONrN1jbgW
3ydJJ03u3tQHbWETufayz3-eN9GlufTpidUBgQ
ike V=root:0:FCT_EMS_TAG_ph2_0: EMS ZTNA token tags check success ZTNA_a
ike V=root:0:FCT_EMS_TAG_ph2_0:40481: send EMS_ZTNA_TAGS_CHECK_SUCCESS
```

This indicates the JWT security posture tags have been validated.

3. Verify the VPN is established:

```
# diagnose vpn ike gateway list name FCT_EMS_TAG_ph2_0
vd: root/0
name: FCT_EMS_TAG_ph2_0
version: 2
interface: vlan160 1123
addr: 10.160.1.1:4500 -> 10.160.1.197:55846
tun_id: 10.139.1.2/::10.0.0.183
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 39081s ago
eap-user: u1
2FA: no
peer-id: 10.160.1.197
peer-id-auth: no
FortiClient UID: F8A6AFF3531843B4972CBDBEEEB30E3C
assigned IPv4 address: 10.139.1.2/255.255.255.255
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 20/20/20 ms
IPsec SA: created 1/2  established 1/2  time 0/0/0 ms
  id/spi: 40481 db9b0112a09c6850/ac673ffc75f4cdca
  direction: responder
  status: established 39081-39081s ago = 20ms
...
```

```
# diagnose vpn tunnel list name FCT_EMS_TAG_ph2_0
list ipsec tunnel by names in vd 0
------------------------------------------------------
name=FCT_EMS_TAG_ph2_0 ver=2 serial=196 10.160.1.1:4500->10.160.1.197:55846 nexthop=0.0.0.0
tun_id=10.139.1.2 tun_id6=::10.0.0.183 status=up dst_mtu=1500 weight=1 country=ZZ
bound_if=1123 real_if=1123 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none options
[0x23a8]=npu rgwy-chg rport-chg frag-rfc  run_state=0 role=sync-primary accept_traffic=1
overlay_id=0
parent=FCT_EMS_TAG_ph2 index=0
proxyid_num=1 child_num=0 refcnt=7 ilast=15 olast=15 ad=/0
stat: rxp=11054 txp=11874 rxb=3338588 txb=4823896
dpd: mode=on-idle on=1 status=ok idle=60000ms retry=3 count=0 seqno=2
natt: mode=keepalive draft=0 interval=10 remote_port=55846
fec: egress=0 ingress=0
proxyid=FCT_EMS_TAG_ph2 proto=0 sa=1 ref=4 serial=1 add-route
```

TLS 1.3 based VPN over TCP

This information is also available in the FortiOS 8.0 Administration Guide:

- TLS 1.3 based VPN over TCP

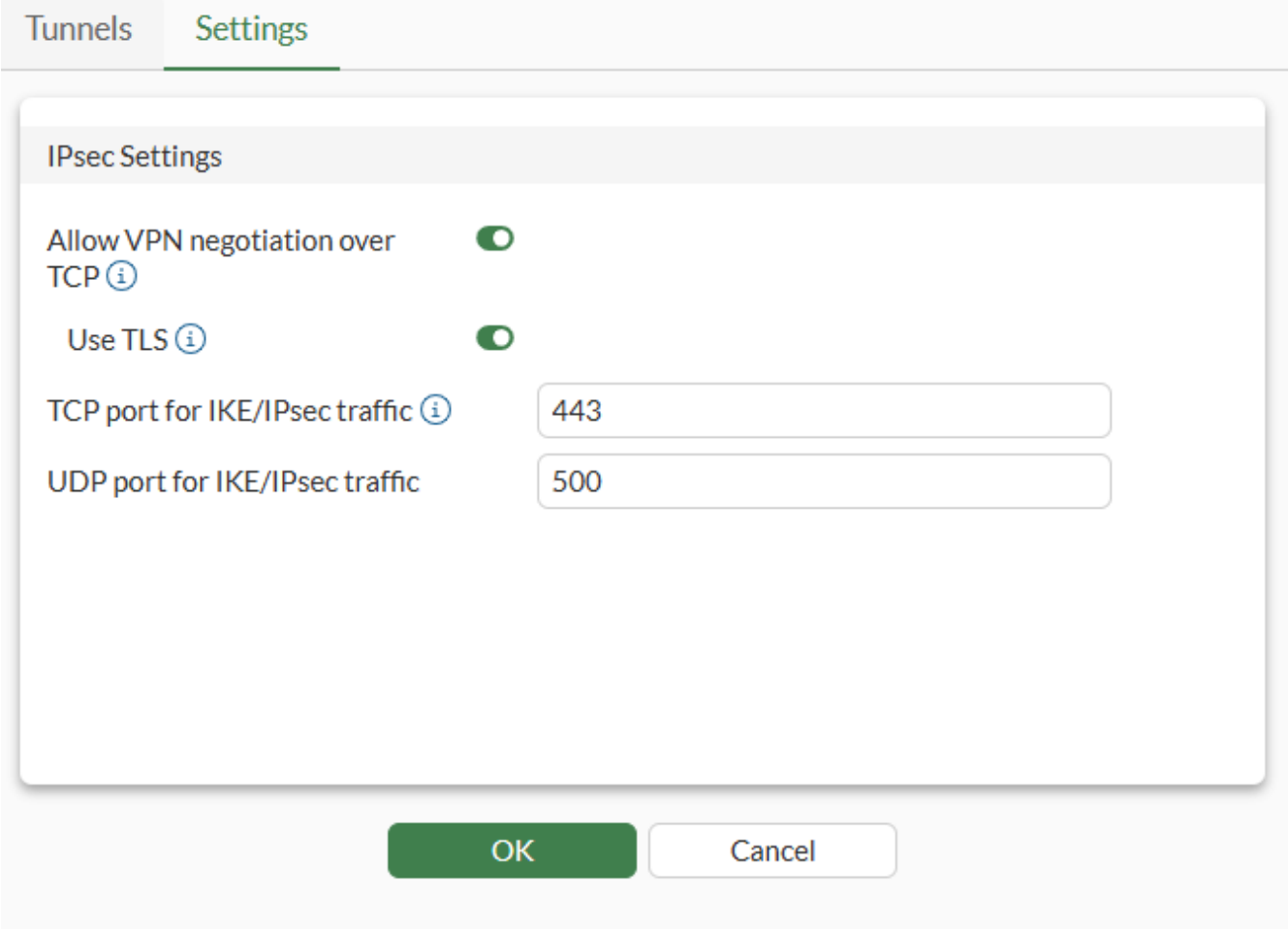
Before this enhancement, IKE based VPN tunnels could be configured to be ESP, UDP or TCP based. When TCP is used as the transport protocol, IKE and ESP packets are encapsulated in TCP in the port that you specified.

In this enhancement, TLS 1.3 is added as an option for VPN over TCP. The TLS option can be selected when *Allow VPN negotiation over TCP* is enabled in the *VPN > VPN Tunnels > Settings* page. With TLS enabled, IKE and ESP traffic are transported over TLS.

This option is supported for Site-to-Site and Dial-up VPN between FortiGates. FortiClient endpoints do not currently support TLS based VPN over TCP.

Usage

In the GUI, go to *VPN > VPN Tunnels* and select the *Settings* tab:



In the CLI:

```
config system settings
    set ike-tcp-service { enable | disable }
    set ike-tls-service { enable | disable }
end
```

Both settings are per-vdom settings to control whether IKE over TCP and TLS is used in VPN tunnels.

When `ike-tcp-service` is enabled, dialup VPNs configured in the VDOM will behave as the *Auto* transport option - transport over UDP will be preferred, but when the IKE connection cannot be made over UDP, communication will be attempted in TCP or TLS. For site-to-site VPNs, you must set the transport option to *Auto* to use TCP or TLS.

When disabled (default), dialup VPNs will not attempt to connect over TCP.

After upgrading:

1. If a dynamic-type dialup VPN with auto or TCP transport mode exists, the `ike-tcp-service` setting is enabled for the VDOM.
2. If a static VPN with TCP transport mode exists, the `ike-tcp-service` setting is enabled for the VDOM.
3. If a static VPN with auto transport mode exists, that VPN will be changed to use transport UDP.

Example

An IKEv2 over TLS 1.3 VPN tunnel is formed between a VPN Client (FGTA) and a dialup VPN Gateway (FGTD).

To configure the hub FortiGate, FGTD, in the GUI:

1. Go to *VPN* > *VPN Tunnels* and select the *Settings* tab.
2. Enable *Allow VPN negotiation over TCP*.
3. Enable *Use TLS*.
4. Click *OK*.
5. Go to *VPN* > *VPN Tunnels*, select the *Tunnels* tab, and click *Create new* > *Custom IPsec tunnel*.
6. Configure the following settings:

Field	Value
Name	tofgta
Remote gateway	Dialup user
Interface	port13
Authentication Method	Signature
Certificate name	Select your certificate
IKE	Version 2 (default)

7. In the *Phase 2 selectors* table, click *Create New* and configure the following:

Field	Value
Name	tofgta
Encryption – authentication	Remove all SHA1 based entries

8. Click *OK* to save the phase 2 selector.
9. Click *OK* to save the tunnel.

10. Create firewall policies to allow traffic from *tofgta* to a protected network, and to allow traffic from a protected network to *tofgta*.

To configure the hub FortiGate, FGTD, in the CLI:

```
config system settings
    set ike-tcp-service enable
    set ike-tls-service enable
end
config vpn ipsec phase1-interface
    edit "tofgta"
        set type dynamic
        set interface "port13"
        set ike-version 2
        set authmethod signature
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set certificate "test1"
    next
end
config vpn ipsec phase2-interface
    edit "tofgta"
        set phase1name "tofgta"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "tofgta"
        set action accept
        set srcaddr "192.168.5.0"
        set dstaddr "10.1.100.0"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "tofgta"
        set dstintf "port2"
        set action accept
        set srcaddr "10.1.100.0"
        set dstaddr "192.168.5.0"
        set schedule "always"
        set service "ALL"
    next
end
```

To configure the dialup client FortiGate, FGTA, on the GUI:

1. Go to *VPN* > *VPN Tunnels* and select the *Settings* tab.
2. Enable *Allow VPN negotiation over TCP*.
3. Enable *Use TLS*.
4. Click *OK*.
5. Go to *VPN* > *VPN Tunnels*, select the *Tunnels* tab, and click *Create new* > *Custom IPsec tunnel*.
6. Configure the following settings:

Field	Value
Name	tofgtd
Remote gateway	Static IP address
IP address	173.1.1.1
Interface	port9
Transport	Auto
Authentication Method	Signature
Certificate name	Select your certificate
IKE	Version 2 (default)

7. In the *Phase 2 selectors* table, click *Create New* and configure the following:

Field	Value
Name	tofgtd
Encryption – authentication	Remove all SHA1 based entries

8. Click *OK* to save the phase 2 selector.
9. Click *OK* to save the tunnel.
10. Create firewall policies to allow traffic from *tofgtd* to a protected network, and to allow traffic from a protected network to *tofgtd*.

To configure the dialup client FortiGate, FGTA, on the CLI:

```
config system settings
    set ike-tcp-service enable
    set ike-tls-service enable
end
config vpn ipsec phase1-interface
    edit "tofgtd"
        set interface "port9"
        set ike-version 2
        set authmethod signature
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
```

```
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set transport auto
        set remote-gw 173.1.1.1
        set certificate "test3"
    next
end
config vpn ipsec phase2-interface
    edit "tofgtd"
        set phase1name "tofgtd"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "tofgtd"
        set action accept
        set srcaddr "10.1.100.0"
        set dstaddr "192.168.5.0"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "tofgtd"
        set dstintf "port2"
        set action accept
        set srcaddr "192.168.5.0"
        set dstaddr "10.1.100.0"
        set schedule "always"
        set service "ALL"
    next
end
```

Verification

When the tunnel is established, run the following debugs on FGTA to verify:

```
# diagnose vpn ike gateway list

vd: root/0
name: tofgtd
version: 2
interface: port9 15
addr: 11.101.1.1:443 -> 173.1.1.1:7179
tun_id: 173.1.1.1/::173.1.1.1
remote_location: 0.0.0.0
network-id: 0
transport: TLS
tcp-tx-1m-rate : 0 kbps
```

```
tcp-tx-5m-rate : 0 kbps
tcp-tx-drop    : 0
tcp-rx-1m-rate : 0 kbps
tcp-rx-5m-rate : 0 kbps
tcp-rx-drop    : 0
created: 4968s ago
peer-id: C = CA, ST = BC, L = Burnaby, O = Fortinet, OU = QA, CN = test3
peer-id-auth: yes
pending-queue: 0
PPK: no
IKE SA: created 1/2  established 0/1  time 0/0/0 ms
IPsec SA: created 1/2  established 1/1  time 0/0/0 ms

  id/spi: 3 823c08a1ddd29111/9e2f8c8339d5ba43
  direction: responder
  status: established 4948-4948s ago = 0ms
  proposal: aes128-sha256
  child: no
  SK_ei: bf126e0a84efa7ff-8d6e8c56694a1850
  SK_er: 78c57cbadd9d949c-ff88391ff06083ee
  SK_ai: 451eec850591848d-10d9deae1e58df5a-741e1cadf1922d98-a81086392bcfd0e2
  SK_ar: c83ecb5fb337286a-908b54e1454e859f-8e5ad8c4fd5184c3-48ea79a494ad6bef
  PPK: no
  message-id sent/recv: 0/2
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/81181
  DPD sent/recv: 00000000/00000000
  peer-id: C = CA, ST = BC, L = Burnaby, O = Fortinet, OU = QA, CN = test3
```

To verify the TLS connection in IPsec VPN tunnel real-time debugs:

```
# diagnose debug application ike 127

ike V=root:0:tofgtd: going to be deleted
ike V=root:0:tofgtd: reset TCP ports and transport mode
...
ike V=root:tcp deletes tcp-transport(vd=0, vrf=0, intf=15:15, 11.101.1.1:443->173.1.1.1:7179
sock=43 refcnt=2 ph1=0x55c2b67379e0) (1).
ike V=root:tcp destroys tcp-transport(vd=0, vrf=0, intf=15:15, 11.101.1.1:443->173.1.1.1:7179
sock=43 refcnt=0 ph1=(nil)) (0).
ike V=root:0:tofgtd: set oper down
ike V=root:0:tofgtd: schedule auto-negotiate
ike V=root:0:tofgtd: auto-negotiate connection
ike V=root:0:tofgtd:tofgtd: created connection: 0x55c2b673f5d0 15 11.101.1.1->173.1.1.1:443.
ike V=root:0:tofgtd:tofgtd: chosen to populate IKE_SA traffic-selectors
ike V=root:0:tofgtd: no suitable IKE_SA, queuing CHILD_SA request and initiating IKE_SA
negotiation
ike V=root:tls connection is inprogress tcp-transport(vd=0, vrf=0, intf=15:15, 11.101.1.1:7007-
>173.1.1.1:443 sock=41 refcnt=2 ph1=0x55c2b6730a10) (1).
ike V=root:0:tofgtd:4: generate DH public value request pending
```

```
ike V=root:tcp accepts ike tcp-transport(vd=0, vrf=0, intf=0:15, 11.101.1.1:443->173.1.1.1:4205
sock=43 refcnt=2 ph1=(nil)) (2).
ike V=root:tls connecting, ssl_want(2)
ike V=root:tls accepting.
ike V=root:tls connected, IKE tcp prefix sent
ike V=root:0:tofgtd:4: create NAT-D hash local 11.101.1.1/7007 remote 173.1.1.1/443
ike V=root:tls accepted tcp-transport(vd=0, vrf=0, intf=0:15, 11.101.1.1:443->173.1.1.1:4205
sock=43 refcnt=2 ph1=(nil)).
ike V=root:tls verified connection tcp-transport(vd=0, vrf=0, intf=0:15, 11.101.1.1:443-
>173.1.1.1:4205 sock=43 refcnt=2 ph1=(nil)).
ike V=root:0: tcp comes 173.1.1.1:443->11.101.1.1:7007,ifindex=15,vrf=0,len=264....
...
```

Native VPN remote access support in VPN wizard

This information is also available in the FortiOS 8.0 Administration Guide:

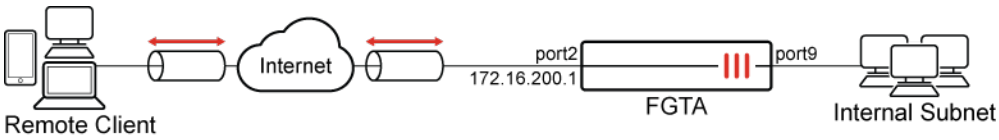
- Native VPN remote access support in VPN wizard

Native VPN remote access configurations have been improved on the VPN wizard. For supported OS's, configurations from the VPN wizard will work out of the box. Native VPN client defaults to using L2TP over IPsec for Windows, Android, and macOS/iOS clients. Administrators can also configure IKEv2 for Windows and Android clients. DH groups and proposals have been updated to work with native clients.

In L2TP over IPsec mode, only pre-shared key is supported; signature cannot be chosen as Authentication method. Addressing mode must be set to Manual. These limitations do not apply to IKEv2 mode.

Organizations that manage a mixed environment (Windows, macOS, iOS, Android) can use the wizard to deploy a VPN tunnel that automatically aligns with each OS's native VPN client defaults.

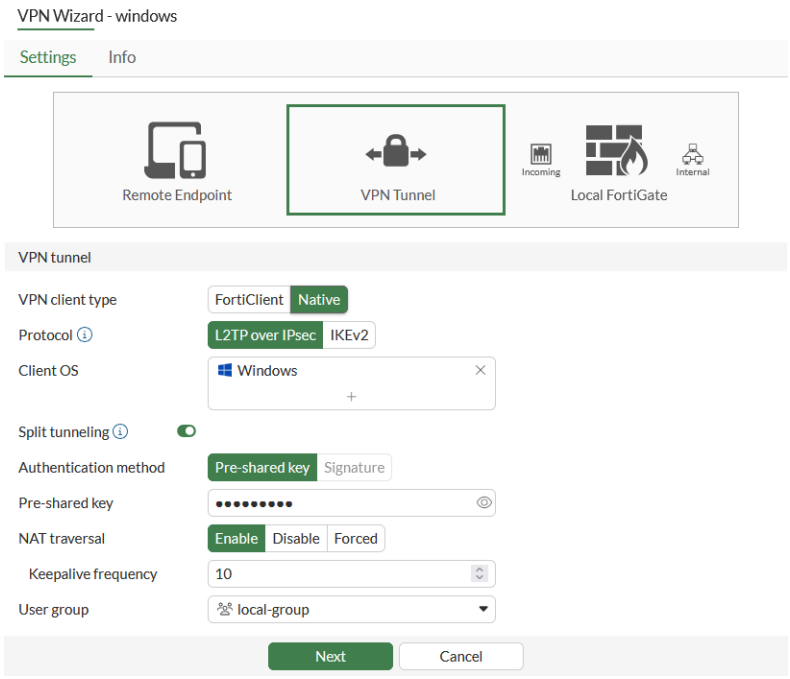
Example 1: Configure IPsec tunnel for native VPN clients



In this example, the FortiGate acts as a VPN gateway to provide secure internet access (SIA) through the shared WAN on port2, and for remote access to the Internal network through port9. The settings are compatible for Windows, Android, and macOS/iOS clients connecting through L2TP over IPsec. However, connecting from a Windows client is demonstrated here. User authentication is based on pre-defined usergroup *local-group* that contains a user called *userc*.

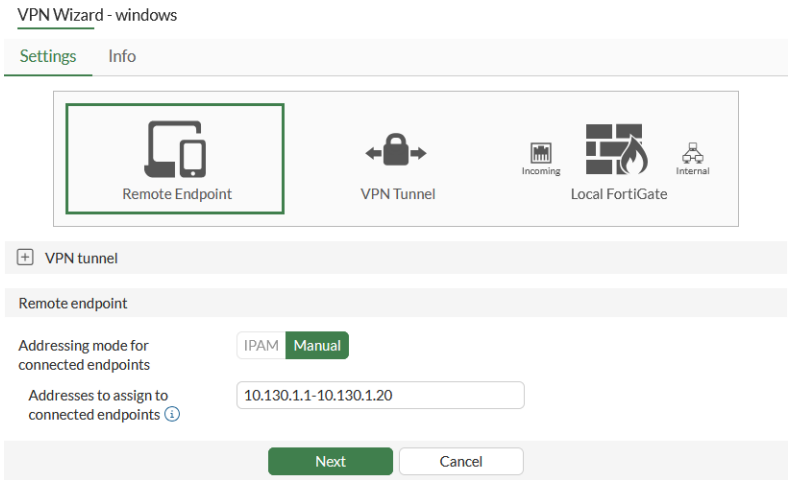
To configure L2TP IPsec tunnel from a template for Windows, Android, and macOS:

1. Go to *VPN* > *VPN Wizard*.
2. Name the VPN tunnel, select *Remote Access* template, and click *Begin*.
3. Configure the VPN tunnel settings, then click *Next*:

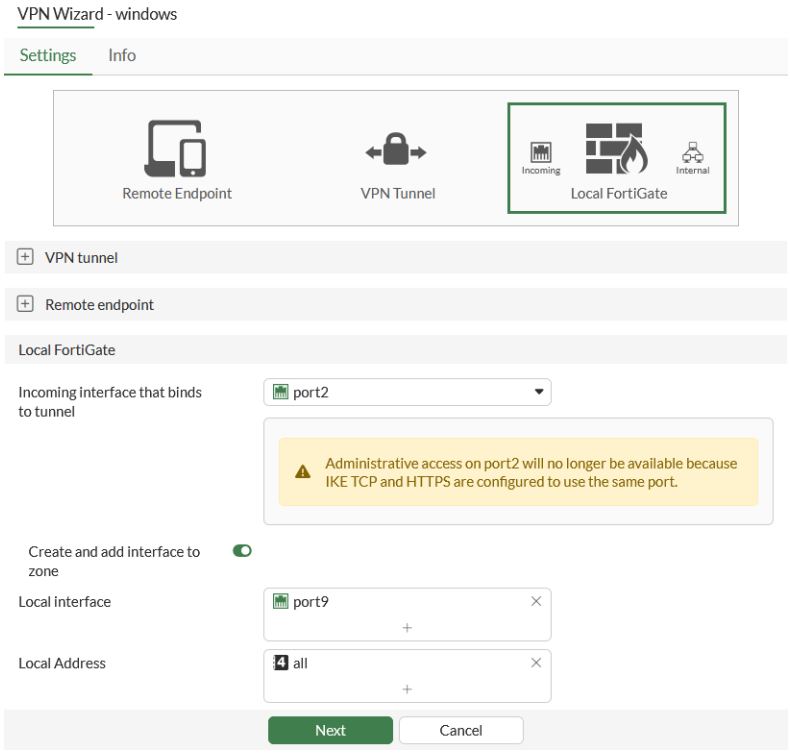


Parameter	Value
VPN client type	Native
Protocol	L2TP over IPsec
Client OS	Windows
Shared WAN	port2
Authentication method	Pre-Shared key
Pre-Shared key	Enter the pre-shared key
User group	local-group

4. Set *Addresses to assign to connected endpoints* to *10.130.1.1-10.130.1.20*, then click *Next*.

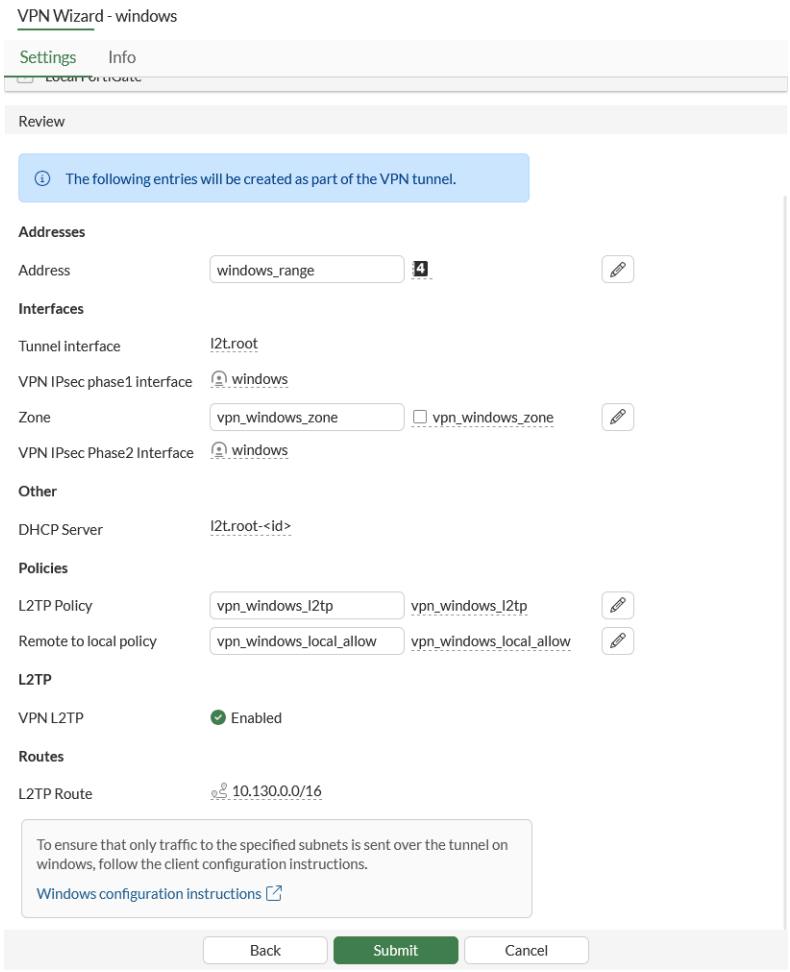


5. Configure the local FortiGate options, then click *Next*:



Parameter	Value
Incoming interface that binds to tunnel	port2
Local interface	port9
Local Address	all

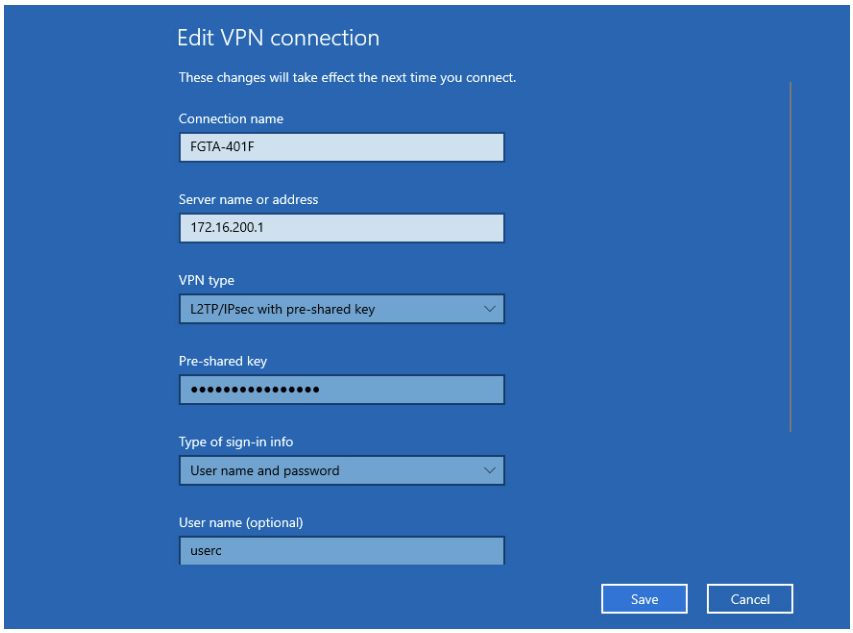
6. Review the configuration, then click *Submit*.



Only Android 12 and below are supported for L2TP VPN.

To configure L2TP connection for Windows:

1. Open *VPN* or go to *Settings* > *Network & internet*.
2. Add and configure a VPN:



Parameter	Value
VPN provider	Windows (built-in)
Connection name	Enter a connection name.
Server name or address	Enter your FortiGate Firewall's hostname or IP address.
VPN type	L2TP/IPsec with pre-shared key
Pre-shared key	Enter the pre-shared key.
Type of sign-in info	Username and password
Username (optional)	Enter the username.
Password (optional)	Enter the password.

3. Click *Save*.

To configure L2TP IPsec tunnel in the CLI:

```
config user group
    edit "local-group"
        set member "usera" "userc"
    next
end
```

```
config vpn ipsec phase1-interface
    edit "windows"
        set type dynamic
        set interface "port1"
        set ike-version 1
        set peertype any
```

```
        set net-device disable
        set proposal aes256-sha512 aes256-sha256 aes256-sha1
        set dhgrp 19 14 2
        set wizard-type dialup-l2tp
        set psksecret <password>
    next
end
```

```
config vpn ipsec phase2-interface
    edit "windows"
        set phase1name "windows"
        set proposal aes256-sha512 aes256-sha256 aes256-sha1
        set pfs disable
        set keepalive enable
        set encapsulation transport-mode
        set l2tp enable
    next
end
```

```
config vpn l2tp
    set status enable
    set eip 10.130.1.20
    set sip 10.130.1.1
    set enforce-ipsec enable
    set usrgrp "local-group"
end
```

```
config firewall address
    edit "windows_range"
        set type iprange
        set start-ip 10.130.1.1
        set end-ip 10.130.1.20
    next
end
```

Three policies are created:

- One to allow L2TP traffic into the WAN interface
- One to allow VPN traffic to the local subnet
- One to allow VPN traffic to the Internet

```
config firewall policy
    edit 3
        set name "vpn_windows_l2tp"
        set srcintf "windows"
        set dstintf "port2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "L2TP"
    next
```

```
    edit 4
        set name "vpn_windows_local_allow"
        set srcintf "l2t.root"
        set dstintf "port9"
        set action accept
        set srcaddr "windows_range"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
    edit 5
        set name "vpn_windows_utm_allow"
        set srcintf "l2t.root"
        set dstintf "port2"
        set action accept
        set srcaddr "windows_range"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

```
config router static
    edit 2
        set dst 10.130.0.0 255.255.0.0
        set device "l2t.root"
    next
end
```

To check the IPsec phase1/phase2 interface status:

```
# diagnose vpn ike gateway list

vd: root/0
name: windows_0
version: 1
interface: port1 7
addr: 172.16.200.1:4500 -> 172.16.200.254:64916
tun_id: 172.16.200.254/::10.0.0.15
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 19s ago
peer-id: 31.1.1.198
peer-id-auth: no
nat: me peer
pending-queue: 0
IKE SA: created 1/1  established 1/1  time 20/20/20 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms
```

```
  id/spi: 16 ccfbf5c901e3899f/547217d62b9bc1ef
  direction: responder
  status: established 19-19s ago = 20ms
  proposal: aes256-sha1
  key: cd7c7a2385ef9085-30a95bab3c1a6178-4dea8ede135046e0-fbb2c44074b6fd74
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 28800/28510
  DPD sent/recv: 00000000/00000000
  peer-id: 31.1.1.198
```

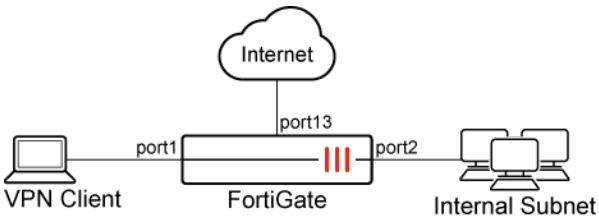
```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=windows_0 ver=1 serial=10 172.16.200.1:4500->172.16.200.254:64916 nexthop=0.0.0.0 tun_
id=172.16.200.254 tun_id6=::10.0.0.15 status=up dst_mtu=1500 weight=1 country=ZZ
bound_if=7 real_if=7 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none options[0x23a8]=npu rgwy-
chg rport-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=windows index=0
proxyid_num=1 child_num=0 refcnt=6 ilast=0 olast=42955707 ad=/0
stat: rxp=6 txp=0 rxb=696 txb=0
dpd: mode=on-demand on=1 status=ok idle=60000ms retry=3 count=0 seqno=0
natt: mode=keepalive draft=32 interval=10 remote_port=64916
fec: egress=0 ingress=0
proxyid=windows proto=17 sa=1 ref=2 serial=1 transport-mode add-route
  src: 17:172.16.200.1-172.16.200.1:1701
  dst: 17:172.16.200.254-172.16.200.254:0
  SA:  ref=3 options=1a7 type=00 soft=0 mtu=1454 expire=3563/0B replaywin=2048
       seqno=401 esn=0 replaywin_lastseq=00000006 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=3589/3600
  dec: spi=a91127c2 esp=aes key=32
e54c49f9205f06ff55e9e974eefa00079bd57466041cf0a0512a11193a533a36
       ah=sha1 key=20 3eca7a57dda0ae2dbdf589f2798a2539b3ca774b
  enc: spi=c8be2f10 esp=aes key=32
008379628c4d67a232fb43dd2b7606bb43c8e02a731df7b90833485e2bb9f522
       ah=sha1 key=20 0291e163d22963ebe01bc8cc075cfabef935c10d
  dec:pkts/bytes=6/696, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=0.0.0.0:0 npu_lgwy=0.0.0.0:0npu_selid=e
  dec_npuid=0 enc_npuid=0 dec_engid=-1 enc_engid=-1 dec_saidx=-1 enc_saidx=-1
------------------------------------------------------
name=windows ver=1 serial=2 172.16.200.1:0->0.0.0.0:0 nexthop=0.0.0.0 tun_id=10.0.0.1 tun_
id6=::10.0.0.1 status=up dst_mtu=0 weight=1
bound_if=7 real_if=0 lgwy=static/1 tun=intf mode=dialup/2 encap=none options[0x228]=npu frag-rfc
role=primary accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=1 refcnt=3 ilast=42955707 olast=42955707 ad=/0
stat: rxp=30 txp=0 rxb=3480 txb=0
dpd: mode=on-demand on=0 status=ok idle=20000ms retry=3 count=0 seqno=0
```

```
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
```

```
# diagnose debug enable
# diagnose debug app ike -1
# diagnose debug app fnbam -1
ike V=root:0:ccfbf5c901e3899f/0000000000000000:16: responder: main mode get 1st message...
...
ike V=root:0:windows: adding new dynamic tunnel for 172.16.200.254:64916
ike V=root:0:windows_0: tunnel created tun_id 172.16.200.254/::10.0.0.15 remote_location 0.0.0.0
ike V=root:0:windows_0: added new dynamic tunnel for 172.16.200.254:64916
ike V=root:0:windows_0:16: established IKE SA ccfbf5c901e3899f/547217d62b9bc1ef
ike V=root:0:windows_0: DPD disabled, not negotiated
ike V=root:0:windows_0:16: check peer route: if_addr4_rcvd=0, if_addr6_rcvd=0, mode_cfg=0
ike V=root:0:windows_0:16: no pending Quick-Mode negotiations
ike V=root:0: udp comes 172.16.200.254:64916->172.16.200.1:4500,ifindex=7,vrf=0,len=448....
ike V=root:0: IKEv1 exchange=Quick id=ccfbf5c901e3899f/547217d62b9bc1ef:00000001 len=444 vrf=0
...
ike V=root:0:windows_0:16:61: received NATOA-i 31.1.1.198
ike V=root:0:windows_0:16:61: received NATOA-r 172.18.68.206
ike V=root:0:windows_0:16:61: peer proposal is: peer:17:31.1.1.198-31.1.1.198:1701,
me:17:172.18.68.206-172.18.68.206:1701
ike V=root:0:windows_0:16:windows:61: trying
ike V=root:0:windows_0:16:61: L2TP selectors 17:172.18.68.206-172.18.68.206:1701 -> 17:31.1.1.198-
31.1.1.198:1701
ike V=root:0:windows_0:16:61: transport mode, override with 17:172.16.200.1-172.16.200.1:1701 ->
17:172.16.200.254-172.16.200.254:0
ike V=root:0:windows_0:16:windows:61: matched phase2
ike V=root:0:windows_0:16:windows:61: dynamic client
ike V=root:0:windows_0:16:windows:61: my proposal:
ike V=root:0:windows_0:16:windows:61: proposal id = 1:
ike V=root:0:windows_0:16:windows:61:   protocol id = IPSEC_ESP:
ike V=root:0:windows_0:16:windows:61:      trans_id = ESP_AES_CBC (key_len = 256)
ike V=root:0:windows_0:16:windows:61:      encapsulation = ENCAPSULATION_MODE_TRANSPORT
ike V=root:0:windows_0:16:windows:61:         type = AUTH_ALG, val=SHA2_512
ike V=root:0:windows_0:16:windows:61:      trans_id = ESP_AES_CBC (key_len = 256)
ike V=root:0:windows_0:16:windows:61:      encapsulation = ENCAPSULATION_MODE_TRANSPORT
ike V=root:0:windows_0:16:windows:61:         type = AUTH_ALG, val=SHA2_256
ike V=root:0:windows_0:16:windows:61:      trans_id = ESP_AES_CBC (key_len = 256)
ike V=root:0:windows_0:16:windows:61:      encapsulation = ENCAPSULATION_MODE_TRANSPORT
ike V=root:0:windows_0:16:windows:61:         type = AUTH_ALG, val=SHA1
ike V=root:0:windows_0:16:windows:61: incoming proposal:
ike V=root:0:windows_0:16:windows:61: proposal id = 1:
ike V=root:0:windows_0:16:windows:61:   protocol id = IPSEC_ESP:
ike V=root:0:windows_0:16:windows:61:      trans_id = ESP_AES_CBC (key_len = 256)
ike V=root:0:windows_0:16:windows:61:      encapsulation = UDP_ENCAPSULATION_MODE_TRANSPORT_
RFC3947
ike V=root:0:windows_0:16:windows:61:         type = AUTH_ALG, val=SHA1
ike V=root:0:windows_0:16:windows:61: negotiation result
ike V=root:0:windows_0:16:windows:61: proposal id = 1:
ike V=root:0:windows_0:16:windows:61:   protocol id = IPSEC_ESP:
ike V=root:0:windows_0:16:windows:61:      trans_id = ESP_AES_CBC (key_len = 256)
```

```
ike V=root:0:windows_0:16:windows:61:      encapsulation = ENCAPSULATION_MODE_TRANSPORT
ike V=root:0:windows_0:16:windows:61:         type = AUTH_ALG, val=SHA1
ike V=root:0:windows_0:16:windows:61: using udp transport mode.
ike V=root:0:windows_0:16:61: sending NATOA-i 172.16.200.254
ike V=root:0:windows_0:16:61: sending NATOA-r 172.16.200.1
ike V=root:0:windows_0:16:windows:61: replay protection enabled
ike V=root:0:windows_0:16:windows:61: SA life soft seconds=3589.
ike V=root:0:windows_0:16:windows:61: SA life hard seconds=3600.
ike V=root:0:windows_0:16:windows:61: IPsec SA selectors #src=1 #dst=1
ike V=root:0:windows_0:16:windows:61: src 0 7 17:172.16.200.1-172.16.200.1:1701
ike V=root:0:windows_0:16:windows:61: dst 0 7 17:172.16.200.254-172.16.200.254:0
ike V=root:0:windows_0:16:windows:61: add dynamic IPsec SA selectors 183
ike V=root:0:windows_0:16:windows:61: added dynamic IPsec SA proxyids new 1 183
ike V=root:0:windows:61: add route 172.16.200.254/255.255.255.255 gw 172.16.200.254 oif windows
(51) metric 15 priority 1
ike V=root:0:windows_0:16:windows:61: tunnel 1 of VDOM limit 0/0
ike V=root:0:windows_0:16:windows:61: add IPsec SA: SPIs=a91127c2/c8be2f10
ike 0:windows_0:16:windows:61: IPsec SA dec spi a91127c2 key
32:E54C49F9205F06FF55E9E974EEFA00079BD57466041CF0A0512A11193A533A36 auth
20:3ECA7A57DDA0AE2DBDF589F2798A2539B3CA774B
ike 0:windows_0:16:windows:61: IPsec SA enc spi c8be2f10 key
32:008379628C4D67A232FB43DD2B7606BB43C8E02A731DF7B90833485E2BB9F522 auth
20:0291E163D22963EBE01BC8CC075CFABEF935C10D
ike V=root:0:windows_0:16:windows:61: transport mode encapsulation is enabled
ike V=root:0:windows_0:16:windows:61: added IPsec SA: SPIs=a91127c2/c8be2f10
ike V=root:0:windows_0:16:windows:61: sending SNMP tunnel UP trap
ike V=root:0:windows_0: tunnel up event
...
```

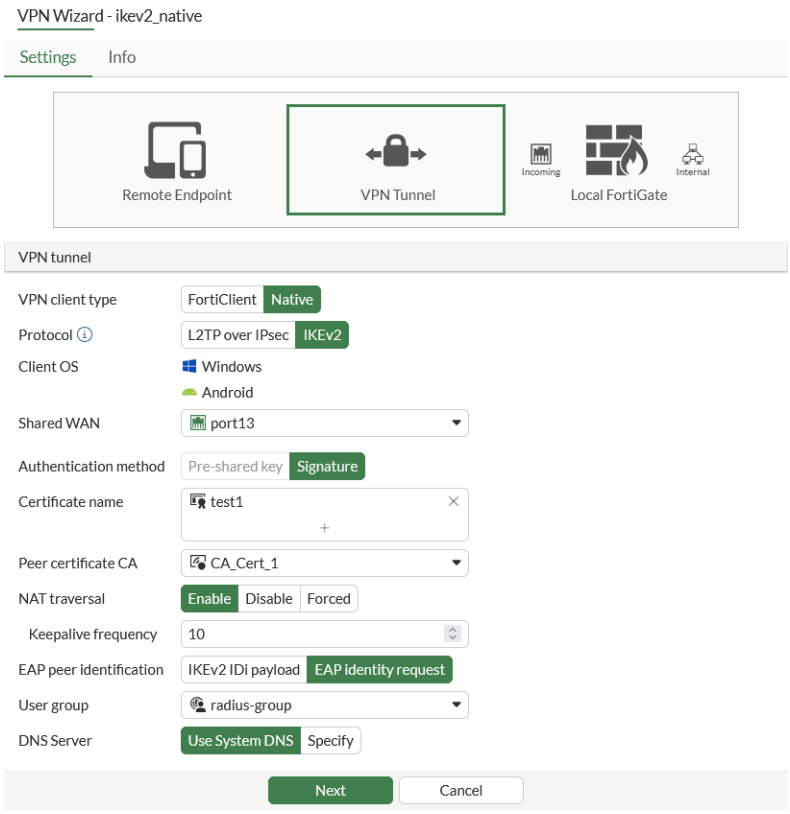
Example 2: Configure IKE v2 IPsec tunnel from template for Windows



In this example, the FortiGate acts as a VPN gateway to provide secure internet access (SIA) through the shared WAN on port13, and for remote access to the Internal network through port2. The settings are compatible for Windows and Android clients only connecting through IKEv2. However, connecting from a Windows client is demonstrated here. Signature authentication is used, and user authentication is based on RADIUS authentication.

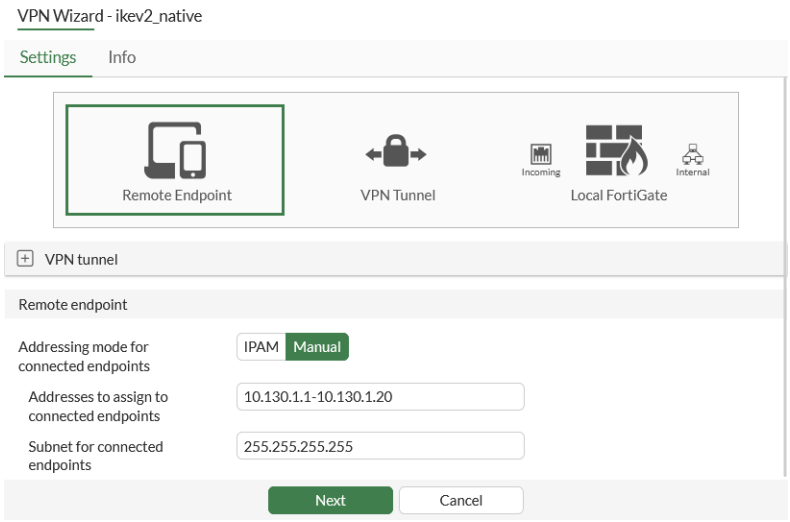
To configure IKE v2 IPsec tunnel from a template for Windows:

1. Go to *VPN > VPN Wizard*.
2. Name the VPN tunnel, select *Remote Access* template, and click *Begin*.
3. Configure the VPN tunnel settings, then click *Next*:

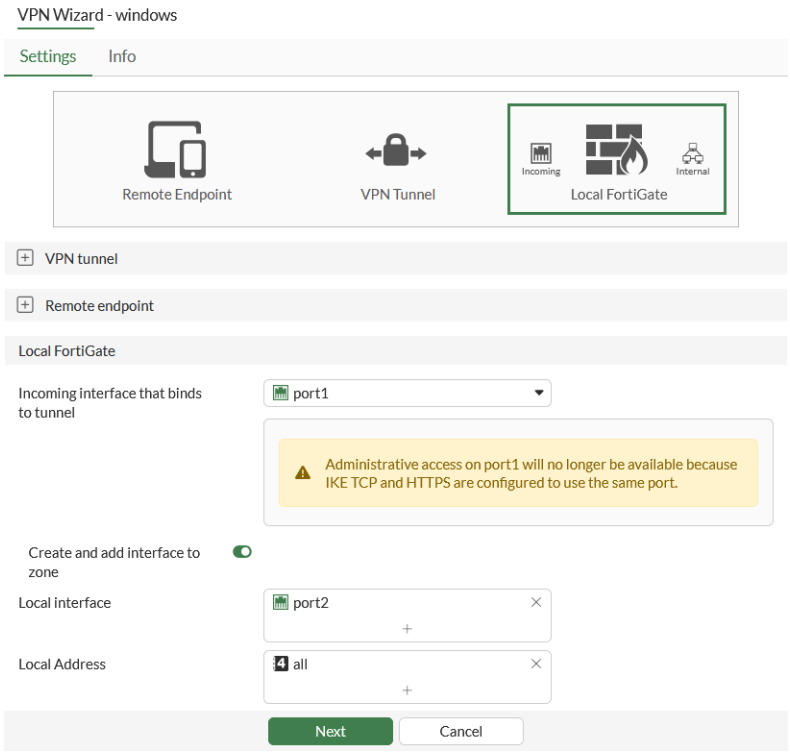


Parameter	Value
VPN client type	Native
Protocol	IKEv2
Shared WAN	port13
Authentication method	Signature
Certificate names	test1
Peer Certificate CA	CA_Cert_1
User group	radius-group

4. Set *Addressing mode for connected endpoints* to *Manual* and set *Addresses to assign to connected endpoints* to *10.130.1.1-10.130.1.20*, then click *Next*.

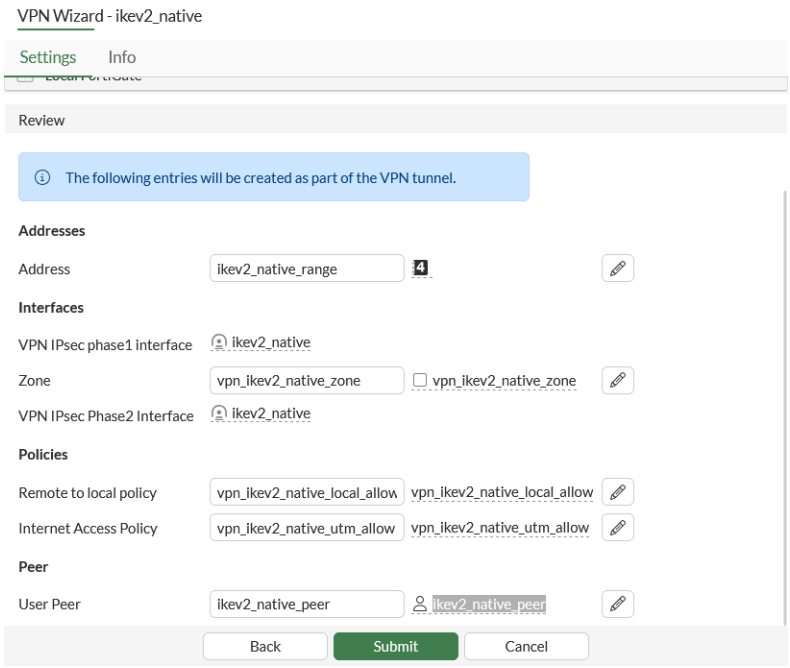


5. Configure the local FortiGate options, then click *Next*:



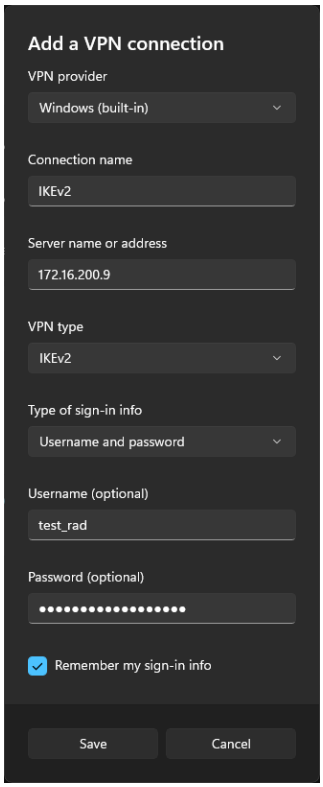
Parameter	Value
Incoming interface that binds to tunnel	port1
Local interface	port2
Local Address	all

6. Review the configuration, then click *Submit*.



To configure IKEv2 VPN connection for Windows:

1. Open *VPN* or go to *Settings* > *Network & internet*.
2. Add and configure a VPN:



Parameter	Value
VPN provider	Windows (built-in)
Connection name	Enter a connection name.
Server name or address	Enter your FortiGate Firewall's hostname or IP address.
VPN type	IKEv2
Type of sign-in info	Username and password
Username (optional)	Enter the username.
Password (optional)	Enter the password.

3. Click *Save*.

To configure IKE v2 IPsec tunnel for Windows in the CLI:

```
config vpn ipsec phase1-interface
    edit "ikev2_native"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set authmethod signature
        set net-device disable
        set mode-cfg enable
        set proposal aes256-sha256 aes256gcm-prfsha256 aes256gcm-prfsha384
        set dhgrp 20 14 2
        set eap enable
        set eap-identity send-request
        set wizard-type dialup-ikev2
        set authusrgrp "radius-group"
        set certificate "test1"
        set peer "ikev2_native_peer"
        set assign-ip-from name
        set dns-mode auto
        set ipv4-name "ikev2_native_range"
    next
end
```

```
config vpn ipsec phase2-interface
    edit "ikev2_native"
        set phase1name "ikev2_native"
        set proposal aes256gcm aes256-sha1 aes128-sha1
        set pfs disable
        set keepalive enable
    next
end
```

```
config firewall policy
    edit 1
        set name "vpn_ikev2_native_local_allow"
```

```
        set uuid 402dbbe2-0de4-51f1-202f-ea123f6da629
        set srcintf "ikev2_native"
        set dstintf "port2"
        set action accept
        set srcaddr "ikev2_native_range"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
    edit 2
        set name "vpn_ikev2_native_utm_allow"
        set uuid 402fbab4-0de4-51f1-97c2-2e4456a64492
        set srcintf "ikev2_native"
        set dstintf "port13"
        set action accept
        set srcaddr "ikev2_native_range"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

```
config firewall address
    edit "ikev2_native_range"
        set uuid 402b4d9e-0de4-51f1-c355-1375e255836b
        set type iprange
        set start-ip 10.130.1.1
        set end-ip 10.130.1.20
    next
end
```

```
config router static
    edit 1
        set gateway 172.16.200.254
        set device "port1"
    next
end
```

To check the IPsec phase1/phase2 interface status:

```
# diagnose vpn ike gateway list

vd: root/0
name: ikev2_native_0
version: 2
interface: port1 7
addr: 172.16.200.4:4500 -> 172.16.200.254:64917
tun_id: 10.130.1.1/::10.0.0.5
remote_location: 0.0.0.0
```

```
network-id: 0
transport: UDP
created: 9s ago
eap-user: test_rad
2FA: no
assigned IPv4 address: 10.130.1.1/0.0.0.0
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 230/230/230 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 89 bc5aa3bb99c0963c/290cf660b0977bf5
  direction: responder
  status: established 9-8s ago = 230ms
  proposal: aes256-sha256
  child: no
  SK_ei: 1f769b39b6e45680-bc358b9bbd82e462-5f0843ec67ac6a5a-56ef87e3cd629067
  SK_er: 094a6134c1e9e472-8730eb86ceb92a04-292ea04fa3514643-1ad51077fc429420
  SK_ai: 1cb620c46998be6e-aaa2136a04ccae9a-d478963782112a87-c35806662125531b
  SK_ar: 9d3c9205a68e9efb-3d95fce5b3bbb81a-b2642e584c509f6c-08be1b3cd27a2a02
  PPK: no
  message-id sent/recv: 0/7
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/86121
  DPD sent/recv: 00000000/00000000
```

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=ikev2_native ver=2 serial=4 172.16.200.4:0->0.0.0.0:0 nexthop= tun_id=10.0.0.2 tun_
id6=::10.0.0.2 status=up dst_mtu=0 weight=1
bound_if=7 real_if=0 lgwy=static/1 tun=intf mode=dialup/2 encap=none options[0x228]=npu frag-rfc
role=primary accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=1 refcnt=3 ilast=42973580 olast=42973580 ad=/0
stat: rxp=203 txp=0 rxb=31104 txb=0
dpd: mode=on-demand on=0 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
------------------------------------------------------
name=ikev2_native_0 ver=2 serial=6 172.16.200.4:4500->172.16.200.254:64917 nexthop= tun_
id=10.130.1.1 tun_id6=::10.0.0.5 status=up dst_mtu=0 weight=1 country=ZZ
bound_if=7 real_if=0 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none options[0x23a8]=npu rgwy-
chg rport-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=ikev2_native index=0
proxyid_num=1 child_num=0 refcnt=6 ilast=0 olast=42973580 ad=/0
stat: rxp=72 txp=0 rxb=11248 txb=0
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
```

```
natt: mode=keepalive draft=0 interval=10 remote_port=64917
fec: egress=0 ingress=0
proxyid=ikev2_native proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.130.1.1-10.130.1.1:0
  SA:  ref=4 options=4a7 type=00 soft=0 mtu=1280 expire=43173/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43189/43200
  dec: spi=5c8e4a53 esp=aes key=32
34420e33f9e18eea020278fdeadfc111aa099f84e1dd2a0f25ec20fd242bff9f
       ah=sha1 key=20 4a4f61a0a89017d5ac711be76adb3631b1f24be5
  enc: spi=ec7bfcfc esp=aes key=32
80a0b64bdf7e1caca170a0641dca00dd657629af8913c5baeacc209edc4ccac3
       ah=sha1 key=20 ebd8bc8a2f88cd2dc8a2232fdadb7fc627694a77
  dec:pkts/bytes=72/11248, enc:pkts/bytes=0/0
  npu_flag=02 npu_rgwy=172.16.200.254:64917 npu_lgwy=172.16.200.4:4500npu_selid=3
  dec_npuid=1 enc_npuid=0 dec_engid=-1 enc_engid=-1 dec_saidx=3 enc_saidx=-1
```

```
# diagnose debug enable
# diagnose debug app ike -1
# diagnose debug app fnbam -1

ike V=root:0: udp comes 172.16.200.254:1012->172.16.200.4:500,ifindex=7,vrf=0,len=624....
ike V=root:0: IKEv2 exchange=SA_INIT id=bc5aa3bb99c0963c/0000000000000000 len=624
...
ike V=root:0:ikev2_native:89: reassembled fragmented message
ike V=root:0:ikev2_native:89: responder received AUTH msg
ike V=root:0:ikev2_native:89: processing notify type MOBIKE_SUPPORTED
ike V=root:0:ikev2_native:89: peer identifier IPV4_ADDR 172.17.249.189
ike V=root:0:ikev2_native:89: eap-peer=yes
ike V=root:0:ikev2_native:89: re-validate gw ID
ike V=root:0:ikev2_native:89: gw validation OK
...
ike V=root:0:ikev2_native:89: responder preparing EAP identity request
ike V=root:0:ikev2_native:89: local cert, subject='172.18.68.209', issuer='eku-ca'
ike V=root:0:ikev2_native:89: local CA cert, subject='eku-ca', issuer='eku-ca'
...
ike V=root:0:ikev2_native:89: sent IKE msg (AUTH_RESPONSE): 172.16.200.4:4500-
>172.16.200.254:64917, len=148, vrf=0, id=bc5aa3bb99c0963c/290cf660b0977bf5:00000001, oif=7
ike V=root:0: udp comes 172.16.200.254:64917->172.16.200.4:4500,ifindex=7,vrf=0,len=100....
ike V=root:0: IKEv2 exchange=AUTH id=bc5aa3bb99c0963c/290cf660b0977bf5:00000002 len=96
ike 0: in
BC5AA3BB99C0963C290CF660B0977BF52E2023080000000020000006030000044 2BFE21A653DCF88A4F0CA11AD6C68644EC
CF0388EE7FB9AB1E00C94DD75AE4A1D24007C230347591E0B69DADA0D9878F315EDC6D47851E78EF816D43CBA95A67
ike 0:ikev2_native:89: dec
BC5AA3BB99C0963C290CF660B0977BF52E2023080000000020000003130000004000000 11025A000D01746573745F726164
ike V=root:0:ikev2_native:89: responder received EAP msg
ike V=root:0:ikev2_native:89: send EAP message to FNBAM
ike V=root:0:ikev2_native:89: initiating EAP authentication
ike V=root:0:ikev2_native: EAP user "test_rad"
ike V=root:0:ikev2_native: auth group radius-group
ike V=root:0:ikev2_native: EAP 24730597347332 pending
```

```
...
ike V=root:0:ikev2_native:89 EAP 24730597347332 result FNBAM_CHALLENGED
[1098] __rad_del_job_timer-
[1386] freeze_auth_session-
ike V=root:0:ikev2_native: EAP challenged for user "test_rad"
ike V=root:0:ikev2_native:89: responder preparing EAP pass through message
ike 0:ikev2_native:89: enc 0000001A015B00160410812E67D63D7945B8A54A6049EEA98733050403020105
ike 0:ikev2_native:89: out
BC5AA3BB99C0963C290CF660B0977BF52E202320000000020000006030000044669A4D0DFD7FC82E33253FD3AEB8ED5D55
0976ED0D67F21B71DB16A0ABEB36BC652DCC055A81D59E5A36C27E1BE94EF9135A01ED5E27E4009BBB1DEF26EC8925
ike V=root:0:ikev2_native:89: sent IKE msg (AUTH_RESPONSE): 172.16.200.4:4500-
>172.16.200.254:64917, len=96, vrf=0, id=bc5aa3bb99c0963c/290cf660b0977bf5:00000002, oif=7
ike V=root:0: udp comes 172.16.200.254:64917->172.16.200.4:4500,ifindex=7,vrf=0,len=84....
ike V=root:0: IKEv2 exchange=AUTH id=bc5aa3bb99c0963c/290cf660b0977bf5:00000003 len=80
ike 0: in
BC5AA3BB99C0963C290CF660B0977BF52E2023080000000300000050300000 3439E6158679F4F96936D70706661FE64307
3C60362D82775DCB70395B4B3FC89F31EF9D45E142EB6BB224B76BE92F1EE3
ike 0:ikev2_native:89: dec
BC5AA3BB99C0963C290CF660B0977BF52E2023080000000300000002A300000040000000A025B0006031A
ike V=root:0:ikev2_native:89: responder received EAP msg
ike V=root:0:ikev2_native:89: send EAP message to FNBAM
ike V=root:0:ikev2_native: EAP 24730597347332 pending
...
[672] __group_match-Use 'test_rad' for user group matching.
[710] __group_match-Check if radius-server is a group member
[716] __group_match-Group 'radius-group' passed group matching
[719] __group_match-Add matched group 'radius-group'(7)
[276] find_matched_usr_grps-Passed group matching
[1024] update_auth_token_session-config does not require 2fa
[336] fnbamd_comm_send_result-Sending result 0 (nid 0) for req 24730597347332, len=2784
[668] destroy_auth_silent-delete session 24730597347332
[1938] handle_child_rsp-Auth rsp 24730597347332, session created, line 126
[1650] fnbamd_rad_stop-
[1271] __rad_stop-
[545] __rad_udp_close-closed.
[1266] __rad_conn_stop-Stop rad conn timer.
[1098] __rad_del_job_timer-
ike V=root:0:ikev2_native:89 EAP 24730597347332 result FNBAM_SUCCESS
ike V=root:0:ikev2_native: EAP succeeded for user "test_rad" group "radius-group" 2FA=no
ike V=root:0:ikev2_native:89: responder preparing EAP pass through message
...
ike V=root:0:ikev2_native:89: responder received AUTH msg
ike V=root:0:ikev2_native:89: peer identifier IPV4_ADDR 172.17.249.189
ike V=root:0:ikev2_native:89: eap-peer=yes
ike V=root:0:ikev2_native:89: auth verify done
ike V=root:0:ikev2_native:89: responder AUTH continuation
ike V=root:0:ikev2_native:89: authentication succeeded
ike V=root:0:ikev2_native:89: responder creating new child
ike V=root:0:ikev2_native:89: mode-cfg type 1 request 0:''
ike V=root:0:ikev2_native: mode-cfg allocate 10.130.1.1/0.0.0.0
ike V=root:0:ikev2_native:89: mode-cfg using allocated IPv4 10.130.1.1
...
```

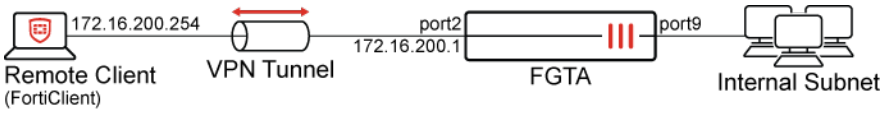
```
ike V=root:0:ikev2_native_0: tunnel created tun_id 10.130.1.1/::10.0.0.5 remote_location 0.0.0.0
ike V=root:0:ikev2_native_0: added new dynamic tunnel for 172.16.200.254:64917
ike V=root:0:ikev2_native_0:89: established IKE SA bc5aa3bb99c0963c/290cf660b0977bf5
ike V=root:0:ikev2_native_0:89: check peer route: if_addr4_rcvd=0, if_addr6_rcvd=0, mode_cfg=1
ike V=root:0:ikev2_native_0:89: mode-cfg assigned (1) IPv4 address 10.130.1.1
ike V=root:0:ikev2_native_0:89: mode-cfg send (3) IPv4 DNS(1) 172.17.254.148
ike V=root:0:ikev2_native_0:89: mode-cfg send (3) IPv4 DNS(2) 8.8.8.8
ike V=root:0:ikev2_native_0:89: mode-cfg IPv6 DNS ignored, no IPv6 DNS servers found
ike V=root:0:ikev2_native_0:89:ikev2_native:1: replay protection enabled
ike V=root:0:ikev2_native_0:89:ikev2_native:1: set sa life soft seconds=43189.
ike V=root:0:ikev2_native_0:89:ikev2_native:1: set sa life hard seconds=43200.
ike V=root:0:ikev2_native_0:89:ikev2_native:1: IPsec SA selectors #src=1 #dst=1
ike V=root:0:ikev2_native_0:89:ikev2_native:1: src 0 7 0:0.0.0.0-255.255.255.255:0
ike V=root:0:ikev2_native_0:89:ikev2_native:1: dst 0 7 0:10.130.1.1-10.130.1.1:0
ike V=root:0:ikev2_native_0:89:ikev2_native:1: add dynamic IPsec SA selectors 483
ike V=root:0:ikev2_native_0:89:ikev2_native:1: added dynamic IPsec SA proxyids new 1 483
ike V=root:0:ikev2_native:1: add route 10.130.1.1/255.255.255.255 gw 10.130.1.1 oif ikev2_native
(49) metric 15 priority 1
ike V=root:0:ikev2_native_0:89:ikev2_native:1: tunnel 1 of VDOM limit 0/0
ike V=root:0:ikev2_native_0:89:ikev2_native:1: add IPsec SA: SPIs=5c8e4a53/ec7bfcfc
ike 0:ikev2_native_0:89:ikev2_native:1: IPsec SA dec spi 5c8e4a53 key
32:34420E33F9E18EEA020278FDEADFC111AA099F84E1DD2A0F25EC20FD242BFF9F auth
20:4A4F61A0A89017D5AC711BE76ADB3631B1F24BE5
ike 0:ikev2_native_0:89:ikev2_native:1: IPsec SA enc spi ec7bfcfc key
32:80A0B64BDF7E1CACA170A0641DCA00DD657629AF8913C5BAEACC209EDC4CCAC3 auth
20:EBD8BC8A2F88CD2DC8A2232FDADB7FC627694A77
ike V=root:0:ikev2_native_0:89:ikev2_native:1: added IPsec SA: SPIs=5c8e4a53/ec7bfcfc
ike V=root:0:ikev2_native_0:89:ikev2_native:1: sending SNMP tunnel UP trap
ike V=root:0:ikev2_native_0: tunnel up event assigned address 10.130.1.1
...
```

Default to IPAM addressing for remote access VPNs in VPN Wizard

The VPN wizard for creating remote access and SIA tunnels defaults to using IPAM for addressing, instead of manually entering the client IP range. When IPAM is not enabled, an option is displayed in the wizard to enable it and to assign basic IPAM parameters. Once enabled, administrators only need to select a network size in order to automatically assign a block of addresses from the IPAM address pool to the VPN clients.

Example

In this example, we enable IPAM and assign a default block of addresses for use by IPAM. Using the Remote Access VPN Wizard, we create the VPN settings and reserve a 256 address block for VPN client address assignments. IPAM automatically assigns the address range with 256 addresses from the default address block. This simplifies the role of the admin in reserving and defining an address range manually for client endpoints.

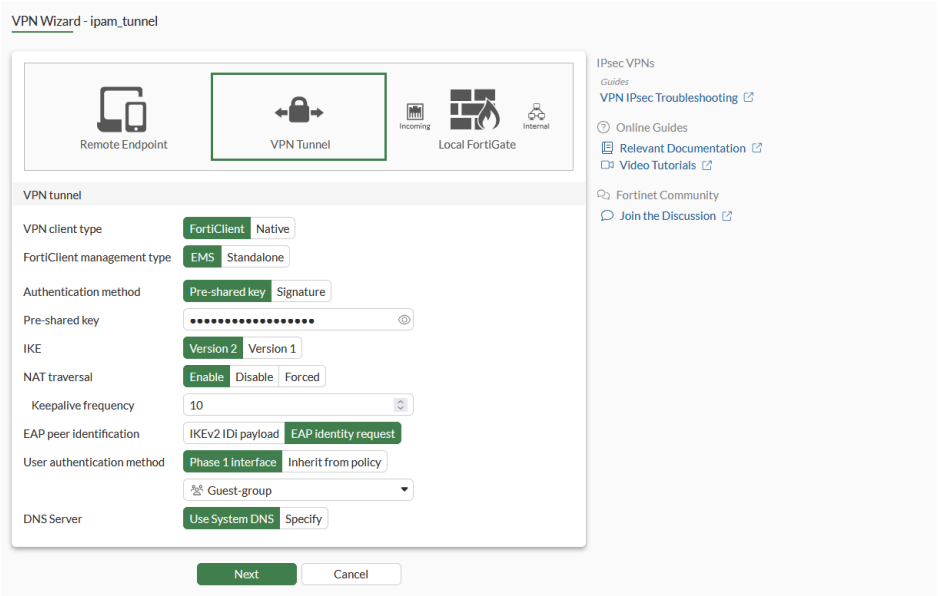


To enable IPAM and IPAM Subnet in the GUI:

1. Go to *Network* > *IPAM* and select the *IPAM Settings* tab.
2. Set *Status* to *Enabled*.
3. Click *OK*.
4. Go to the *IPAM Subnets* tab.
5. Define a subnet to allow IPAM to assign addresses from, for example 172.31.0.0 255.255.0.0.
6. Click *OK*.

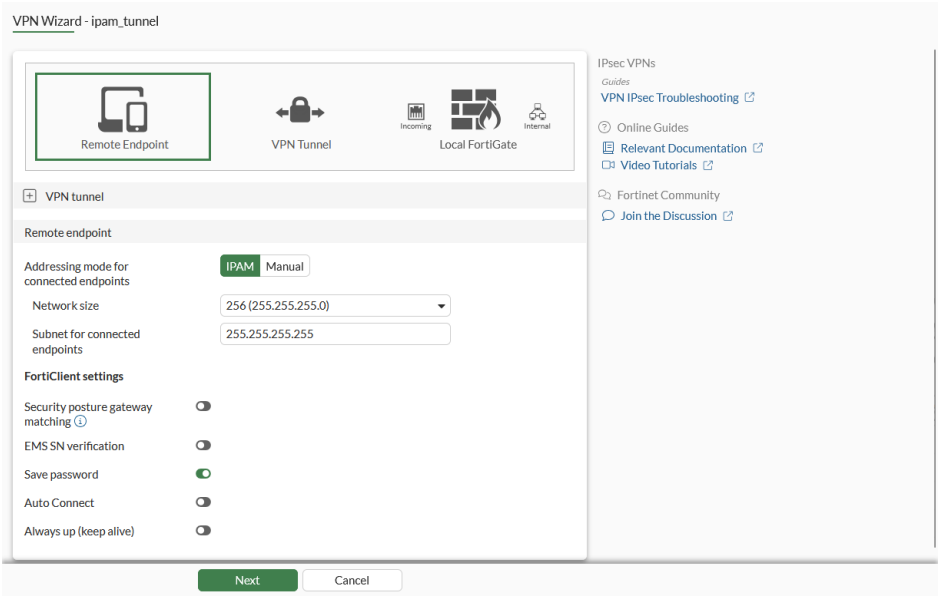
To configure an IPsec tunnel from template in the GUI:

1. Go to *VPN* > *VPN Wizard*.
2. Enter a *Tunnel name* and select *Remote Access*, then click *Begin*.
3. Set *VPN client type* to *FortiClient* and *FortiClient management type* to *EMS*.
4. Set *Authentication method* to *Pre-shared key* and enter the *Pre-shared key*.
5. Set *User authentication method* to *Phase 1 interface* and select a user group.

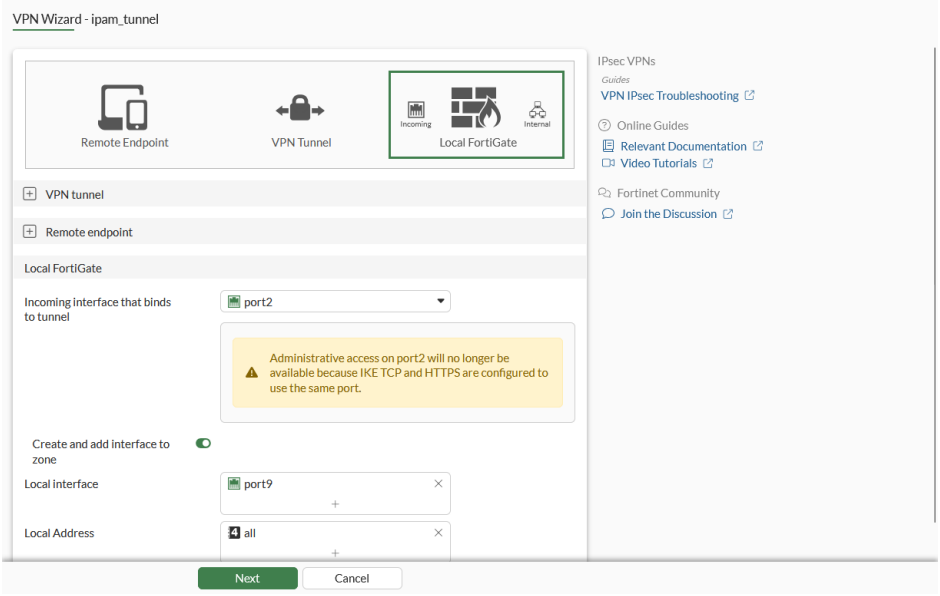


6. Click *Next*.

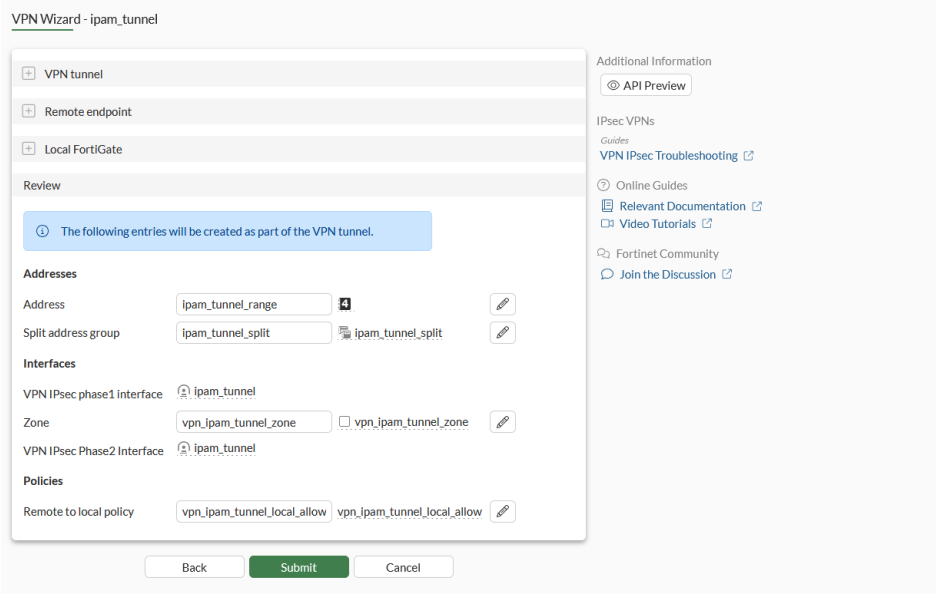
   Addressing mode for connected endpoints is set to *IPAM* by default. If IPAM was not previously enabled, an option *Edit IPAM Settings* is shown.
7. Set the *Network size*. This will reserve a range of addresses in the IPAM address range for VPN clients.



8. Leave *Subnet for connected endpoints* as is, and click *Next*.
9. Set the *Incoming interface that binds to tunnel*, *Local interface*, and *Local Address*.



10. Click *Next*.



11. Review the settings, then click *Submit*.

To enable IPAM and IPAM Subnet in the CLI:

```
config system ipam
    set status enable
    config pools
        edit "default-pool"
            set subnet 172.31.0.0 255.255.0.0
        next
    end
end
```

To configure an IPsec tunnel in the CLI:

```
config vpn ipsec phase1-interface
    edit "ipam_tunnel"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set comments "VPN: ipam_tunnel -- Created by VPN wizard"
        set dhgrp 20 21
        set eap enable
        set eap-identity send-request
        set wizard-type dialup-forticlient
        set authusrgrp "local-group"
        set assign-ip-from name
```

```
        set dns-mode auto
        set ipv4-split-include "ipam_tunnel_split"
        set ipv4-name "ipam_tunnel_range"
        set save-password enable
        set psksecret PLAIN "0x313233343536"
    next
end
config vpn ipsec phase2-interface
    edit "ipam_tunnel"
        set phase1name "ipam_tunnel"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
        set comments "VPN: ipam_tunnel -- Created by VPN wizard"
    next
end
config firewall address
    edit "ipam_tunnel_range"
        set type ipam
        set comment "VPN: ipam_tunnel -- Created by VPN wizard"
        set subnet 172.31.0.0 255.255.255.0
    next
end

config firewall policy
    edit 1
        set srcintf "any"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
    edit 2
        set name "vpn_ipam_tunnel_local_allow"
        set srcintf "ipam_tunnel"
        set dstintf "port9"
        set action accept
        set srcaddr "ipam_tunnel_range"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
        set comments "VPN: ipam_tunnel -- Created by VPN wizard"
    next
end
```

The firewall address type is set to `ipam`, see Firewall address support for IPAM rules on page 59.

Display interface and tunnel alias in various diagnose and get commands

An interface alias can be displayed in various diagnose and get commands to more easily identify the name of the interface or VPN tunnel. The interface alias name is also extended from 15 characters to 25 characters.

These commands will display the interface alias:

```
diagnose vpn tunnel list
```

```
diagnose vpn ike gateway list
```

```
diagnose netlink interface list [interface-name]
```

```
diagnose sniffer packet [interface-name] '' 4
```

```
get vpn ipsec tunnel summary
```

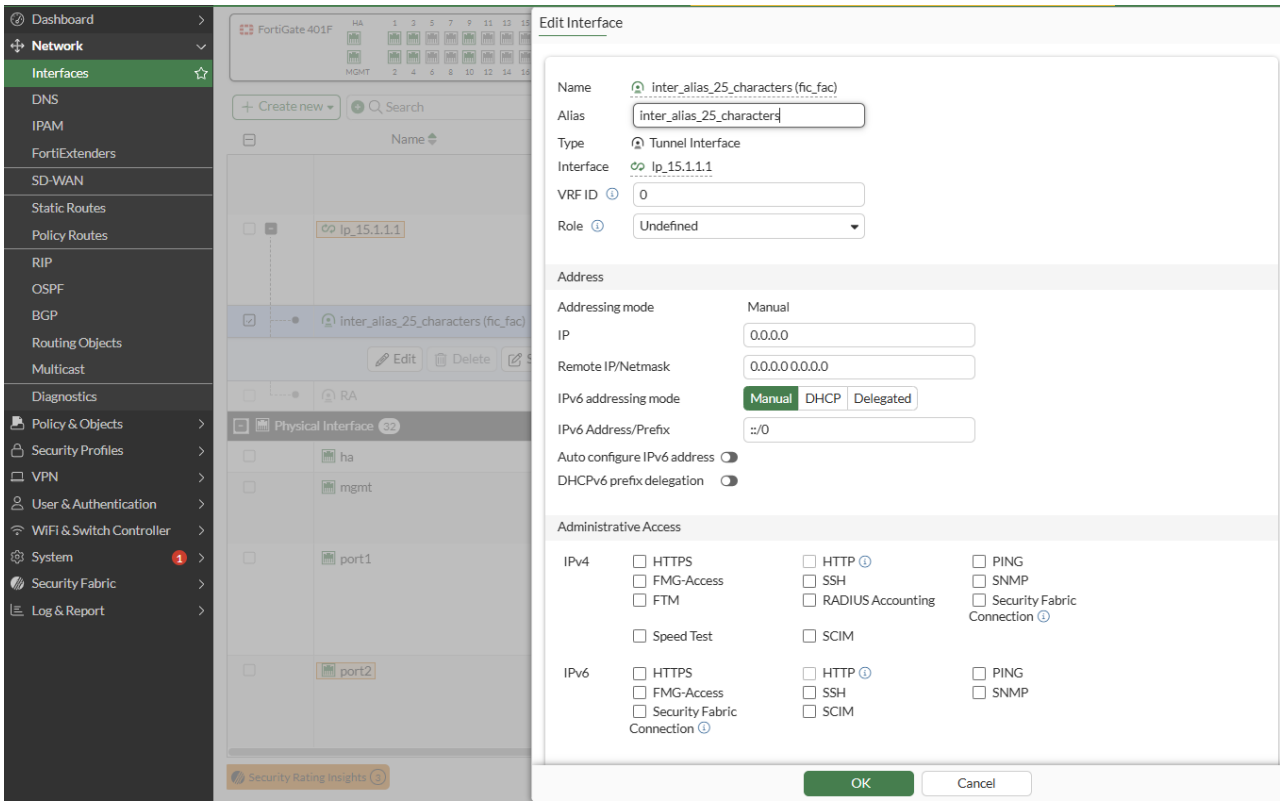
```
get vpn ike gateway
```

Example

In this example, a VPN tunnel is created named *fic_fac*. The administrator edits the VPN tunnel interface and gives it an alias. In debugs, the alias name can be displayed.

To configure an interface in the GUI:

1. Go to *Network* > *Interfaces*.
2. Click *Create New* or *Edit* > *Interface*.
3. Enter an alternate name in the *Alias* field. The maximum length of the alias is 25 characters.



4. Click *OK*.

To configure an interface in the CLI:

```
config system interface
    edit "fic_fac"
        set vdom "root"
        set type tunnel
        set alias "inter_alias_25_characters"
        set snmp-index 43
        set interface "lp_15.1.1.1"
    next
end
```

To configure a VPN in the CLI:

```
config vpn ipsec phase1-interface
    edit "fic_fac"
        set type dynamic
        set interface "lp_15.1.1.1"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
```

```
        set dhgrp 20 21
        set eap enable
        set eap-identity send-request
        set wizard-type dialup-forticlient
        set authusrgrp "fic_fac_user_groups"
        set assign-ip-from name
        set dns-mode auto
        set ipv4-split-include "fic_fac_split"
        set ipv4-name "fic_fac_range"
        set save-password enable
        set psksecret *
    next
end
config vpn ipsec phase2-interface
    edit "fic_fac"
        set phase1name "fic_fac"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

To check the debugs:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=fic_fac ver=2 serial=3 15.1.1.1:0->0.0.0.0:0 nexthop= tun_id=10.0.0.2 tun_id6=::10.0.0.3
status=up dst_mtu=0 weight=1
bound_if=0 real_if=0 lgwy=static/1 tun=intf mode=dialup/2 encap=none options[0x228]=npu frag-rfc
role=primary accept_traffic=1 overlay_id=0

alias=inter_alias_25_characters
proxyid_num=0 child_num=1 refcnt=3 ilast=42954707 olast=42954707 ad=/0
stat: rxp=9972 txp=16780 rxb=2056148 txb=19407385
dpd: mode=on-demand on=0 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
------------------------------------------------------
name=fic_fac_0 ver=2 serial=5 15.1.1.1:4500->172.16.200.242:65224 nexthop=0.0.0.0 tun_
id=10.122.1.2 tun_id6=::10.0.0.5 status=up dst_mtu=1500 weight=1 country=ZZ
bound_if=0 real_if=7 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none options[0x23a8]=npu rgwy-
chg rport-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=fic_fac index=0 alias=inter_alias_25_characters
proxyid_num=1 child_num=0 refcnt=6 ilast=4 olast=4 ad=/0
stat: rxp=9508 txp=16439 rxb=1923368 txb=19262090
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=15
natt: mode=keepalive draft=0 interval=10 remote_port=65224
fec: egress=0 ingress=0
proxyid=fic_fac proto=0 sa=1 ref=3 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.122.1.2-10.122.1.2:0
```

```
  SA:  ref=6 options=6a6 type=00 soft=0 mtu=1422 expire=41436/0B replaywin=2048
       seqno=4410 esn=0 replaywin_lastseq=00002400 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43187/43200
  dec: spi=7e0b8dec esp=aes key=32
999d2cdafd2740f2bc75701137a0e697957c031e8fc688f196f43447a5eb0444
       ah=sha256 key=32 4dce873c9c54efd53231b4f21f08f75ae52e3bbbe2c7a9a8e5b0a73d0372f6df
  enc: spi=406f058f esp=aes key=32
7f3406a641764e3f0b8bb2a978ee9663bf1a5ea3afa47877cf3ca0d2d4b509a3
       ah=sha256 key=32 0e63fb90879d0fde12c1503644a944679b044b92ff96be24acca15c14fc24b1f
  dec:pkts/bytes=9508/1923368, enc:pkts/bytes=16439/19262090
  npu_flag=03 npu_rgwy=172.16.200.242:65224 npu_lgwy=15.1.1.1:4500npu_selid=2
  dec_npuid=1 enc_npuid=1 dec_engid=-1 enc_engid=-1 dec_saidx=5 enc_saidx=2
```

```
# diagnose vpn ike gateway list

vd: root/0
name: fic_fac_0/inter_alias_25_characters
version: 2
interface:  0
addr: 15.1.1.1:4500 -> 172.16.200.242:65224
tun_id: 10.122.1.2/::10.0.0.5
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 1763s ago
eap-user: u2
2FA: no
peer-id: 172.16.200.242
peer-id-auth: no
FortiClient UID: 6108A9179A5C40D7BD57504E15114C1F
assigned IPv4 address: 10.122.1.2/255.255.255.255
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 90/90/90 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 4 b371bacfe004baad/a942a5382b39098f
  direction: responder
  status: established 1763-1763s ago = 90ms
  proposal: aes256-sha256
  child: no
  SK_ei: ddddf2d839bfc32c-0d887c0894914c60-2af3b45f28e79dd2-122b5635e1f3f1c0
  SK_er: 733f7137bd59743b-e3b1d2d75c30b5d2-1b8fb35a330664b2-1481e361e223f64c
  SK_ai: b34c5ac2ae945046-ba98de12b9aec6f5-979f920d656daddf-eb702b094145ae54
  SK_ar: d64e726824012f32-e331aa4c58b509e8-c763084ab2bc2f01-81bd2b12fed10bb5
  PPK: no
  message-id sent/recv: 1/94
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/84366
```

```
  DPD sent/recv: 00000001/00000001
  peer-id: 172.16.200.242
```

```
# diagnose sniffer packet fic_fac ' ' 4 0 1
interfaces=[fic_fac/inter_alias_25_characters]
filters=[ ]
pcap_lookupnet: fic_fac: no IPv4 address assigned
17.806578 fic_fac -- 10.122.1.2 -> 11.101.1.1: icmp: echo request
17.806615 fic_fac -- 11.101.1.1 -> 10.122.1.2: icmp: echo reply
20.110606 fic_fac -- 10.122.1.2 -> 11.101.1.1: icmp: echo request
20.110617 fic_fac -- 11.101.1.1 -> 10.122.1.2: icmp: echo reply
21.497194 fic_fac -- 10.122.1.2.53486 -> 172.18.58.85.8013: syn 2827809601
21.498994 fic_fac -- 172.18.58.85.8013 -> 10.122.1.2.53486: syn 2024540250 ack 2827809602
21.499271 fic_fac -- 10.122.1.2.53486 -> 172.18.58.85.8013: ack 2024540251
21.499769 fic_fac -- 172.18.58.85.8013 -> 10.122.1.2.53486: ack 2827810842
21.506144 fic_fac -- 10.122.1.2.53486 -> 172.18.58.85.8013: fin 2827811617 ack 2024541040
21.506506 fic_fac -- 172.18.58.85.8013 -> 10.122.1.2.53486: fin 2024541064 ack 2827811618
21.507673 fic_fac -- 10.122.1.2.53486 -> 172.18.58.85.8013: rst 2827811618 ack 2024541064
22.423492 fic_fac -- 10.122.1.2 -> 11.101.1.1: icmp: echo request
22.423501 fic_fac -- 11.101.1.1 -> 10.122.1.2: icmp: echo reply
```

Support SM3/SM4 cryptographic algorithms for IKEv1/ IKEv2

The ShangMi SM4 encryption algorithm and SM3 hash algorithm have been added as supported proposals for IKEv1 and IKEv2 site-to-site VPN configurations. The SM4-SM3 pair can be selected for Phase1 IKE proposal and Phase2 ESP tunnel proposal.

These encryption algorithms are often used in regulatory requirements in China. They are incorporated into ISO/IEC standards and recognized by IETF in the Internet-Draft for Using ShangMi in the Internet Key Exchange Protocol Version 2 (IKEv2).

For IKEv2:

FortiGate supports the following transform types and names for IKEv2:

Transform Type	Type	Name
1	Encryption Algorithm Transform	ENCR_SM4_CBC
2	Pseudorandom Function Transform	PRF_HMAC_SM3
3	Integrity Algorithm Transform	AUTH_HMAC_SM3

In IKEv2, when SM4-SM3 proposal is chosen:

- HMAC-SM3 is used as the PRF and integrity algorithm.
- SM4-CBC is used for IKE encryption.
- 128-bit IV length defined by SM4-CBC block size.
- For IPsec:

- SM4-CBC is used for payload encryption. IV length: 128 bits.
- HMAC-SM3 is used for ESP integrity. ICV uses the full 256-bit hash output without truncation.

For IKEv1:

Similarly, IKEv1 uses the following transform identifiers:

- OAKLEY_ATTR_ENC_ALG_SM4
- OAKLEY_ATTR_HASH_ALG_SM3
- IPSECDOI_ESP_SM4_CBC
- IPSECDOI_ATTR_AUTH_HMAC_SM3_256

Limitations

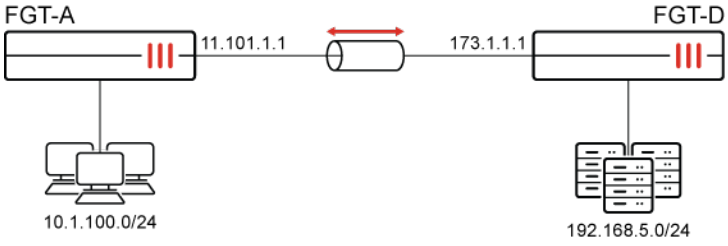
- Any tunnels using SM4-SM3 proposal do not support hardware acceleration.
- It is only available to physical or FortiGate-VMs using x86_64 processors.
- FGT using 3.2 Linux kernel are not supported

Supported FortiGate models:

- 200G series
- 400F
- 600F
- 900G
- All high-end F-series from 1000F and above
- 7000F
- FG-VM64

Example

SM4-SM3 is used in the IKE proposal in IKEv2 to meet regulatory requirement for a site-to-site VPN.

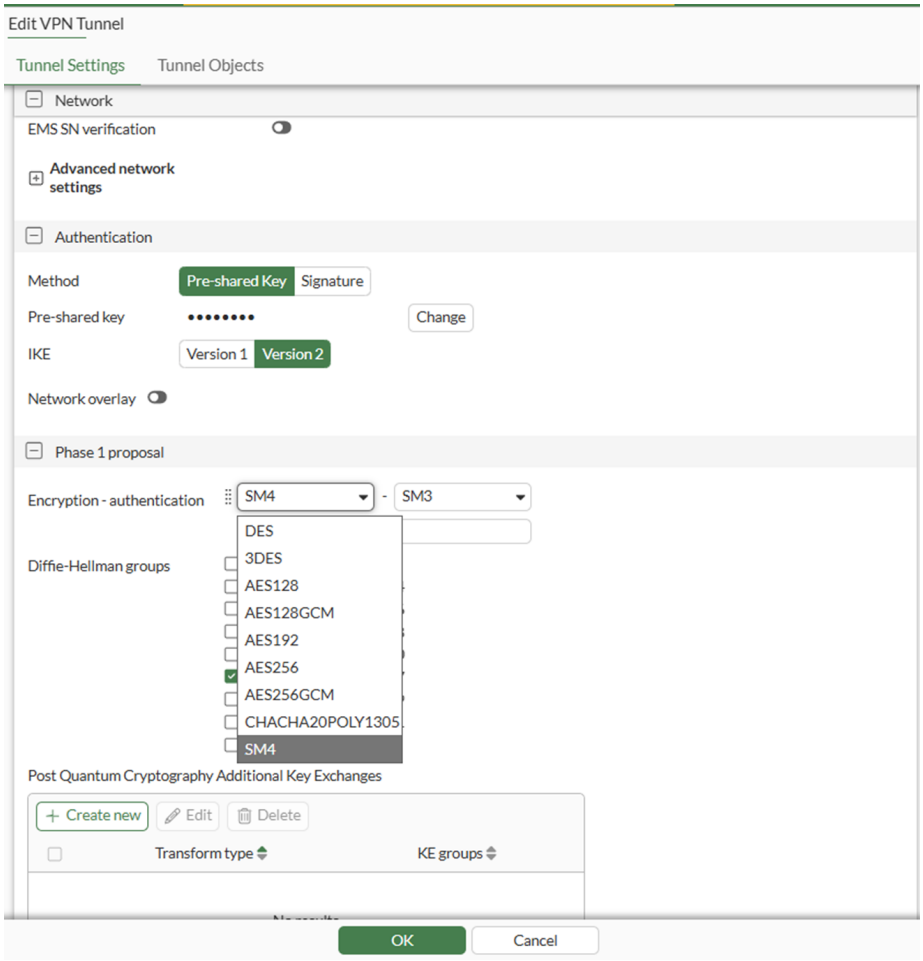


To configure in the GUI:

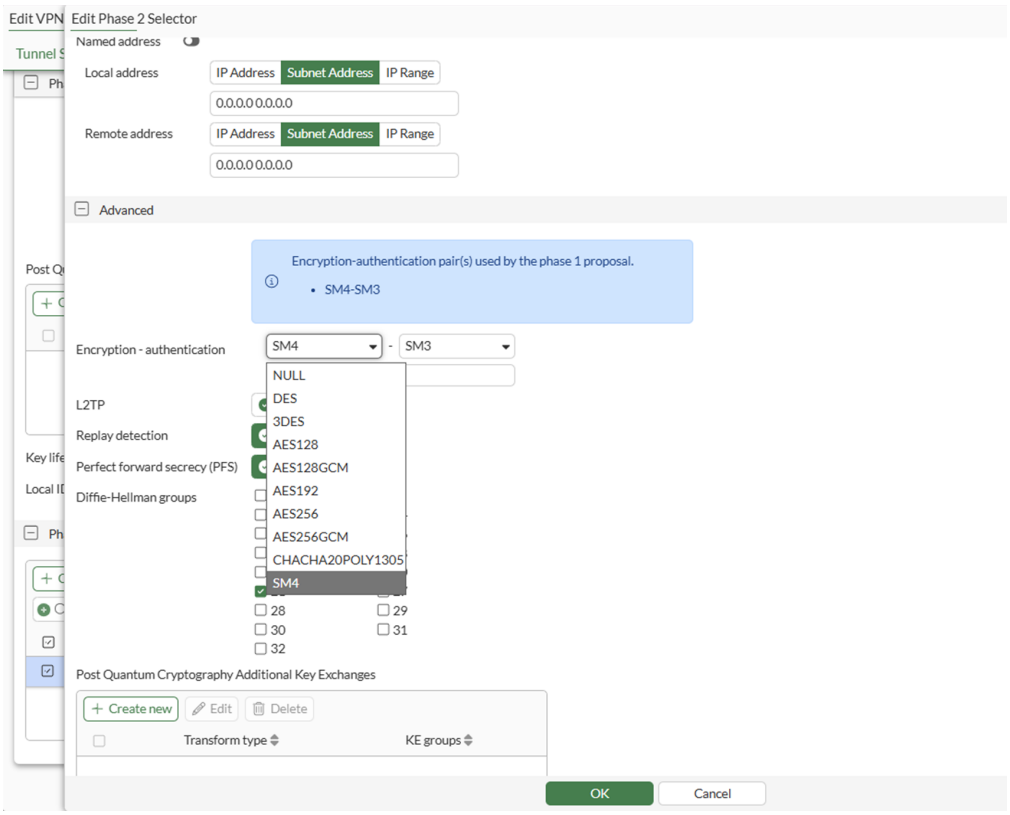
Exact steps are not provided in detail. See CLI for exact configurations.

1. On FGT-A, go to *VPN > VPN Tunnels* and create a new tunnel.
2. Set *Remote gateway* to *Static IP address*, and configure the *Network* and *Authentication* settings as needed.

3. For Phase 1 proposal:
   a. Remove all default proposals.
   b. Select *SM4* for the *Encryption* algorithm. SM3 will be selected automatically for the *authentication* algorithm.



4. For Phase 2, create a new selector:
   a. Under *Advanced*, remove all default proposals.
   b. Select *SM4* for the *Encryption* algorithm. SM3 will be selected automatically for the *authentication* algorithm.
   c. Click *OK*.



5. Click *OK*.
6. Similarly configure SM4-SM3 proposals in the VPN configurations on your remote FortiGate FGT-D.

To configure in the CLI:

```
config system settings
    set ike-proposal-visibility all
end
```

See Show only recommended IKE proposals by default on page 356.

```
config vpn ipsec phase1-interface
    edit "HUB"
        set type dynamic
        set interface "port13"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal sm4-sm3
        set dhgrp 20 21
        set psksecret <password>
    next
end
```

```
config vpn ipsec phase2-interface
    edit "HUB"
```

```
        set phase1name "HUB"
        set proposal sm4-sm3
        set dhgrp 20 21
    next
end
```

```
config firewall policy
    edit 1
        set srcintf "any"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
    next
end
```

This policy is for example purposes only. Do not configure an any-to-any allow policy in production.

Verification

Use the following CLI commands to view the real-time debugs when establishing the VPN tunnel. Note the proposals that were offered and matched in the three transform types (ENCRYPTION, INTEGRITY, PRF) for IKE (P1) and for ESP (P2).

```
# diagnose debug application ike -1
# diagnose debug enable

ike V=root:0:toHUB:376: incoming proposal:
ike V=root:0:toHUB:376: proposal id = 1:
ike V=root:0:toHUB:376:   protocol = IKEv2:
ike V=root:0:toHUB:376:      encapsulation = IKEv2/none
ike V=root:0:toHUB:376:         type=ENCR, val=SM4_CBC (key_len = 128)
ike V=root:0:toHUB:376:         type=INTEGR, val=AUTH_HMAC_SM3
ike V=root:0:toHUB:376:         type=PRF, val=PRF_HMAC_SM3
ike V=root:0:toHUB:376:         type=DH_GROUP, val=ECP384.
ike V=root:0:toHUB:376: matched proposal id 1
ike V=root:0:toHUB:376: proposal id = 1:
ike V=root:0:toHUB:376:   protocol = IKEv2:
ike V=root:0:toHUB:376:      encapsulation = IKEv2/none
ike V=root:0:toHUB:376:         type=ENCR, val=SM4_CBC (key_len = 128)
ike V=root:0:toHUB:376:         type=INTEGR, val=AUTH_HMAC_SM3
ike V=root:0:toHUB:376:         type=PRF, val=PRF_HMAC_SM3
ike V=root:0:toHUB:376:         type=DH_GROUP, val=ECP384.
...
ike V=root:0:toHUB:376:toHUB:495: incoming child SA proposal:
ike V=root:0:toHUB:376:toHUB:495: proposal id = 1:
```

```
ike V=root:0:toHUB:376:toHUB:495:   protocol = ESP:
ike V=root:0:toHUB:376:toHUB:495:      encapsulation = TUNNEL
ike V=root:0:toHUB:376:toHUB:495:         type=ENCR, val=SM4_CBC (key_len = 128)
ike V=root:0:toHUB:376:toHUB:495:         type=INTEGR, val=SM3
ike V=root:0:toHUB:376:toHUB:495:         type=ESN, val=NO
ike V=root:0:toHUB:376:toHUB:495:         PFS is disabled
ike V=root:0:toHUB:376:toHUB:495: matched proposal id 1
ike V=root:0:toHUB:376:toHUB:495: proposal id = 1:
ike V=root:0:toHUB:376:toHUB:495:   protocol = ESP:
ike V=root:0:toHUB:376:toHUB:495:      encapsulation = TUNNEL
ike V=root:0:toHUB:376:toHUB:495:         type=ENCR, val=SM4_CBC (key_len = 128)
ike V=root:0:toHUB:376:toHUB:495:         type=INTEGR, val=SM3
ike V=root:0:toHUB:376:toHUB:495:         type=ESN, val=NO
ike V=root:0:toHUB:376:toHUB:495:         PFS is disabled
...
ike V=root:0:toHUB:376:toHUB:495: sending SNMP tunnel UP trap
ike V=root:0:toHUB: static tunnel up event 0.0.0.0 (dev=53)
ike V=root:0:toHUB: static tunnel up event :: (dev=53)
```

Use the following CLI commands to verify the tunnel is established with the SM4-SM3 proposal:

```
# diagnose vpn ike gateway list name <tunnel>

vd: root/0
name: toHUB
version: 2
interface: port9 15
addr: 11.101.1.1:500 -> 173.1.1.1:500
tun_id: 173.1.1.1/::173.1.1.1
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 146s ago
peer-id: 173.1.1.1
peer-id-auth: no
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 1/1  established 1/1  time 10/10/10 ms

  id/spi: 375 b276017ea711477c/e963d7bfba066129
  direction: initiator
  status: established 146-146s ago = 10ms
  proposal: sm4_128-sm3
  child: no
  SK_ei: 152ebe7d097da0e6-459a04571cc85cc0
  SK_er: 8613e0506e88b1df-2129cd5e00349db8
  SK_ai: d3a2a1a84a6581c1-1a97633532fa0d51-829a21602f0276f7-b8e79b2a888e2204
  SK_ar: fa08e32590ed6d7c-f27a4e6dc8159c12-be0f8cf0b1b56483-a955315d5caf2fa2
  PPK: no
  message-id sent/recv: 9/1
  QKD: no
  PQC-KEM (IKE): no
```

```
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/85953
  DPD sent/recv: 00000033/00000033
  peer-id: 173.1.1.1
```

Similarly, use the following CLI commands to show the encryption and authentication algorithms used in the IPsec SAs.

```
# diagnose vpn tunnel list name <tunnel>

list all ipsec tunnel in vd 0
------------------------------------------------------
name=toHUB ver=2 serial=3 11.101.1.1:0->173.1.1.1:0 nexthop=11.101.1.2 tun_id=173.1.1.1 tun_
id6=::173.1.1.1 status=up dst_mtu=1500 weight=1 country=US
bound_if=15 real_if=15 lgwy=static/1 tun=intf mode=auto/1 encap=none options[0x228]=npu frag-rfc
run_state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=15 olast=0 ad=/0
stat: rxp=0 txp=146 rxb=0 txb=12264
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=51
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=toHUB proto=0 sa=1 ref=3 serial=4
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=10226 type=00 soft=0 mtu=1422 expire=42749/0B replaywin=2048
       seqno=93 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42900/43200
  dec: spi=c7c6a1b5 esp=sm4 key=16 869f98b2bd18bdfd7211c0db9488e850
       ah=sm3 key=32 bfaff448ea0b23bca803420429d871a8d349ff12089bf277f89258d2567b3f3d
  enc: spi=d5e84636 esp=sm4 key=16 83ac4c93abfb1b57e6d06d3e4da595bf
       ah=sm3 key=32 832da674f1beb4fd9a8a01c3a992a55cff12792fb72e7bd38f0d21ca17cc586a
  dec:pkts/bytes=0/0, enc:pkts/bytes=146/12264
  npu_flag=20 npu_rgwy=0.0.0.0:0 npu_lgwy=0.0.0.0:0 npu_selid=6
  dec_npuid=0 enc_npuid=0 dec_engid=-1 enc_engid=-1 dec_saidx=-1 enc_saidx=-1
```

Show only recommended IKE proposals by default

FortiGate now by default only shows recommended configurations for IKE proposals in the CLI. This helps administrators more easily define secured and recommended cryptographic algorithms in the VPN configurations that adhere to security best practices.

The following DH groups are available in the CLI by default:

- 15, 16, 17, 18, 19, 20, 21, 28, 29, 30, 31, 32 (1,2,5,14,27 are hidden)

The following encryption algorithms are available in the CLI by default:

- AES128_CBC
- AES128_GCM
- AES192_CBC
- AES256_CBC

- AES256_GCM
- CHACHA20POLY1305

The following authentication algorithms are available in the CLI by default:

- SHA256
- SHA384
- SHA512

To set your preference for displaying IKE proposals:

```
config system settings
    set ike-proposal-visibility {recommended | all}
end
```

User and authentication

This section includes information about user and authentication related new features:

- Authentication on page 358

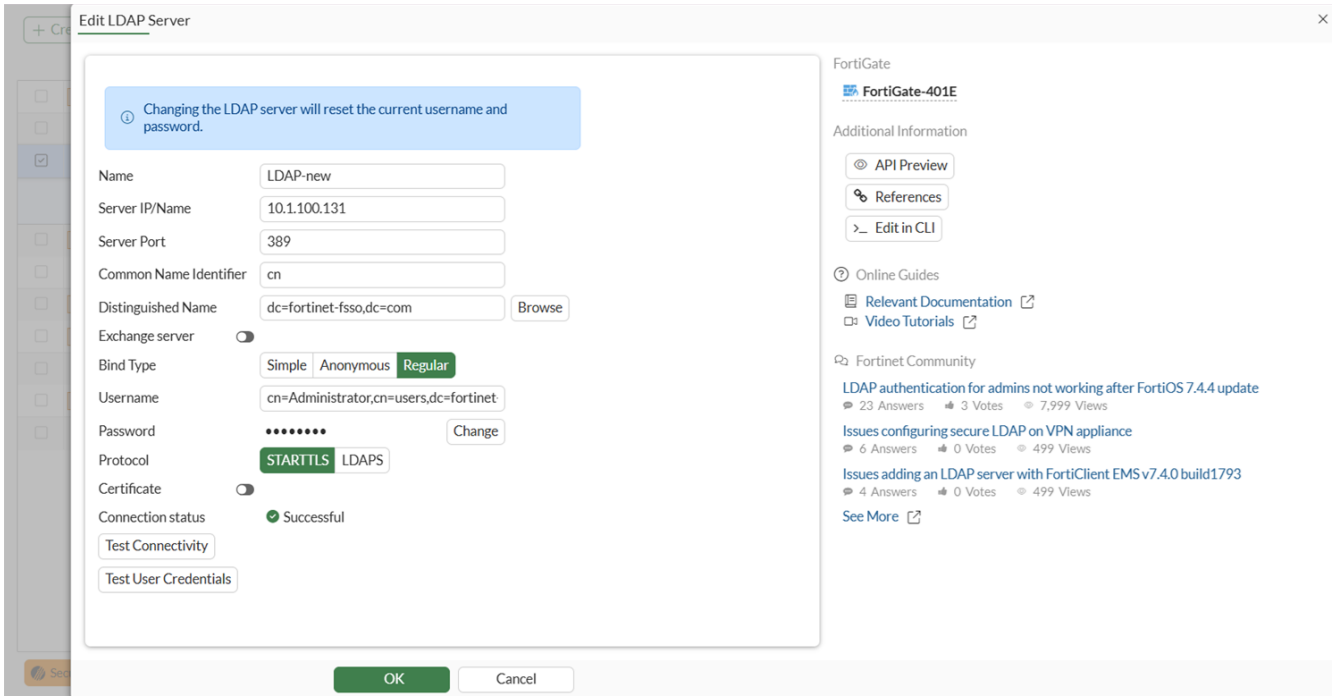
Authentication

This section includes information about authentication related new features:

- Enable secure LDAP connection by default in the GUI on page 358

Enable secure LDAP connection by default in the GUI

In LDAP configurations in the GUI, STARTTLS is selected by default, and there is no option to disable secure connection. Disabling secure connection can still be done in the CLI. After upgrading from an older FortiOS version, if secure connection was disabled then it will remain disabled.



If the remote LDAP server does not support either STARTTLS or LDAPS, communication will fail and the LDAP server will need to be configured to accept LDAPS /STARTTLS connections, which requires a LDAP server certificate issued by a public or local CA in the customer's environment. In case a local CA is issued the LDAP server certificate, the CA public certificate also needs to be imported into the local FortiGate CA root store.

To disable secure connection:

```
config user ldap
    edit <name>
        set secure disable
    next
end
```

LAN Edge

This section includes information about LAN Edge related new features:

- Wireless on page 360
- Switch controller on page 407

Wireless

This section includes information about wireless related new features:

- Support FortiAP management through IPv6 on page 360
- Improve CAPWAP stability on page 362
- Support dynamic redirect URLs from Cisco ISE on page 364
- Support IPsec traffic offloading from the FortiAP on page 367
- Enhance DARRP with FortiAIOps on page 369
- Support up to 16 VAPs on FortiAP G and K-series models on page 372
- Support Wi-Fi 7 MLO on FortiAP K-series models on page 374
- Support FQDN in Layer 3 ACLs on page 377
- Add GUI support for 6 GHz Channel Utilization on page 379
- Support customizable DHCP Option 82 configurations on page 380
- Optimize and update default DARRP parameters on page 383
- Add GUI support for capturing wireless packets on page 384
- Support tuning Clear Channel Assessment threshold values on page 387
- Allow FortiAP radio and FortiWiFi local radio to be disabled on page 391
- Support DVLAN assignments from RADIUS and VLAN pooling together on page 394
- Support multiple FortiAP groups per VLAN entry for VLAN pooling on page 396
- Add RADIUS authentication survivability for 802.1X SSIDs on page 396
- Automated FortiAP location positioning system on page 399
- Support FortiGuest assignments of RADIUS attributes to MPSK groups on page 404

Support FortiAP management through IPv6

In this release, FortiGate can directly manage FortiAPs over IPv6 for all major wireless functions. You can use IPv6 addresses when setting up AP Controller (AC) discovery methods, running CAPWAP data channels, upgrading managed FortiAPs, establishing tunnel-mode and bridge SSID connections, and more.

When FortiAP is managed over IPv6 on an NP7/NP7lite FortiGate, CAPWAP data traffic on IPv4 and IPv6 can offloaded.

CAPWAP offloading is not supported on NP6/NP6xlite FortiGates.

A new CLI command has been added to set an IPv6 multicast address for the FortiGate AC:

```
config wireless-controller global
  set discovery-mc-addr6 <IPv6 address>
end
```

`discovery-mc-addr6`	Multicast IPv6 address for AP discovery.

Ways to configure IPv6 management

You can configure IPv6 addresses on FortiGate to manage FortiAP in the following locations:

1. You can set IPv6 addresses and access options under the IPv6 section of the system interface.

```
config system interface
  edit "port3.66"
    config ipv6
      set ip6-address 2001:18:56:66::1/64
      set ip6-allowaccess ping fabric
      set ip6-send-adv enable
      set ip6-manage-flag enable
      set ip6-other-flag enable
    end
  next
end
```

2. You can set the IPv6 multicast address of an AC to enable FortiAP to discover the FortiGate through the multicast method.

```
config wireless-controller global
    set discovery-mc-addr6 FF02::18C
end
```

3. You can configure a DHCP6 server to enable FortiAP to discover the FortiGate through the DHCP method.

   On a third-party IPv6 DHCP server that FortiAP can reach, configure Option 52 for AC IPv6 Address. In the following example, the FortiGate is used as the DHCP server:

```
config system dhcp6 server
  edit 3
    set subnet 2001:18:56:66::/64
    set interface "port3.66"
    config options
      edit 1
        set code 52
        set type ip6
```

```
        set ip6 2001:18:56:66::1
      next
    end
    config ip-range
      edit 1
        set start-ip 2001:18:56:66::1000
        set end-ip 2001:18:56:66::1100
      next
    end
  next
end
```

Improve CAPWAP stability

This release improves CAPWAP session stability between the wireless controller and FortiAP, particularly in cloud environments, by providing more control over keep-alive message intervals and maximum retransmit intervals. The controller now includes the WTP index, keep-alive interval, and maximum retransmit interval in the JOIN_RESP message sent to the FortiAP.

The WTP index acts as a unique identifier for the FortiAP session. The FortiAP embeds the least significant 24 bits of its WTP index into the reserved field of the CAPWAP DTLS header. Controllers, such as FortiEdge Cloud, use this embedded value to associate incoming encrypted DTLS packets with the correct session. This mechanism allows the controller to maintain session continuity, even if the AP's data port changes due to NAT or intermediate network fluctuations.

Configuring the keep-alive interval to be shorter than the echo interval enables quicker detection and recovery from connectivity issues. The keep-alive message also includes the session ID, enabling the controller to accurately rebind the data and control channels if the AP's data port changes.

These enhancements improve AP recovery behavior in cloud-managed environments, where echo intervals are typically longer by design.

The following CLI commands have been added:

```
config wireless-controller timers
  set keep-alive-interval <seconds>
  set max-retransmit-interval <seconds>
end
```

`keep-alive-interval`	Maximal time in seconds between control requests sent by the managed WTP, AP, or FortiAP (1 to 255 sec, default = 0).
`max-retransmit-interval`	Maximal time in seconds that a FortiAP will wait before retransmitting a CAPWAP control message (such as keep-alives or join requests) when a response is not received from the controller (1 to 255 sec, default = 0).

If you leave the timers at their default value of 0, the `keep-alive-interval` will equal the `echo-interval`, and the `max-retransmit-interval` will equal half the `echo-interval`.

To configure CAPWAP timer settings for improved CAPWAP session stability:

1. From the FortiGate, configure the `keep-alive-interval` and `max-retransmit-interval` settings:

```
config wireless-controller timers
  set keep-alive-interval 15
  set max-retransmit-interval 6
end
```

2. From the FortiGate, enable `wtpidx` debug to ensure the WTP index is pushed to the FortiAP and used as an identifier.

```
FortiGate-81F-POE(Interim)#diag wireless-controller wlac debug wtpidx 1
Module Name: wtpidx          (null)                          Debug Level 1
```

3. Connect the FortiAP to the FortiGate.
4. Authorize the FortiAP and wait for it to come online.

```
config wireless-controller wtp
  edit "FP431FTF23021146"
    set admin enable
  next
end
```

5. Check the WTP index that FortiGate pushed to the FortiAP.

```
FortiGate-81F-POE (Interim)# diagnose wireless-controller wlac -d wtp
vf=0 mpId=0 wtp=2 id=FP431FTF23021146 base=84:39:8f:89:8c:e8 0-99.99.99.3:48307(88)<->0-
99.99.99.1:5247 use=19
```

6. From the FortiAP, verify that the FortiAP receives the correct keep-alive-interval, max-retransmit-interval, and the WTP Index value from the FortiGate.

```
FortiAP-431F # cw_diag -c acs | grep -C 3 keep-aliv
    radio cnt            : 3
    sta info             : 0/0
    echo-interval        : 30
    keep-alive-interval  : 15
    nat-sess-keep-alive  : 0
    max-retransmit       : 3
    max-rexmit-interval  : 6
    dc-dead-interval     : 120
--
ACS 0 info
    wcha info            : mode=0 max=10 wait=10 peer_cnt=0
    acPri                : 1
    wtp-index            : 0x2(2) keep-alive 15
    fsm-state            : RUN 10 ac_discovery 1
    ac-ip-addr           : 99.99.99.1:5246,5247        MULTICAST
    ac-name              : FortiGate-81F-POE
--
ACS 1 info
    wcha info            : mode=0 max=0 wait=0 peer_cnt=0
```

```
    acPri               : 2
    wtp-index           : 0x0(0) keep-alive 0
    fsm-state           : START 33 ac_discovery 0
    ac-ip-addr          : 0.0.0.0:0,0        UNKNOWN
    ac-name             :
```

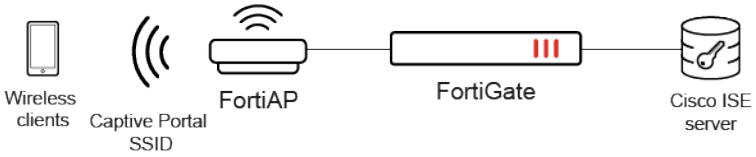
Support dynamic redirect URLs from Cisco ISE

FortiOS WiFi controller now supports dynamic redirect URLs from Cisco Identity Services Engine (ISE). When enabled, clients are redirected to a customized or session-specific portal pages for authentication without requiring manual URL input.

When a wireless client connects to a captive portal SSID integrated with a Cisco ISE server, the FortiOS wireless controller can redirect the client to a dynamic portal URL specified in the RADIUS Access-Accept message. This URL is provided through the "Fortinet-Captive-Portal-URL" RADIUS attribute when `captive-portal-dynamic-redirect-url` is enabled. This CLI option can be enabled in a tunnel mode captive portal VAP when the portal-type is set to `auth-mac`:

```
config wireless-controller vap
  edit <name>
    set captive-portal enable
    set portal-type auth-mac
    set captive-portal-dynamic-redirect-url [enable | disable]
  next
end
```

`captive-portal-dynamic-redirect-url`	Enable/disable captive portal dynamic redirect URL (default = disable).



To configure dynamic redirect URLs from Cisco ISE:

1. Configure the Cisco ISE server with a Fortinet Authorization Profile.

   Note: For information about configuring Cisco ISE, refer to the Cisco ISE documentation.

   a. In Cisco ISE, navigate to *Dictionaries > System > RADIUS > RADIUS Vendors > Fortinet* and add the RADIUS attribute *Fortinet-Captive-Portal-URL" (id: 27)*.

   b. Go to *Policy > Policy Elements >Authorization >Authorization Profiles* and create a *FTNT-Guest-Redirection* Authorization Profile.

   c. Under *Advanced Attributes Settings*, add the *Fortinet-Captive-Portal-URL* attribute with the value from the *Self-Registered Guest Portal (default)* setting.

   d. Go to *Policy Set > Default > Authorization Policy* and select the *FTNT-Guest-Redirection* Authorization Profile you configured.

e. When you are finished, click *Save*.

2. Configure the FortiOS to use the Cisco ISE server.

a. Create a RADIUS server to the remote Cisco ISE server:

```
config user radius
  edit "ISE-152"
    set server "172.30.60.152"
    set secret ENC
    set radius-coa enable
  next
end
```

b. Create two user groups: one for unauthorized users, and one for authorized users.

```
config user group
  edit "unregistered"
    set member "ISE-152"
    config match
      edit 1
        set server-name "ISE-152"
        set group-name "unregistered"
      next
    end
  next
  edit "AuthorizedGuest"
    set member "ISE-152"
    config match
      edit 1
        set server-name "ISE-152"
        set group-name "AuthorizedGuest"
      next
    end
  next
end
```

c. Create a tunnel-mode captive portal VAP using the Cisco ISE address, and enable `captive-portal-dynamic-redirect-url`.

```
config wireless-controller vap
  edit "ISE-CWP"
    set ssid "ISE-CWP"
    set security wpa-personal
    set external-web "https://172.30.60.152:8443/portal/g?p=XO8mXJ9WqmpMWFQqEKko4mX18q"
    set radius-mac-auth enable
    set radius-mac-auth-server "ISE-152"
    set radius-mac-auth-usergroups "AuthorizedGuest"
    set passphrase *
    set captive-portal enable
    set portal-type auth-mac
    set captive-portal-dynamic-redirect-url enable
    set schedule "always"
```

```
 next
end
```

d. Create two firewall policies to ensure the connected wireless clients can access the portal server and the Internet if they pass authentication.

```
config firewall policy
  edit 11
    set name "unregistered Guests to ISE Portal"
    set uuid 582440b6-4ddb-51ee-4674-92d87a9c84ce
    set srcintf "ISE-CWP"
    set dstintf "port1"
    set action accept
    set srcaddr "all"
    set dstaddr "ISE-152"
    set schedule "always"
    set service "ALL"
    set nat enable
    set timeout-send-rst enable
    set captive-portal-exempt enable
  next
  edit 12
    set name "AuthorizedGuest users to internet"
    set uuid b499c910-4ddb-51ee-b1c0-a036a863b8d8
    set srcintf "ISE-CWP"
    set dstintf "port1"
    set action accept
    set srcaddr "all"
    set dstaddr "all"
    set schedule "always"
    set service "ALL"
    set nat enable
    set groups "AuthorizedGuest"
    set timeout-send-rst enable
  next
end
```

To verify Cisco ISE captive portal setup:

1. Connect a wireless device to the SSID, and then verify that the client can pass RADIUS-MAC-Authentication first and are assigned to the *unregistered* user group.

```
# diagnose wireless-controller wlac -d sta online
 ? vf=0 mpId=10 wtp=1 rId=2 wlan=ISE-CWP vlan_id=0 ip=10.6.131.2 ip6=fe80::1c20:6e17:fcae:18e6
mac=18:65:90:cb:de:5b vci= host=Offload-MAC-2 user=18-65-90-CB-DE-5B group=unregistered
signal=-53 noise=-95 idle=126 bw=0 use=5 chan=36 radio_type=11AC_5G security=wpa_wpa2_
personal+captive mpsk= encrypt=aes cp_authed=no l3r=1,0 G=0.0.0.0:0,0.0.1.0:0-0-0 -- 0.0.0.0:0
0,0 online=yes mimo=3
               ip6=*fe80::1c20:6e17:fcae:18e6,21,
                        rad_mac_auth=allow  age=268
```

The wireless client is directed to the Cisco ISE captive portal page.

2. After the client logs in with the correct credentials, they are assigned to the *AuthorizedGuest* group and can pass traffic to the Internet.

```
# diagnose wireless-controller wlac -d sta online
  vf=0 mpId=10 wtp=1 rId=2 wlan=ISE-CWP vlan_id=0 ip=10.6.131.2 ip6=fe80::1c20:6e17:fcae:18e6
mac=18:65:90:cb:de:5b vci= host=Offload-MAC-2 user=18-65-90-CB-DE-5B group= AuthorizedGuest
signal=-52 noise=-95 idle=18 bw=0 use=5 chan=36 radio_type=11AC_5G security=wpa_wpa2_
personal+captive mpsk= encrypt=aes cp_authed=yes l3r=1,0 G=0.0.0.0:0,0.0.0.0:0-0-0 --
0.0.0.0:0 0,0 online=yes mimo=3
                ip6=*fe80::1c20:6e17:fcae:18e6,22,
                         rad_mac_auth=allow  age=3322

# diagnose firewall auth list
10.6.131.2, 18-65-90-CB-DE-5B
        src_mac: 18:65:90:cb:de:5b
        type: wsso, id: 0, duration: 89, idled: 71
        expire: 229, allow-idle: 300
        flag(10): radius
        server: ISE-152
        packets: in 191 out 187, bytes: in 79750 out 47445
        group_id: 3
        group_name: AuthorizedGuest

----- 1 listed, 0 filtered ------
```

Support IPsec traffic offloading from the FortiAP

This release supports IPsec traffic offloading from the FortiAP. When the FortiAP profile's DTLS policy is set to `ipsec-vpn` or `ipsec-vpn-sn` in the CLI, you can choose to enable or disable IPsec traffic offloading using the new `ipsec-offload` CLI option.

In secure branch office deployments, FortiAPs are connected to a remote FortiGate. To ensure compliance and security, all data inside the CAPWAP must be encrypted between the FortiAP and the FortiGate. When IPsec offloading is enabled on FortiAP, the FortiAP can offload traffic by itself, improving the client throughput.

The following CLI commands have been added:

```
config wireless-controller wtp-profile
  edit <name>
    set dtls-policy ipsec-vpn
    set ipsec-offload [enable | disable]
  next
end
```

ipsec-offload	Enable/disable data channel IPsec offloading (default = enable). This option is available when `dtls-policy` is set to `ipsec-vpn` or `ipsec-sn-vpn`.

To configure IPsec offloading:

1. Configure a FortiAP profile to use an `ipsec-vpn` or `ipsec-sn-vpn` data channel, then enable `ipsec-offload`.

```
config wireless-controller wtp-profile
  edit "23JF"
    set dtls-policy ipsec-vpn
    set ipsec-offload enable
  next
end
```

2. Apply the FortiAP profile to a FortiAP.

```
config wireless-controller wtp
  edit "FP23JFTF23006371"
    set wtp-profile "23JF"
  next
end
```

3. From the FortiGate, verify that the configurations have been applied.

```
# diagnose wireless-controller wlac -c wtp FP23JFTF23006371
------------------------------WTP    1----------------------------
WTP vd               : vdom1, 1-FP23JFTF23006371    MP00
    uuid             : 7a03d876-8cfa-51f0-4a7a-8127729c2294
    mgmt_vlanid      : 0
    region code      : A  valid
    refcnt           : 3 own(1) wtpprof(1) ws(1)   deleted(no)
    apcfg status     : N/A,N/A cfg_ac=0.0.0.0:0 val_ac=0.0.0.0:0 cmds T 0 P 0 U 0 I 0 M 0
    apcfg cmd details:
    plain_ctl        : disabled
    image-dl(wtp,rst): yes,no
    admin            : enable
    wtp-profile      : cfg(23JF) override(disabled) oper(23JF)
    … …
  Capability        :
      local standalone            : enabled
      lan port                    : enabled
      local switch                : enabled
      vlan                        : enabled
      local bridge                : enabled
      DFS                         : enabled
      timestamp offset            : enabled
      txpower percentage          : enabled
      wpa3                        : enabled
      station health              : enabled
      DTLS v1.2                   : enabled
      multiple time schedule      : enabled
      energy-efficient-ethernet   : enabled
      wan lan mode                : disabled
      led dark                    : enabled
      kernel DTLS data            : enabled
```

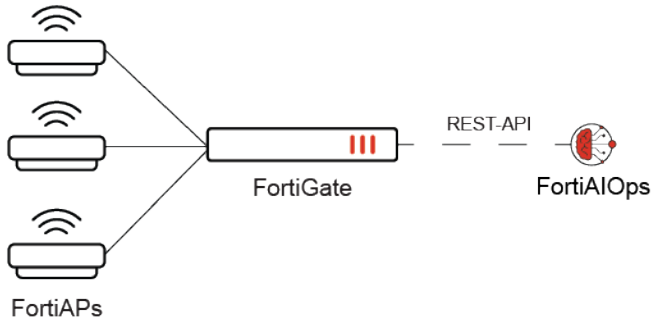
```
     ipsec offload                : enabled
… …
```

4. From the FortiAP, verify that the configurations have been pushed from the FortiGate.

```
FortiAP-23JF # wcfg
WTP Configuration
    name                 : FortiAP-23JF
    ...
    data-chan-sec-cfg    : clear-text dtls ipsec-sn ipsec
    data-chan-sec-oper   : ipsec
    ipsec-offload        : enabled(cfg)/enabled(oper)
    ...
```

Enhance DARRP with FortiAIOps

This release enhances Distributed Radio Resource Provisioning (DARRP) channel selection on FortiGates managed by FortiAIOps. FortiAIOps collects radio and channel data from FortiGate via REST APIs, and recommends optimal channels to reduce interference. This enables smarter, centralized Wi-Fi tuning in high-density environments.



A new command has been added under `radio configurations in the wtp-profile`

```
config wireless-controller wtp-profile
  edit <name>
    config radio-1
      set ai-darrp-support [enable | disable]
    next
  next
end
```

`ai-darrp-support`	Enable/disable support for FortiAIOps REST API calls for DARRP data (default = disabled).

The new `ai-darrp-support` command can be combined with the old `darrp` command to trigger the following behaviors:

darrp	ai-darrp-support	Description
disable	disable	The FortiGate local DARRP is disabled and FortiAIOps DARRP is disabled.
enable	disable	The FortiGate local DARRP is enabled and FortiAIOps DARRP is disabled.
disable	enable	The FortiGate local DARRP is disabled and FortiAIOps can only collect data through GET REST-APIs.
enable	enable	The FortiGate lets the remote FortiAIOps run the DARRP algorithm instead of running it locally. FortiAIOps can collect data through GET REST-APIs and trigger FortiAP channel changes using POST REST-APIs.

To configure FortiAIOps DARRP on a FortiAP profile:

1. Enable `ai-darrp-support` in the radio configuration of a FortiAP profile:

```
config wireless-controller wtp-profile
  edit "FAP231G-default"
    config radio-1
      set band 802.11g 802.11n-2G 802.11ax-2G
      set ai-darrp-support enable
      set darrp enable
      set arrp-profile "arrp-default"
      end
    config radio-2
      set band 802.11a 802.11n-5G 802.11ac-5G 802.11ax-5G
      set channel-bonding 40MHz
      set ai-darrp-support enable
      set darrp enable
      set arrp-profile "arrp-default"
      end
    config radio-3
      set band 802.11ax-6G
      set channel-bonding 160MHz
      set ai-darrp-support enable
      set darrp enable
      set arrp-profile "arrp-default"
      end
  next
end
```

2. Once enabled, FortiAPOps collects radio and channel information from the FortiGate through a GET REST-API. Example API response from all APs on all radios:
`https://172.18.56.47/api/v2/monitor/wifi/darrp?val=3`


```
{"http_method":"GET","results":
[{"wtp_id":"FP231GTF24061397","radio_id":1,"bw":"20","base_
mac":"04:01:a1:18:b3:80","index":0,"chan_cnt":3,"vfid":0,"band":"2.4G","new_chan":6,"opr_
chan":6,"age":19608,"darrp":true,"ai_support":true,"channel":
[{"chan":1,"chan_allow":1,"rssi_score_w":0,"rssi_score_o":0,"rssi_score_bcn_w":0,"rssi_score_
```

```
bcn_o":0,"rssi_norm":0,"nf":0,"chan_load":100,"spectral_rssi":0,"is_weather":0,"is_
dfs":0,"chan_score":2000,"flag0":0,"flag1":16},

{"wtp_id":"FP241KTF25005382","radio_id":1,"bw":"20","base_
mac":"b4:b2:e9:53:d8:50","index":1,"chan_cnt":3,"vfid":0,"band":"2.4G","new_chan":1,"opr_
chan":1,"age":402,"darrp":true,"ai_support":true,"channel":
[{"chan":1,"chan_allow":1,"rssi_score_w":0,"rssi_score_o":289,"rssi_score_bcn_w":0,"rssi_
score_bcn_o":4,"rssi_norm":72,"nf":9,"chan_load":58,"spectral_rssi":32,"is_weather":0,"is_
dfs":0,"chan_score":3520,"flag0":0,"flag1":40},

……


}],"vdom":"root","path":"wifi","name":"darrp","status":"success","serial":"FGVM08TM25003066","
version":"v7.6.5","build":3619}
```


3. FortiAIOps then adjusts the DARRP algorithm according to the collected data. In then sends the recommended channels to the FortiGate through a POST REST-API.

   Example POST API to a FortiGate:

   `https://172.18.56.47/api/v2/monitor/wifi/darrp/apply-value`

```
{
    "radio_entry": [
        {
            "wtp_id": "FP231GTF24061397",
            "radio_id": 1,
            "chan": 6
        },
        {
            "wtp_id": "FP231GTF24061397",
            "radio_id": 2,
            "chan": 44
        },
        {
            "wtp_id": "FP231GTF24061397",
            "radio_id": 3,
            "chan": 121
        }
    ]
}
```

4. The FortiGate receives the recommended channels from FortiAIOps and applies the configurations to the FortiAP.

```
# diagnose wireless-controller wlac -c wtp FP231GTF24061397
WTP vd               : root, 0-FP231GTF24061397    MP00

    admin            : enable
    wtp-profile      : cfg(FAP231G-default) override(disabled) oper(FAP231G-default)
    active sw ver    : FP231G-v7.6.2-build0972
    local IP addr    : 10.20.20.2
    connection state : Connected (0-10.20.20.2:5246)
```

```
  …
 Radio 1            : AP
   …
   oper chan        : 6 (AI 6)
   …

 Radio 2            : AP
   …
   oper chan        : 44 (AI 44)

Radio 3            : AP
   …
   oper chan        : 121 (AI 121)
```

5. The channel changes triggered by FortiAIOps are logged in the FortiGate wireless event logs.

```
1: date=2025-10-03 time=14:04:06 eventtime=1759525446461433323 tz="-0700" logid="0104043727"
type="event" subtype="wireless" level="notice" vd="root" logdesc="Physical AP radio DARRP
channel change from AI" sn="FP231GTF24061397" ap="FP231GTF24061397" ip=10.20.20.2 radioid=3
ssid="N/A" trafficmode="N/A" channel=121 radioband="802.11ax (6G)" bandwidth="160MHz"
configcountry="US " opercountry="US" cfgtxpower=30 opertxpower=20 slctdrmamode="ap"
operdrmamode="ap" action="ai-darrp-channel" msg="AP FP231GTF24061397 radio 3 channel 189 ==>
121 from AI DARRP."

2: date=2025-10-03 time=14:04:06 eventtime=1759525446461321551 tz="-0700" logid="0104043727"
type="event" subtype="wireless" level="notice" vd="root" logdesc="Physical AP radio DARRP
channel change from AI" sn="FP231GTF24061397" ap="FP231GTF24061397" ip=10.20.20.2 radioid=2
ssid="N/A" trafficmode="N/A" channel=44 radioband="802.11ax (5G)" bandwidth="40MHz"
configcountry="US " opercountry="US " cfgtxpower=17 opertxpower=28 slctdrmamode="ap"
operdrmamode="ap" action="ai-darrp-channel" msg="AP FP231GTF24061397 radio 2 channel 157 ==>
44 from AI DARRP."

3: date=2025-10-03 time=14:04:06 eventtime=1759525446461187958 tz="-0700" logid="0104043727"
type="event" subtype="wireless" level="notice" vd="root" logdesc="Physical AP radio DARRP
channel change from AI" sn="FP231GTF24061397" ap="FP231GTF24061397" ip=10.20.20.2 radioid=1
ssid="N/A" trafficmode="N/A" channel=6 radioband="802.11ax,n,g-only (2G)" bandwidth="20MHz"
configcountry="US " opercountry="US " cfgtxpower=27 opertxpower=26 slctdrmamode="ap"
operdrmamode="ap" action="ai-darrp-channel" msg="AP FP231GTF24061397 radio 1 channel 1 ==> 6
from AI DARRP."
```

Support up to 16 VAPs on FortiAP G and K-series models

In this release, select FortiAP G and K-series models can support up to 16 Virtual Access Points (VAPs) per radio. Depending on the model type, you can choose between 8 and 16 as the maximum number of supported SSIDs.

A new CLI command has been added to configure the maximum number of supported VAPs per radio:

```
config wireless-controller global
  set max-vap-per-radio [8 | 16]
end
```

max-vap-per-radio	Maximum number of SSIDs supported on the radio (default = 8). • 8: Support at most 8 SSIDs on the radio. • 16: Support at most 16 SSIDs on the radio.

FAP-221K, FAP-231K, FAP-23JK, FAP-222KL, and FAP-231KD only support 8 VAPS per radio.

To configure the number of supported VAPs on a radio:

1. Under the wireless controller global settings, set `max-vap-per-radio` to 16.

```
config wireless-controller global
  set max-vap-per-radio 16
end
```

2. Configure a FortiAP profile and add up to 16 VAPs.

```
config wireless-controller wtp-profile
  edit "FAP231G-default"
    config radio-2
      set vap-all manual
      set vaps "wifi.br.01" "wifi.br.02" "wifi.br.03" "wifi.br.04" "wifi.br.05" "wifi.br.06"
"wifi.br.07" "wifi.br.08" "wifi.fap.01" "wifi.fap.02" "wifi.fap.03" "wifi.fap.04"
"wifi.fap.05" "wifi.fap.06" "wifi.fap.07" "wifi.fap.08"
    next
  end
end
```

3. Apply the FortiAP profile to a FortiAP.

```
config wireless-controller wtp
  edit "FP231GTF23090954"
    set wtp-profile "FAP231G-default"
  next
end
```

4. Verify that the FortiAP is successfully broadcasting the SSIDs.

```
# diag wireless-controller wlac -c ws
-------------------------------WTP SESSION    1----------------------------
WTP session             : 0-10.6.30.254:25246  MP00   CWAS_RUN,528 1,1
......
  Radio 2            : AP
    wlan on             : enabled
    wlan cfg            : wifi.br.01  wifi.br.02  wifi.br.03  wifi.br.04  wifi.br.05
 wifi.br.06  wifi.br.07  wifi.br.08  wifi.fap.01  wifi.fap.02  wifi.fap.03  wifi.fap.04
 wifi.fap.05  wifi.fap.06  wifi.fap.07  wifi.fap.08
      vap-00(1)         : wifi.br.01       38:c0:ea:1c:67:18   lsw lbr          wifi.br.01
  Config success State DFS_WAIT(6) Age 36
      vap-01(2)         : wifi.br.02       38:c0:ea:1c:67:19   lsw lbr          wifi.br.02
  Config success State DFS_WAIT(6) Age 35
      vap-02(3)         : wifi.br.03       38:c0:ea:1c:67:1a   lsw lbr          wifi.br.03
```

```
  Config success State DFS_WAIT(6) Age 34
      vap-03(4)         : wifi.br.04       38:c0:ea:1c:67:1b   lsw lbr          wifi.br.04
  Config success State DFS_WAIT(6) Age 33
      vap-04(5)         : wifi.br.05       38:c0:ea:1c:67:1c   lsw lbr          wifi.br.05
  Config success State DFS_WAIT(6) Age 33
      vap-05(6)         : wifi.br.06       38:c0:ea:1c:67:1d   lsw lbr          wifi.br.06
  Config success State DFS_WAIT(6) Age 32
      vap-06(7)         : wifi.br.07       38:c0:ea:1c:67:1e   lsw lbr          wifi.br.07
  Config success State DFS_WAIT(6) Age 31
      vap-07(8)         : wifi.br.08       38:c0:ea:1c:67:1f   lsw lbr          wifi.br.08
  Config success State DFS_WAIT(6) Age 31
      vap-08(9)         : wifi.fap.01      3a:c0:ea:1c:67:18   lsw              wifi.fap.01
  Config success State DFS_WAIT(6) Age 30
      vap-09(10)         : wifi.fap.02      3a:c0:ea:1c:67:19   lsw              wifi.fap.02
   Config success State DFS_WAIT(6) Age 30
      vap-10(11)         : wifi.fap.03      3a:c0:ea:1c:67:1a   lsw              wifi.fap.03
   Config success State DFS_WAIT(6) Age 29
      vap-11(12)         : wifi.fap.04      3a:c0:ea:1c:67:1b   lsw              wifi.fap.04
   Config success State DFS_WAIT(6) Age 29
      vap-12(13)         : wifi.fap.05      3a:c0:ea:1c:67:1c   lsw              wifi.fap.05
   Config success State DFS_WAIT(6) Age 29
      vap-13(14)         : wifi.fap.06      3a:c0:ea:1c:67:1d   lsw              wifi.fap.06
   Config success State DFS_WAIT(6) Age 29
      vap-14(15)         : wifi.fap.07      3a:c0:ea:1c:67:1e   lsw              wifi.fap.07
   Config success State DFS_WAIT(6) Age 29
      vap-15(16)         : wifi.fap.08      3a:c0:ea:1c:67:1f   lsw              wifi.fap.08
   Config success State DFS_WAIT(6) Age 28
......
-------------------------------Total    1 WTP SESSIONs----------------------------
```

Support Wi-Fi 7 MLO on FortiAP K-series models

Support for enabling Multi-Link Operation (MLO) on FortiAP K-series models using the CLI was added in FortiOS 7.6.5. FortiOS 8.0.0 adds GUI support.

FortiAP K-series models running FAP 7.6.3 and later support Multi-Link Operation (MLO), a feature of 802.11be (Wi-Fi 7). MLO enables devices to simultaneously transmit and receive data across multiple frequency bands (2.4 GHz, 5 GHz, and 6 GHz) and channels, enabling higher data rates, reduced latency, and improved reliability.

The following table shows which FortiAP model and radio support MLO.

FortiAP Model	2.4 GHz radio	5 GHz radio	6 GHz radio
FAP-231K	√	√	√
FAP-23JK	√	√	√
FAP-241K	√	√	√
FAP-243K	√	√	√

FortiAP Model	2.4 GHz radio	5 GHz radio	6 GHz radio
FAP-441K*	N/A	√	√
FAP-443K*	N/A	√	√

*On FAP-441K and FAP-443K, the 2.4 GHz radio does not support Wi-Fi 7 and therefore does not support MLO.

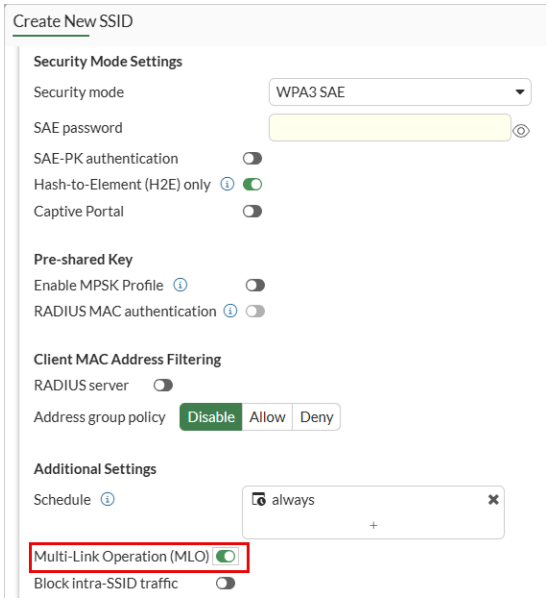
A new CLI command has been added to enable MLO:

```
config wireless-controller vap
  edit <name>
    set security [wpa3-sae | wpa3-enterprise | wpa3-only-enterprise | owe]
    set mlo [enable | disable]
  next
end
```

mlo	Enable/disable Wi-Fi 7 Multi-Link-Operation (default = disable). **Note:** MLO can only be configured if `security` is set to `wpa3-sae`, `wpa3-enterprise`, `wpa3-only-enterprise`, or `owe`.

To enable MLO - GUI:

1. From the FortiGate, go to *WiFi & Switch Controller > SSIDs* and create or edit an SSID.
2. Under *Security Mode Settings*, set the *Security mode* to either*WPA3-SAE*, or *WPA3 Enterprise Only*.
   If you selected *WPA3-SAE*, enable *Hash-to-Element (H2E) only*.
3. Under *Additional Settings*, enable *Multi-Link-Operation (MLO)*.



4. When you are finished, click *OK*.
5. Apply the SSID to a FortiAP Profile under the radios you want to broadcast from.
6. Verify that you can connect an MLO-capable device to the SSID.

To enable MLO - CLI:

1. Configure a VAP with `security` set to `wpa3-sae`, `wpa3-enterprise`, `wpa3-only-enterprise`, or `owe`, and then enable `mlo`.

```
config wireless-controller vap
  edit "VAP1"
    set ssid "VAP1"
    set security wpa3-sae
    set sae-h2e-only enable
    set mlo enable
  next
end
```

   If `security` is set to `wpa3-sae`, `sae-h2e-only` is automatically enabled.
2. Apply the VAP you created to a WTP profile under the radios you want to broadcast from.

```
config wireless-controller wtp-profile
  edit "FAP241K-default"
    config radio-1
      set band 802.11g 802.11n-2G 802.11ax-2G 802.11be-2G
      set vap-all manual
      set vaps "VAP1"
    end
    config radio-2
      set band 802.11be-5G
      set channel-bonding 80MHz
      set power-mode dBm
      set power-value 33
      set vap-all manual
      set vaps "VAP1"
      set channel "149" "153" "157" "161"
    end
    config radio-3
      set band 802.11ax-6G 802.11be-6G
      set vap-all manual
      set vaps "VAP1"
    end
    config radio-4
      set mode monitor
    end
  next
end
```

3. Verify that you can connect a MLO-capable device to the SSID.

To verify that MLO is working:

1. From the FortiGate, check the MLO status.

```
FortiGate-80F-POE (root) (Interim)# diagnose wpa wpad mld
[0] 0-134.134.134.3:5246-VAP1 04:01:a1:5f:d8:00, num_links: 3, ref: 3, used_IDs: 0007
```

```
HOSTAPD: 0-134.134.134.3:5246<1-0>  link_id: 0, 04:01:a1:5f:d8:00
HOSTAPD: 0-134.134.134.3:5246<0-0>  link_id: 1, 04:01:a1:5f:d7:f8
HOSTAPD: 0-134.134.134.3:5246<2-0>  link_id: 2, 04:01:a1:5f:d8:08
```

Check the MLA-capable SSID and the number of linked radios used.

2. From the FortiAP, confirm that `mlo` is enabled on the VAP.

```
FortiAP-241K # vcfg | grep mlo
    okc=disabled, dynamic_vlan=disabled, extern_roaming=disabled, mlo=enabled
    okc=disabled, dynamic_vlan=disabled, extern_roaming=disabled, mlo=enabled
    okc=disabled, dynamic_vlan=disabled, extern_roaming=disabled, mlo=enabled
```

3. From the FortiAP, check that client entries can be detected across multiple radios.

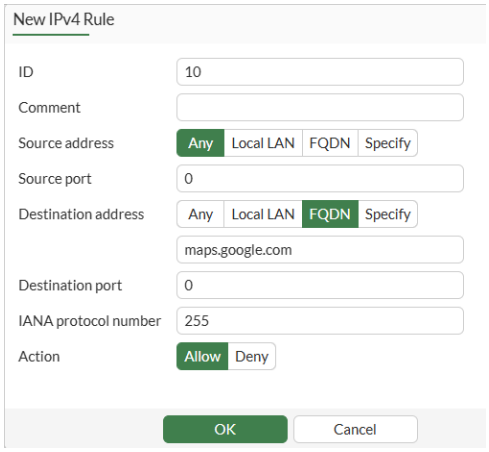
```
FortiAP-241K # sta

wlan00 (VAP1) client count 1
    MAC:d2:e1:a0:c6:1e:63 ip:0.0.0.0 ip_proto:none ip_age:6666 host: vci:
        vlanid:0 Auth:Yes l3r:0 channel:1 rate:6Mbps rssi:0dB idle:6666s
        Rx bytes:0 Tx bytes:0 Rx rate:0Mbps Tx rate:0Mbps Rx mcs:0 Tx mcs:0 Rx last:6666s Tx
last:6666s
        AssocID:89 Mode:  Normal Flags:400000f PauseCnt:0
        KEY type=aes_ccm pad=0 keyix=0 keylen=16 flags=3(xmit recv) RSC=0 TSC=0
            4f 69 17 fb   fd 8e 1f 00   40 93 87 b9   04 9c 3a 1a
            00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00
            00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00
        KEY type=aes_ccm pad=0 keyix=1 keylen=16 flags=7(xmit recv) RSC=0 TSC=0
            45 45 b8 bf   7f 9e 82 0f   88 d1 5f 98   d5 f6 ea 63
            00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00
            00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00

wlan10 (VAP1) client count 1
    MAC:56:81:6e:bf:6d:5b ip:134.134.134.4 ip_proto:dhcp ip_age:5802 host: vci:
        vlanid:0 Auth:Yes l3r:0 channel:149 rate:6Mbps rssi:41dB idle:42s
        Rx bytes:315674 Tx bytes:2277910 Rx rate:408Mbps Tx rate:6Mbps Rx mcs:4 Tx mcs:3 Rx
last:42s Tx last:42s
        AssocID:89 Mode:  Normal Flags:1400000b PauseCnt:0
        KEY type=aes_ccm pad=0 keyix=0 keylen=16 flags=3(xmit recv) RSC=0 TSC=3
            4f 69 17 fb   fd 8e 1f 00   40 93 87 b9   04 9c 3a 1a
            00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00
            00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00
        KEY type=aes_ccm pad=0 keyix=1 keylen=16 flags=7(xmit recv) RSC=0 TSC=0
            31 87 04 f3   d1 cc d7 b0   d1 9b b4 7c   72 5d 2a d0
            00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00
            00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00
```

Support FQDN in Layer 3 ACLs

Layer 3 Access Control Lists (ACLs) in FortiAP now support Fully Qualified Domain Names (FQDNs) and simple wildcard matching for both source and destination addresses.

To apply a FQDN address to an ACL - GUI:

1. Go to *System > Feature Visibility* and enable *Advanced Wireless Features* from System.
2. Click *Apply*.
3. Go to *WiFi & Switch Controller > Protection Profiles > L3 Firewall Profile* and edit or create an L3 Firewall Profile.
4. Create or edit an IPv4 or IPv6 rule.
5. In the *Source address* and *Destination address*, select *FQDN* as a the address type and enter the FQDN address.



6. When you are finished, click *OK*.

To apply a FQDN address to an ACL - CLI:

1. Create L3 firewall rules. In this example, the firewall is configured to allow all traffic to the destination address of `maps.google.com` while blocking all traffic to the destination address of `*google.com`.

```
config wireless-controller access-control-list
  edit "fqdn_test"
    config layer3-ipv4-rules
      edit 10
        set dstaddr maps.google.com
        set action allow
      next
      edit 20
        set dstaddr *.google.com
        set action deny
      next
    end
  next
end
```

2. Apply the access control list to an SSID.

```
config wireless-controller vap
  edit "wifi.acl.01"
    set ssid "wifi.acl.01"
    set passphrase ******
```

```
    set schedule "always"
    set access-control-list "fqdn_test"
  next
end
```

3. Verify that the rules have been applied onto the FortiGate.

```
# diagnose wireless-controller wlac -c afwprof
AFWPROF (001/001) vdom,name: root, fqdn_test
    refcnt           : 2 own(1) wlan(1)
    deleted          : no
    Layer3 ipv4 rule : 2
        ------------------------------------------------------------------------
         ##### Policy Prot Source ==> Destination
        ------------------------------------------------------------------------
            10 allow  any  any ==> maps.google.com
            20 deny   any  any ==> *.google.com
        ------------------------------------------------------------------------
    wlan cnt         : 1
        vap 001 : 0    wifi.acl.01
```

4. Verify that the rules have been pushed to the FortiAP.

```
FortiAP-441K # cw_diag -c afw-rules
Interface wlan10 firewall rules:
================================================================================
 RuleID HitCounter Policy Prot (IPv4)Source ==> Destination
------- ---------- ------ ---- ----------------------------------------------
     10          0 allow  any  any ==> maps.google.com
     20          0 deny   any  any ==> *.google.com
================================================================================
```

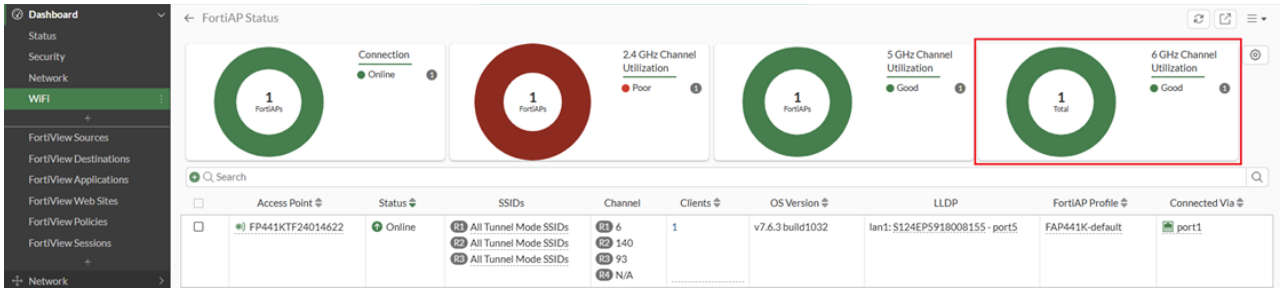
Add GUI support for 6 GHz Channel Utilization

This release adds a 6 GHz Channel Utilization chart for FortiAP G and K-series models in the FortiOS GUI. The charts can be accessed in the *Managed FortiAPs* page and in the *Dashboard* > *WiFi* > *FortiAP Status* widget. Users can also customize the layout of the channel utilization charts.

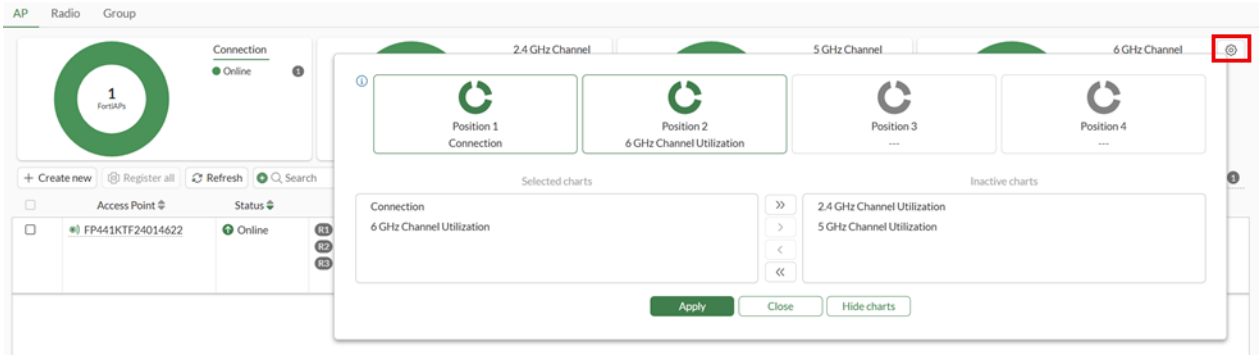
The 6 GHz Channel Utilization chart is available in the following locations:

- On *WiFi & Switch Controller* > *Managed FortiAPs*:

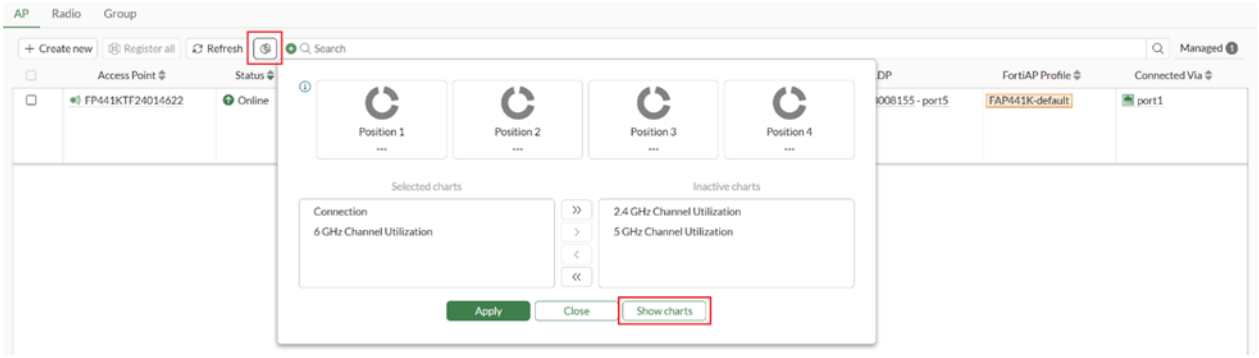
- On the *Dashboard* > *WiFi* > *FortiAP Status* widget:



You can customize and reorganize the Utilization Charts by clicking the *Settings* icon and selecting how you want to arrange the charts.



If you select *Hide charts*, the Utilization Charts are hidden and can be recovered by clicking the *Charts* button and then clicking *Show charts*.



Support customizable DHCP Option 82 configurations

This release adds support for customizable DHCP Option 82 configurations, enabling administrators to select any combination of sub-options and specify a delimiter in the VAP. In previous releases, DHCP82 only had three pre-defined styles that were not customizable, and only one of them could be selected in the VAP. In a network that needs to customize a DHCP for different devices, the administrator can enable Option 82 to only deliver IP addresses to known devices, with custom configurations based on the information sent by the device the client is connected to.

The FortiAP must be running firmware version 7.6.4 or later to support this feature.

CLI changes

A new command has been added to customize the delimiter for Circuit ID, and new options for `set dhcp-option82-circuit-id-insertion` and `set dhcp-option82-remote-id-insertion` are available:

```
config wireless-controller vap
    edit <name>
        set ssid <name>
        set dhcp-option82-insertion enable
        set dhcp-option82-delimiter <string>
        set dhcp-option82-circuit-id-insertion [ap-mac | ap-model | ap-hostname | ssid | ssid-type
| network-type | vlan | wtp-profile]
        set dhcp-option82-remote-id-insertion [client-mac]
    next
end
```

`dhcp-option82-delimiter`	If you want to use more than one Circuit ID, enter a delimiter to define the separate fields (Default = `;`).
`dhcp-option82-circuit-id-insertion`	Select DHCP option 82 Circuit IDs: • `ap-mac`: AP MAC • `ap-model`: AP Model • `ap-hostname`: AP Hostname • `ssid`: SSID • `ssid-type`: SSID Type • `network-type`: Network Type • `vlan`: VLAN ID • `wtp-profile`: WTP Profile Name.
`dhcp-option82-remote-id-insertion`	Select DHCP option 82 Remote ID: • `client-mac`: Client MAC.

After upgrading to this release, the FortiGate automatically converts any old commands to the following new configuration format:

Commands	Old configuration	New configuration
`dhcp-option82-circuit-id-insertion`	`style-1`	`ap-mac ssid ssid-type`
	`style-2`	`ap-mac`
	`style-3`	`network-type wtp-profile vlan ssid ap-model ap-hostname ap-mac`
`dhcp-option82-remote-id-insertion`	`style-1`	`client-mac`

Example configuration

To configure custom DHCP Option 82 configurations:

1. From the VAP, enable custom DHCP option 82 and select the circuit-id and remote-id. If there is more than one Circuit ID, configure the delimiter.

```
config wireless-controller vap
    edit "dhcp82"
        set ssid "dhcp82"
        set passphrase ENC
        set schedule "always"
        set dhcp-option82-insertion enable
        set dhcp-option82-delimiter "+"
        set dhcp-option82-circuit-id-insertion ap-mac ap-model ap-hostname wtp-profile
        set dhcp-option82-remote-id-insertion client-mac
    next
end
```

2. Apply the SSID to a FortiAP profile.

```
config wireless-controller wtp-profile
    edit "23JK"
        config radio-1
            set vap-all manual
            set vaps "dhcp82"
        end
        config radio-2
            set vap-all manual
            set vaps "dhcp82"
        end
    next
end
```

3. From the FortiAP, verify that the configurations have been applied:

```
FortiAP-23JK # vcfg
------------------------------VAP Configuration    1----------------------------
Radio Id  0 WLAN Id  0 dhcp82 ADMIN_UP(INTF_UP) init_done 0.0.0.0/0.0.0.0 unknown (-1)
          vlanid=0, intf=wlan00, vap=0xdc2d02c, bssid=48:3a:02:99:cb:f0
          11ax high-efficiency=enabled target-wake-time=enabled
          bss-color-partial=enabled
          mesh backhaul=disabled
          local_auth=disabled standalone=disabled nat_mode=disabled
          local_bridging=disabled split_tunnel=disabled layer3_roaming=disabled
          intra_ssid_priv=disabled lan_loc=enabled
          mcast_enhance=disabled igmp_snooping=disabled
          mac_auth=disabled fail_through_mode=disabled sta_info=0/0
          mac=local, tunnel=8023, cap=8ce0, qos=disabled
          prob_resp_suppress=disabled
          rx sop=disabled
          sticky client remove=disabled
```

```
            mu mimo=enabled           ldpc_config=rxtx
            dhcp_option43_insertion=enabled, dhcp_option82_insertion=enabled(version 1/1)
            dho82_circuit_id=ap-mac+ap-model+ap-hostname+wtp-profile
            dho82_remote_id=client-mac
            dhcp_enforcement=disabled
            access_control_list=disabled
            bc_suppression=dhcp dhcp-ucast arp
            auth=WPA2, PSK, AES WPA keyIdx=1, keyLen=16, keyStatus=1, gTsc=000000000000
            key=b1fc0607 e918996c 22eb39af b7000000
            pmf=disable
            okc=disabled, dynamic_vlan=disabled, extern_roaming=disabled, mlo=disabled
            80211k=enabled, 80211v=enabled, fast_bss_trans(802.11r)=disabled, mbo=disabled,
sae_h2e_only=disabled, sae_hnp_only=disabled, sae_pk=disabled, akm24_only=disabled
            neighbor_report_dual_band(802.11kv)=disabled
            port_macauth=disable
            airfairness weight: 20%
            schedules=SMTWTFS 00:00->00:00,
            ratelimit(Kbps): ul=0 dl=0 ul_user=0 dl_user=0 burst=disabled
            rates control configuration:
                rates-11ac-mcs-map: 11,11,11,11,11,11,11,11.
                rates-11ax-mcs-map: 11,11,11,11,11,11,11,11.
                rates-11be-mcs-map-20 : 4,4,4,4  4444
                rates-11be-mcs-map-160: 4,4,4,4  4444
                rates-11be-mcs-map-320: 4,4,4,4  4444
            UTM=disabled
            application detection engine: disabled
```

Optimize and update default DARRP parameters

This release removes the old default DARRP profile `arrp-default` and replaces it with two new DARRP profiles, `arrp-default-high-density` and `arrp-default-medium-density`. These new profiles are configured for high and medium density environments, enabling users to choose a profile that best fits their deployment environment. When you first enable DARRP under a radio, the ARRP profile is automatically set to `arrp-default-high-density`.

The maximum value for all weight settings has also been reduced to a more reasonable range of 200, and the default values for `weight-rogue-ap`, `weight-weather-channel`, and `weight-dfs-channel` have been adjusted to enable more realistic and effective channel assessment.

CLI Changes

There are two new DARRP profiles available:

```
config wireless-controller arrp-profile
  edit "arrp-default-high-density"
  next
  edit "arrp-default-medium-density"
    set threshold-noise-floor "-95"
    set threshold-channel-load 50
    set threshold-spectral-rssi "-75"
```

```
    set threshold-tx-retries 200
  next
end
```

The default values for the following options have been adjusted:

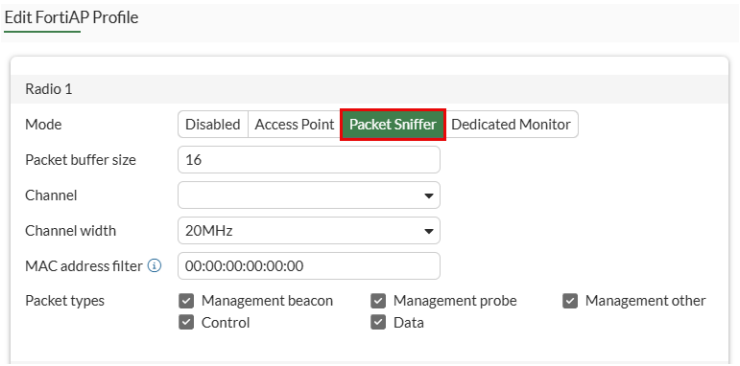
Option	New default
weight-rogue-ap	0
weight-weather-channel	100
weight-dfs-channel	50

Add GUI support for capturing wireless packets

This release supports using the FortiGate GUI to capture wireless packets from the FortiAP. You can configure a FortiAP radio to run in packet sniffer mode, configure the sniffer settings through the GUI, and then save the captured information as a pcap file to your PC.

To configure capturing wireless packets - GUI:

1. From the FortiOS GUI, go to *WiFi & Switch Controller* > *FortiAP Profiles* and edit the profile you want to configure packet sniffing on.
2. In the FortiAP profile, select a radio and set the *Mode* to *Packet Sniffer*.



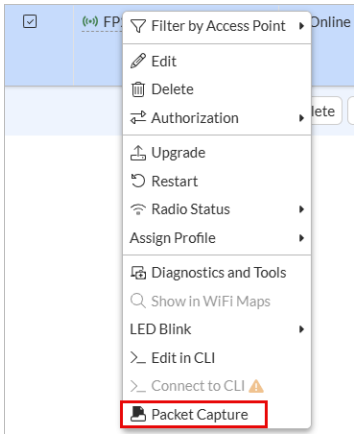
Only one radio can be set to Packet Sniffer mode.

3. Configure the following packet sniffing settings:

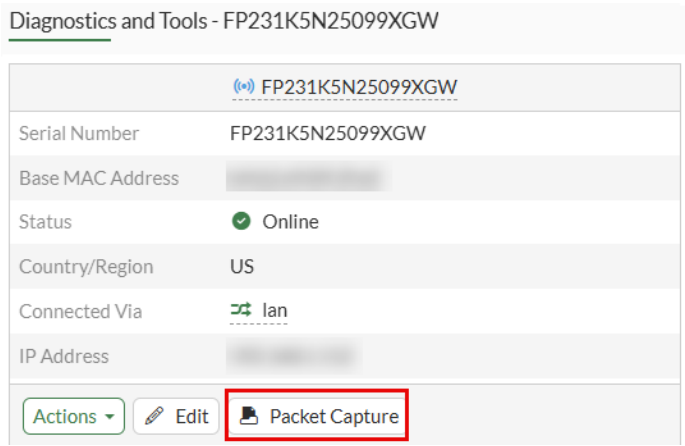
Packet buffer size	Packet file size limit.
Channel	Select a channel on which to operate the sniffer.
Channel width	Select a channel bandwidth for the sniffer.

MAC address filter	Enter a MAC address to monitor, or use MAC address *00:00:00:00:00:00* for no filtering.
Packet types	Select which packet types you want to capture: • *Management beacon*: Sniff Wi-Fi management Beacon frames. • *Management probe*: Sniff Wi-Fi management probe frames. • *Management other*: Sniff Wi-Fi management other frames. • *Control*: Sniff Wi-Fi control frames. • *Data*: Sniff Wi-Fi data frames.

4. When you are finished, click *OK*.
5. Once you configure the packet sniffing settings, you can access the Packet Capture window from the following locations:
   - Go to *WiFi & Switch Controller* > *Managed FortiAP*, right-click or hover over the FortiAP you want to capture packets on, and then select *Packet Capture*.

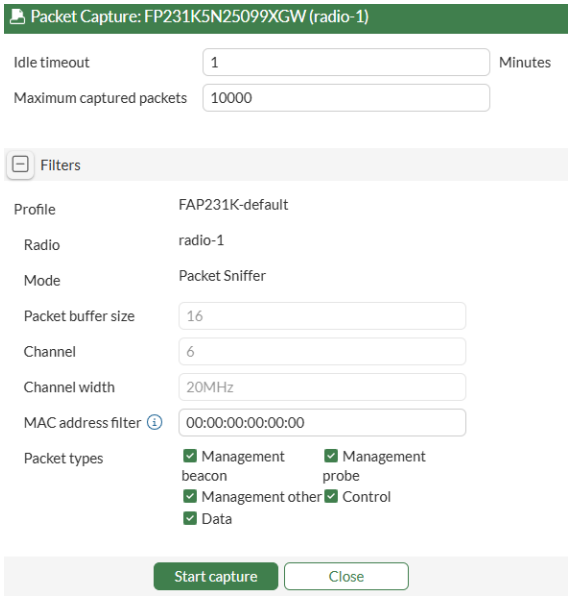


   - Go to *WiFi & Switch Controller* > *Managed FortiAP*, right-click or hover over the FortiAP you want to capture packets on, and then select *Diagnostics and Tools*. In the Diagnostics and Tools panel, click *Packet Capture*.



6. In the Packet Capture window, expand *Filters* and review the packet sniffing settings applied from the FortiAP profile.

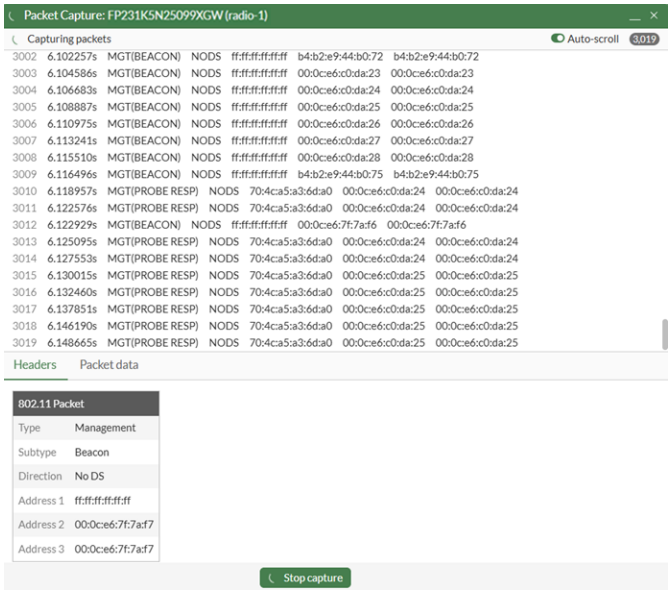
   You can edit the *MAC address filter* and *Packet types*. These changes will also be saved to the FortiAP profile.



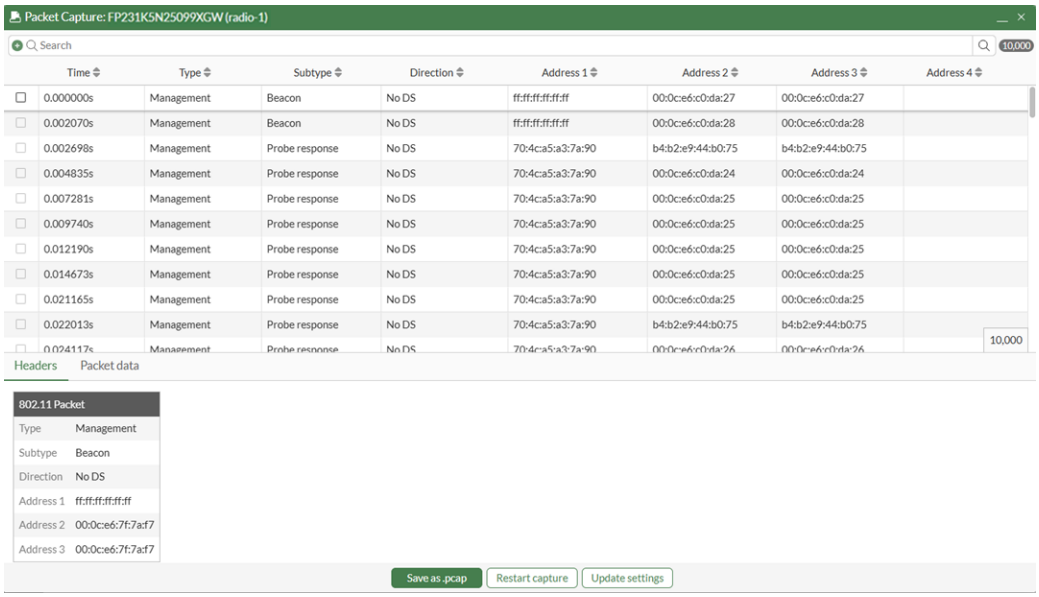
7. When you are ready, click *Start capture*.

   > Only one AP packet capture can be running at a time.

   The Packet Capture dialog loads and starts outputting captured packets in real time.



8. When you are finished, click *Stop capture* and then click *Save as .pcap*.



Packet capture data is not stored on the FortiGate. All packet data is streamed to the browser.

9. Save the pcap file to your PC and use a packet analyzer tool to open the file.

Support tuning Clear Channel Assessment threshold values

This release supports tuning Clear Channel Assessment (CCA) threshold values, enabling users to control how a radio responds to channel activity. A lower threshold causes the radio to detect weaker signals and defer more often, while a higher threshold causes it to ignore weaker signals and transmit more aggressively. In a high-density deployment (such as an office or campus environment with many nearby APs), radios may defer too frequently due to weak signals from neighboring cells. By adjusting the CCA threshold to a higher threshold value, you can reduce unnecessary deferrals, improve channel reuse, and increase overall throughput while maintaining acceptable collision levels.

CLI changes

Commands to configure the CCA value have been added under the WTP profile:

```
config wireless-controller wtp-profile
  edit <profile_name>
    config <radio>
      set cca-threshold <integer>
    end
  next
end
```

`cca-threshold`	Configure Clear Channel Assessment (CCA) threshold in dBm (-94 to -11, default = `0` for unconfigured).

The CCA threshold value can also be overridden on a per radio basis from each FortiAP device.

```
config wireless-controller wtp
  edit <wtp_name>
    config <radio>
      set override-cca-threshold [enable | disable]
      set cca-threshold <integer>
    end
  next
end
```

`override-cca-threshold`	Enable/disable overriding the WTP profile CCA threshold settings (default = `disable`).
`cca-threshold`	When `override-cca-threshold` is enabled, configure a Clear Channel Assessment (CCA) threshold in dBm (-94 to -11, default = `0` for unconfigured).

A lower (more negative) CCA threshold makes the radio more sensitive to neighboring signals and reduces collisions, but also causes more deferrals and lower throughput. You can tune the CCA threshold to a lower value in a relatively lower-density environment.

A higher (less negative) threshold reduces deferrals and increases throughput, but raises the risk of packet collisions and hidden-node interference. You can tune the CCA threshold to a higher value in a relatively high-density environment.

Setting the CCA threshold to the default value of `0` means it is un-configured and the FortiAP will use the predefined value in the Wi-Fi driver.

To configure CCA values on a FortiAP profile:

1. From a FortiAP profile, configure the `cca-threshold` values.

```
config wireless-controller wtp-profile
    edit "FAP431F-default"
        config radio-1
            set band 802.11g 802.11n-2G 802.11ax-2G
            set cca-threshold "-72"
        end
        config radio-2
            set band 802.11a 802.11n-5G 802.11ac-5G 802.11ax-5G
            set channel-bonding 40MHz
            set cca-threshold "-70"
        end
    next
end
```

2. Assign the FortiAP profile to a managed FortiAP.

```
config wireless-controller wtp
    edit "FP431FTF23099ZAZ"
        set admin enable
        set wtp-profile "FAP431F-default"
    next
end
```

3. From the FortiGate, verify that the CCA values have been assigned to a profile using the following diagnostic command:

```
FGVM08TM25003066 (Interim)# diagnose wireless-controller wlac -c wtp
------------------------------WTP    1----------------------------
WTP vd               : root, 0-FP431FTF23099ZAZ    MP00
    admin            : enable
    wtp-profile      : cfg(FAP431F-default) override(disabled) oper(FAP431F-default)
    wtp-mode         : normal
…
  Radio 1            : AP
    80211d enable    : enabled
    80211mc enable   : disabled
    country name     : US
    country code     : 841
    drma_manual_mode : ncf
    radio_type       : 11AX_2G
    channel list     : 1 6 11
    AI support       : disabled
    darrp            : disabled
    airtime fairness : disabled
    MIMO mode        : default
    bss color mode   : Auto
    bss color(actual): 18
    txpower          : 100% (calc 31 oper 31 max 31 dBm)
    beacon_intv      : 100
    rts_threshold    : 2346
    frag_threshold   : 2346
    ap scan          : disable
    ap scan passive  : disabled
    sensor mode      : disabled
    ARRP profile     : ---
    WIDS profile     : ---
    vap on           : enabled
    cca_threshold    : -72
      wlan  0        : ssid1
    max vaps         : 7
    base bssid       : 38:c0:ea:d0:18:c0
    oper chan        : 6
    noise_floor      : -83
    chutil           : enabled
    oper chutil time : Sun Feb 15 15:37:28 2026  (age=4)
    oper chutil data : 98,98,98,99,99, 98,98,99,98,98, 98,98,98,98,98 ->newer
    station info     : 0/0
  Radio 2            : AP
```

```
    80211d enable    : enabled
    80211mc enable   : disabled
    country name     : US
    country code     : 841
    drma_manual_mode : ncf
    radio_type       : 11AX_5G
    channel list     : 36 40 44 48 52 56 60 64
    AI support       : disabled
    darrp            : disabled
    airtime fairness : disabled
    zero wait dfs    : enabled
    MIMO mode        : default
    bss color mode   : Auto
    bss color(actual): 27
    txpower          : 100% (calc 27 oper 27 max 27 dBm)
    beacon_intv      : 100
    rts_threshold    : 2346
    frag_threshold   : 2346
    ap scan          : disable
    ap scan passive  : disabled
    sensor mode      : disabled
    ARRP profile     : ---
    WIDS profile     : ---
    vap on           : enabled
    cca_threshold    : -70
...
```

4. From the FortiAP, verify that the CCA values have been assigned to each radio using the following diagnostic command:

```
FortiAP-431F # cw_diag -c radio-cfg

Radio 0: AP
   country        : cfg=US  oper=US
   countryID      : cfg=841 oper=841
   802.11d enable  : enabled
   802.11mc enable : disabled
   sta info       : 0/0
   radio type     : 11AX_2.4G (pure G)
   mimo,chainmask : 4, 0xf (mimo) 0xf (power) 0xf/0xf (oper)
   airtime fairness   : disabled
   ps optimize    : 0
   tx optimize    : f
   11g prot mode  : 0
   HT20/40 coext  : 1
   beacon intv    : 100
   cca threshold  : -72
   txpwr mode     : set by percentage (100%)
   txpwr cfg/oper/max : 26/26/26 (EIRP +5)
   HE param       : gi=disabled bw=20MHz
   ack timeout    : 64
   r_ac MAX dista : 0 ackt_2G=64 ackt_5G=25
```

```
   r_ac ht_cap    : gi=0 bw=0 bw_ext=1
   r_ac chan      : num=0 age=423884
   channel usable : enabled
   channel        : num=0
   oper_chan      : 6
   r_ac md_cap    :   1,   6,  11,
   r_ac chan list :   1,   6,  11,
        chan list :   1,   6,  11,
   hw_chan list   :   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,
   nol list       :
   chutil meas    : enabled
   sensor mode    : disabled  (applied promisc mode=disabled)
   ap scan        : disabled,
   ap scan thresh : 0 dBm
   darrp          : disabled --info only
   spect analysis : disabled
   bss color mode     : auto
   bss color(actual)  : 18
   wids           : disabled

   fortipresence  : disabled
Radio 1: AP
   country        : cfg=US  oper=US
   countryID      : cfg=841 oper=841
   802.11d enable  : enabled
   802.11mc enable : disabled
   sta info       : 0/0
   radio type     : 11AX_5G
   mimo,chainmask : 4, 0xf0 (mimo) 0xf0 (power) 0xf0/0xf0 (oper)
   airtime fairness   : disabled
   ps optimize    : 0
   tx optimize    : f
   11g prot mode  : 0
   HT20/40 coext  : 1
   beacon intv    : 100
   cca threshold  : -70
   ...
```

Allow FortiAP radio and FortiWiFi local radio to be disabled

This releases adds the ability to turn off the FortiAP radio and FortiWiFi local radio without losing existing SSID configurations. When turning the radio back on, the configurations of the selected SSIDs are kept per radio.

CLI changes

A new `vap-status` option is available under radio configurations in the `wtp-profile`:

```
config wireless-controller wtp-profile
  edit <wtp_name>
    config <radio>
      set vap-status [disable | enable]
    end
  next
end
```

`vap-status`	Enable/disable broadcasting of all configured SSIDs on this radio (default = `enable`)

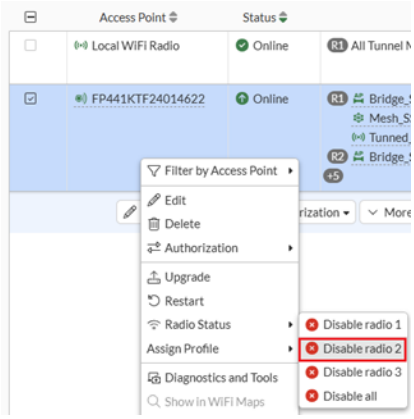
The `vap-status` option can also be overridden on a per radio basis from each FortiAP device.

```
config wireless-controller wtp
  edit <wtp_name>
    config <radio>
      set override-vap-status [enable | disable]
      set vap-status [enable | disable]
    end
  next
end
```

`override-vap-status`	Enable to override WTP profile Virtual Access Point (VAP) enable/disable settings.
`vap-status`	When `override-vap=status` is enabled, enable/disable broadcasting of all configured SSIDs on this radio (default = `enable`).

To disable broadcasting on FortiAP or FortiWiFi radios from the profile level - GUI:

1. Go to *WiFi & Switch Controller* > *Managed FortiAPs* and select the FortiAP or FortiWiFi you want to disable a radio on.
2. Right-click to access the context menu and go to *Radio Status* and select which radio you want to disable.

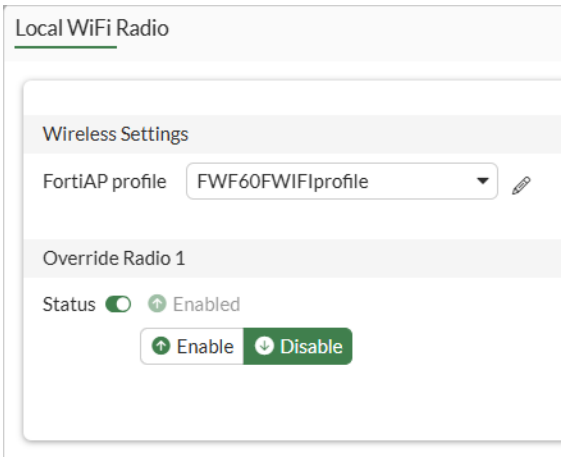


To disable broadcasting on FortiAP or FortiWiFi radios from the profile level - CLI:

```
config wireless-controller wtp-profile
  edit FAP441K-default
    config radio-2
      set vap-status disable
    end
  next
end
```

To disable broadcasting on FortiAP or FortiWiFi radios from the device level - GUI:

1. Go to *WiFi & Switch Controller* > *Managed FortiAPs* and locate the FortiAP or FortiWiFi you want to disable a radio on.
2. Click *Edit* to edit that device.
3. Locate the radio you want to disable broadcasting SSIDs on and enable *Status*, and then select *Disable*.



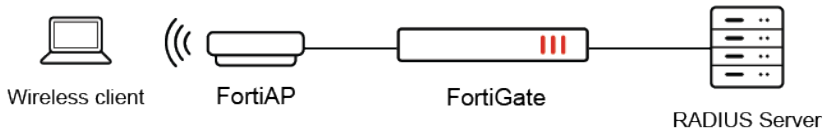
4. When you are finished, click *OK*.

To disable broadcasting on FortiAP or FortiWiFi radios from the device level - CLI:

```
config wireless-controller wtp
  edit FWF60F-WIFI0
    config radio-1
      set override-vap-status enable
      set vap-status disable
    end
  next
end
```

Support DVLAN assignments from RADIUS and VLAN pooling together

When creating a FortiAP profile with RADIUS authentication, FortiOS now supports enabling both Dynamic VLAN (DVLAN) and VLAN pooling simultaneously. The VLAN ID obtained from the RADIUS server takes precedence over the VLAN-Pool assigned VLAN. Once a client successfully authenticates, the wireless controller assigns a VLAN ID based on the Tunnel-Private-Group-ID attribute received in the RADIUS Access-Accept message. If this attribute is not provided, the wireless controller then defaults to assigning a VLAN ID from the local VLAN pool based on the FortiAP the client connects to.



Example configuration using simultaneous DVLAN and VLAN pooling:

1. Create several FortiAP groups.

```
config wireless-controller wtp-group
  edit "group1"
    set wtps "FP233GTF22000023" "FWF81F-WIFI0"
  next
  edit "group2"
    set wtps "FP231FTF20000041" "FP231FTF23037047"
  next
  edit "group3"
    set wtps "FP431FTF20012724"
  next
  edit "group4"
    set wtps "FP231FTF20000051"
  next
end
```

2. Create a FortiAP SSID with RADIUS authentication, enable `dynamic-vlan`, set VLAN pooling to `wtp-group`, and assign the FortiAP groups you created to the VLAN pool.

```
config wireless-controller vap
  edit "wifi2"
    set ssid "FOS_81F_HA_ent_radius"
    set security wpa2-only-enterprise
    set auth radius
    set radius-server "peap"
    set schedule "always"
    set dynamic-vlan enable
    set quarantine enable
    set vlan-pooling wtp-group
    config vlan-pool
      edit 10
```

```
        set wtp-group "group1" "group2"
      next
      edit 20
        set wtp-group "group3" "group4"
      next
    end
  next
end
```

3. To verify that configurations are working, connect a wireless client that can pass RADIUS authentication and receive the "Tunnel-Private-Group" attribute to the SSID. Once it passes RADIUS authentication and receives the attribute `Tunnel-Private-Group-ID=100` in Access-Accept message, then the client is assigned with `vlan_id=100`.
   - The attributes setting of the user:

```
test1   Cleartext-Password := "123456"
           Tunnel-Type = "VLAN",
           Tunnel-Medium-Type = "IEEE-802",
           Tunnel-Private-Group-Id = 100,
           Fortinet-Group-Name = "group1",
           Session-Timeout=300,
           Termination-Action=1
```

   - ```
     FWF-81F-2R-HA_A(Primary) (Interim)# diagnose wireless-controller wlac -d sta online

        vf=0 mpId=6 wtp=1 rId=1 wlan=wifi2 vlan_id=100(dynamic) ip=20.100.80.2
     ip6=fe80::fae4:e3ff:fed8:5eaf mac=f8:e4:e3:d8:5e:af vci= host=WiFi-Client-2 user=test1
     group=group1 signal=-16 noise=-89 idle=2 bw=1 use=5 chan=4 radio_type=11AX_2G
     security=wpa2_only_enterprise mpsk= encrypt=aes cp_authed=no l3r=1,0
     G=0.0.0.0:0,0.0.0.0:0-0-0 -- 0.0.0.0:0 0,0 online=yes mimo=2

                    ip6=*fe80::fae4:e3ff:fed8:5eaf,21,
     ```

4. Now connect a wireless client that can pass RADIUS authentication but does not receive the "Tunnel-Private-Group-ID" attribute in Access-Accept message. Once it passes RADIUS authentication, the client is assigned to the local VLAN pool based on which FortiAP it connects to. In this example, the client connects to FP231FTF23037047, which belongs to `group2`. The client is therefore assigned with `vlan_id=10`.
   - The attributes setting of the user:

```
tester    Cleartext-Password := "123456"
```

   - ```
     FWF-81F-2R-HA_A(Primary) (Interim)# dia wireless-controller wlac -d sta online

        vf=0 mpId=6 wtp=1 rId=2 wlan=wifi2 vlan_id=10 ip=10.1.1.2 ip6=fe80::fae4:e3ff:fed8:5eaf
     mac=f8:e4:e3:d8:5e:af vci= host=WiFi-Client-2 user=tester group=peap signal=-39 noise=-95
     idle=13 bw=0 use=5 chan=153 radio_type=11AX_5G security=wpa2_only_enterprise mpsk=
     encrypt=aes cp_authed=no l3r=1,0 G=0.0.0.0:0,0.0.0.0:0-0-0 -- 0.0.0.0:0 0,0 online=yes
     mimo=2

                    ip6=*fe80::fae4:e3ff:fed8:5eaf,21,
     ```

Support multiple FortiAP groups per VLAN entry for VLAN pooling

This release adds support for assigning multiple FortiAP groups to a single VLAN entry when VLAN pooling is enabled. Assigning multiple FortiAP groups to a VLAN entry is only available from the CLI.

To assign multiple FortiAP groups to a VLAN:

1. Create multiple FortiAP groups.

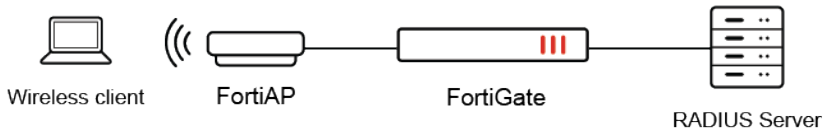
```
config wireless-controller wtp-group
  edit "group1"
    set wtps "FP233GTF22000023" "FWF81F-WIFI0"
  next
  edit "group2"
    set wtps "FP231FTF20000041" "FP231FTF23037047"
  next
  edit "group3"
    set wtps "FP431FTF20012724"
  next
  edit "group4"
    set wtps "FP231FTF20000051"
  next
end
```

2. From the CLI, create a FortiAP SSID, set `vlan-pooling` to `wtp-group`, and then assign multiple FortiAP groups per VLAN entry.

```
config wireless-controller vap
  edit "wifi-test"
    set vlan-pooling wtp-group
    config vlan-pool
      edit 10
        set wtp-group "group1" "group2"
      next
      edit 20
        set wtp-group "group3" "group4"
      next
    end
  next
end
```

Add RADIUS authentication survivability for 802.1X SSIDs

This release enables the FortiOS wireless controller to locally re-authenticate previously authenticated wireless clients if the RADIUS server becomes unreachable. Wireless clients can reconnect using the same credentials, ensuring minimal service disruptions during server outages. You can set a timeout interval so clients will stop being authorized if the RADIUS server remains down.



CLI changes

Two new CLI options have been added to `config wireless-controller vap`.

```
config wireless-controller vap
  edit <vap_name>
    set security wpa2-only-enterprise
    set auth radius
    set radius-server <radius_server_name>
    set radius-auth-survivability [enable | disable]
    set radius-auth-surviv-intv <integer>
  next
end
```

`radius-auth-survivability`	Enable/disable RADIUS authentication survivability (default = disable).
`radius-auth-surviv-intv`	This option is available only when `radius-auth-survivablility` is enabled. Set the RADIUS authentication survivability cache timeout interval in seconds (3600 - 864000, default = 86400). If the RADIUS server is still unreachable after the timeout interval, the client will no longer be authorized.

To configure RADIUS authentication survivability:

If the client authentication method is EAP-PEAP, then a RADIUS attribute "Fortinet-AS-Credential-Hash" or Cisco-AVPair = "AS-Credential-Hash=" should be added under the user account on the RADIUS server.

1. Configure the RADIUS server.

```
config user radius
  edit "peap"
    set uuid e29b5212-11aa-51f1-8909-db5b5ed8cc22
    set server "172.16.200.55"
    set secret *
  next
end
```

2. Import the root CA certificate and server certificate of the RADIUS server.

```
config vpn certificate ca
  edit "CA_Cert_1"
    set range global
  next
```

```
end
config vpn certificate local
  edit "Radius_Cert"
    set password *
    set range global
  next
end
```

3. Under `system global`, select the certificates.

```
config system global
  set wifi-ca-certificate "CA_Cert_1"
  set wifi-certificate "Radius_Cert"
end
```

4. Configure a FortiAP SSID with RADIUS authentication, and enable RADIUS authentication survivability settings:

```
config wireless-controller vap
  edit "wifi2"
    set ssid "FOS_81F_HA_ent_radius"
    set security wpa2-only-enterprise
    set radius-auth-survivability enable
    set radius-auth-surviv-intv 3600
    set auth radius
    set radius-server "peap"
    set schedule "always"
  next
end
```

5. To verify that settings have been applied, connect a wireless client (in this example, the user is called "test1"), and confirm the client is saved in the RADIUS cache:
   - Attributes of "test1":

```
test1    Cleartext-Password := "123456"
            Tunnel-Type = "VLAN",
            Tunnel-Medium-Type = "IEEE-802",
            Tunnel-Private-Group-Id = 100,
            Fortinet-Group-Name = "group1",
            Fortinet-AS-Credential-Hash= "32ED87BDB5FDC5E9CBA88547376818D4",
            Session-Timeout=300,
            Termination-Action=1
```

   - Diagnose commands:

```
# diagnose wireless-controller wlac -d sta online
   vf=0 mpId=0 wtp=1 rId=2 wlan=wifi2 vlan_id=0 ip=10.20.80.2
ip6=fe80::fae4:e3ff:fed8:5eaf mac=f8:e4:e3:d8:5e:af vci= host= user=test1 group=group1
signal=-45 noise=-95 idle=1 bw=0 use=5 chan=153 radio_type=11AX_5G security=wpa2_only_
enterprise mpsk= encrypt=aes cp_authed=no l3r=1,0 G=0.0.0.0:0,0.0.0.0:0-0-0 -- 0.0.0.0:0
0,0 online=yes mimo=2
                ip6=*fe80::fae4:e3ff:fed8:5eaf,1,
```

```
# diagnose wpa wpad radius-auth-surviv-cache FOS_81F_HA_ent_radius
SSID config: SSID(FOS_81F_HA_ent_radius) VAP(wifi2) cache_cnt(1)
    STA=f8:e4:e3:d8:5e:af, expiration 3592, vlan_id 100, vlan_name , eap_type 25, mac_auth
0
         session_timout_set 1, session_timout_intv 300, termination_action 1, acct_
interim_intv 0
         user test1, usergroup_cnt 1, class_cnt 0
```

6. Disconnect the RADIUS server and verify that the wireless client can still pass authentication and successfully reconnect:

```
# diagnose test authserver radius peap chap test1 123456
authenticate 'test1' against 'chap' failed, assigned_rad_session_id=21668246929410 session_
timeout=0 secs idle_timeout=0 secs!

# diagnose wireless-controller wlac -d sta online
   vf=0 mpId=0 wtp=2 rId=2 wlan=wifi2 vlan_id=0 ip=10.20.80.2 ip6=fe80::fae4:e3ff:fed8:5eaf
mac=f8:e4:e3:d8:5e:af vci= host= user=test1 group=group1 signal=-38 noise=-95 idle=9 bw=0
use=5 chan=48 radio_type=11AX_5G security=wpa2_only_enterprise mpsk= encrypt=aes cp_authed=no
l3r=1,0 G=0.0.0.0:0,0.0.0.0:0-0-0 -- 0.0.0.0:0 0,0 online=yes mimo=2
               ip6=*fe80::fae4:e3ff:fed8:5eaf,23,
```

If the RADIUS server is still unreachable after the configured `radius-auth-surviv-intv` timeout, the wireless client will no longer be authorized.

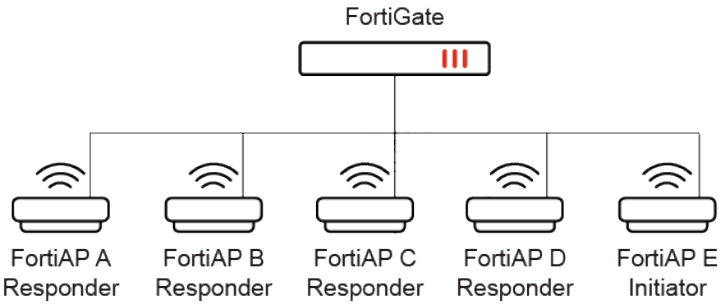
Once the RADIUS server is up, the wireless client will be authorized by the RADIUS server again.

Automated FortiAP location positioning system

This release adds support for an automated FortiAP location positioning system that enables the AP to determine its geographic location using the following three methods listed in order of priority:

1. **User input (Static LoWi):** The manual user-configured location on the controller.
2. **GPS input:** Coordinates obtained from the AP's built-in GNSS systesm.
3. **Location Wireless Interface (LoWi) calculation:** Automatic position calculation using Wi-Fi RTT measurements defined in IEEE 802.11mc. This setup needs at least 4 responder APs and 1 initiator AP (5 AP in total) to thoroughly run the whole LoWi calculation.

   The LoWi calculation process enables an initiator AP to estimate its location through 3D trilateration, measuring the distance to nearby responder APs that already have known positions. This allows for automated AP location discovery when GPS or manual configuration is unavailable, and can determine AP coordinates using existing Wi-Fi infrastructure without requiring additional positioning hardware. Certain regulatory environments, such as AFC requirements for wireless devices, increasingly require accurate device positioning.



If the user inputted information (Static LoWi) is not used, then the FortiAP uses the GPS coordinates, if available. If the GPS coordinates are not available, then the FortiAP uses the LoWi-based position calculation.

CLI changes

CLI options have been added to configure the LoWi positioning system.

A command to configure the 802.11mc mode has been added under the WTP profile.

```
config wireless-controller wtp-profile
  edit <name>
    set 80211mc-mode [auto | initiator | responder]
  next
end
```

`80211mc-mode`	Configure the 802.11mc mode (default = `auto`). • `auto`: The FortiAP automatically selects initiator or responder mode. • `initiator`: The FortiAP sends RTT measurement requests to nearby responder APs. • `responder`: The FortiAP responds to RTT measurement requests from initiators.

The `802.11mc` setting that was introduced by FortiOS 7.6.0 has been revised with the following:

- The function of `set 80211mc` has been changed to enable or disable 802.11mc RTT measurement. It is disabled by default.
- 802.11mc can only be configured on the 5GHz radio to ensure a more accurate and reliable IEEE 802.11mc RTT measurement. It is no longer configurable on the 2.4GHz and 6GHz radio.

```
conf wireless-controller wtp-profile
  edit <name>
    config radio-1
      set 80211mc [enable | disable]
    end
  next
end
```

`80211mc`	Enable or disable 802.11mc RTT measurement. (default = `disable`).

	• `enable`: Enable radio 802.11mc capability. • `disable`: Disable radio 802.11mc capability. 802.11mc can only be configured on the 5GHz radio to ensure a more accurate and reliable IEEE 802.11mc RTT measurement.

Commands to configure the static LoWi have been added under each WTP entry, so the user-configured location can be manually added on individual FortiAP devices.

```
config wireless-controller wtp
  edit <device name>
    set static-lowi [enable | disable]
    set static-lowi-lat <integer>
    set static-lowi-lon <integer>
    set static-lowi-alt <integer>
    set static-lowi-lat-err <integer>
    set static-lowi-lon-err <integer>
    set static-lowi-alt-err <integer>
  next
end
```

`static-lowi`	Enable or disable the static LoWi position of the FortiAP. • `enable`: Enable static LoWi position of the FortiAP. • `disable`: Disable static LoWi position of the FortiAP.
`static-lowi-la`	Set the static LoWi latitude position of the AP (between -90 and 90 degrees).
`static-lowi-lon`	Set the static LoWi longitude position of the AP (between -180 and 180 degrees).
`static-lowi-alt`	Set the static LoWi altitude position of the AP in meters (between 0 and 500 meters).
`static-lowi-lat-err`	Set the expected position error tolerance of the static LoWi latitude for the AP (between 0 and 50 meters).
`static-lowi-lon-err`	Set the expected position error tolerance of the static LoWi longitude for the AP (between 0 and 50 meters).
`static-lowi-alt-err`	Set the expected position error tolerance of the static LoWi altitude for the AP (between 0 and 50 meters).

To view the GPS coordinates of a FortiAP model with a built-in GNSS:

1. Enter the following diagnose command:
   `diagnose wireless-controller wlac -c wtp <FAP SN>`

```
diagnose wireless-controller wlac -c wtp FP441KTF23001330
... ...
    lowi-pos-type : gps
    lowi-lat : 49.25930000(d) +/- 4(m)
    lowi-lon : -123.01158500(d) +/- 4(m)
    lowi-alt : 23(m) +/- 17(m)
... ...
```

To manually configure the static LoWi position of a FortiAP:

1. Manually configure the static LoWi position of an individual FortiAP:

```
config wireless-controller wtp
  edit "FP431GTY23049984"
    set static-lowi enable
    set static-lowi-lat "49"
    set static-lowi-lon "-123"
    set static-lowi-alt 33
    set static-lowi-lat-err 9
    set static-lowi-lon-err 10
    set static-lowi-alt-err 21
  next
end
```

2. To verify the position has been configured, enter the following diagnose command:
`diagnose wireless-controller wlac -c wtp <FAP SN>`

```
diagnose wireless-controller wlac -c wtp FP431GTY23049984
    ...
    lowi-pos-type : static
    lowi-lat : 49.00000000(d) +/- 9(m)
    lowi-lon : -123.00000000(d) +/- 10(m)
    lowi-alt : 33(m) +/- 21(m)
    ...
```

To configure LoWi position calculation using 802.11mc:

In this scenario, the AP location is calculated using LoWi trilateration based on 802.11mc RTT measurements between neighboring FortiAPs. This example uses 4 responder APs that already have a known location and 1 initiator AP.

1. Configure a FortiAP profile for the responder role and enable `80211mc` in the 5GHz radio. Then apply that profile to four FortiAPs with known locations to designate them as responder APs.

```
config wireless-controller wtp-profile
    edit "FAP431G-default"
        set 80211mc-mode responder
        config radio-2
            set band 802.11a 802.11n-5G 802.11ac-5G 802.11ax-5G
            set channel-bonding 40MHz
            set 80211mc enable
            set vap-all bridge
        end
    next
end
```

2. Verify the FortiAPs have been successfully configured as a responder. You can verify with the following methods:

a. From the FortiGate, run the following diagnose command:

```
FortiGate-301E (vdom1) # diagnose wireless-controller wlac -c wtp FP431GTY23049984
…
    lowi-pos-type : static
    lowi-lat : 49.00000000(d) +/- 9(m)
    lowi-lon : -123.00000000(d) +/- 10(m)
    lowi-alt : 33(m) +/- 21(m)
…
Radio 2            : AP
    80211mc enable   : enabled
…
```

b. From the FortiAP CLI, run the following command:

```
FortiAP-431G # rcfg
...
   802.11mc enable : enabled
   802.11mc role : cfg=responder oper=responder
...
```

3. Configure a FortiAP profile for the initiator role and enable `80211mc` in the 5GHz radio. Then apply that profile to the FortiAP designated as the initiator AP.

```
config wireless-controller wtp-profile
  edit "FAP441K-default-initiator"
    set 80211mc-mode initiator
    config radio-2
      set band 802.11a 802.11n-5G 802.11ac-5G 802.11ax-5G 802.11be-5G
      set channel-bonding 40MHz
      set 80211mc enable
      set vap-all bridge
    end
  next
end
config wireless-controller wtp
  edit "FP441KTF24014688"
    set wtp-profile "FAP441K-default-initiator"
  next
end
```

After applying the profile the LoWi calculation begins.

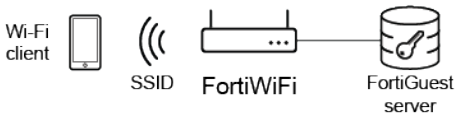
4. To view the results of the LoWi auto positioning calculation, enter the following diagnose command:

```
FortiGate-80F-POE (Interim)# diagnose wireless-controller wlac -c wtp FP441KTF24014688
…
    lowi-pos-type    : lowi
    lowi-lat         : 49.25901913(d) +/- 30(m)
    lowi-lon         : -123.01071321(d) +/- 30(m)
    lowi-alt         : 3(m) +/- 27(m)
…
Radio 2            : AP
```

```
    80211mc enable    : enabled
…
```

Support FortiGuest assignments of RADIUS attributes to MPSK groups

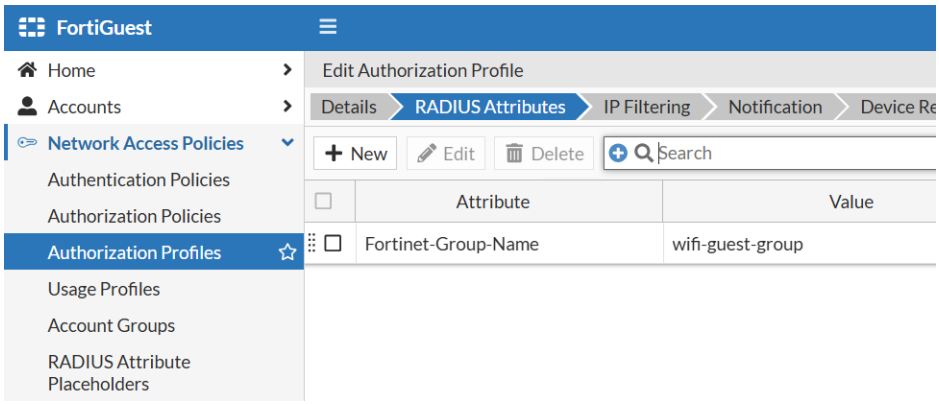
In this release, FortiGate can support dynamic policy enforcement for FortiGuest MPSK-secured SSIDs. When a client connects through a WPA2-Personal or WPA3-SAE SSID, the controller can process the user and group information directly from the RADIUS Access-Accept message. This enables the system to automatically apply the correct security policies and access levels on the connecting device based on its FortiGuest credentials.



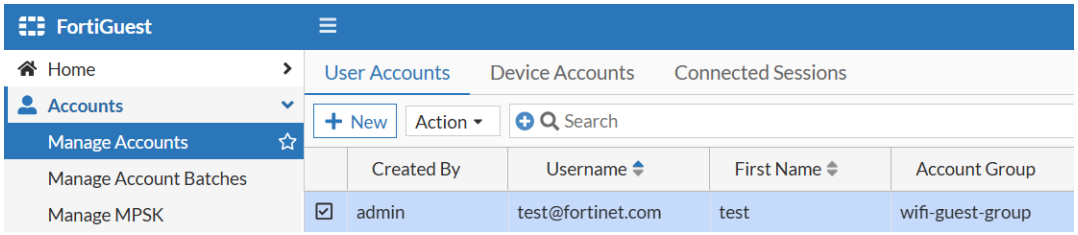
Example configuration

To configure an MPSK key and Authorization policy on FortiGuest:

1. From FortiGuest, go to *Network Access Policies* > *Authorization Profiles* and create or edit an Authorization Profile.
2. Add the *Fortinet-Group-Name* attribute and set the value to "*wifi-guest-group*".



3. Go to *Accounts* > *Managed Accounts* and create a user account, for example *test@fortinet.com*, and set the account group to *wifi-guest-group*.



4. Go to *Accounts* > *Manage MPSK* and create an MPSK key with the following:

Name	new-mpsk
Static VLAN	100
User Account	test@fortinet.com
Devices	Enter the MAC address of the device associated with the user account. For example, F8-E4-E3-D8-5E-AF

To configure the FortiGate to accept the FortiGuest MPSK and apply policies - CLI:

1. From the FortiGate, create a User Group with the name "`wifi-guest-group`" to match the *Fortinet-Group-Name* attribute you configured in FortiGuest.

```
config user group
  edit "wifi-guest-group"
end
```

2. Configure the external FortiGuest server with the FortiGuest server IP.

```
config user radius
    edit "fortiguest"
        set server "172.16.200.117"
        set secret *
        set radius-coa enable
        set require-message-authenticator disable
        set acct-all-servers enable
        config accounting-server
            edit 2
                set status enable
                set server "172.16.200.55"
                set secret *
            next
        end
    next
end
```

3. Create an MPSK profile, enable MPSK external server authentication, and apply the external FortiGuest server you created (`fortiguest`).

```
config wireless-controller mpsk-profile
    edit "wifi-guest"
        set mpsk-external-server-auth enable
        set mpsk-external-server "fortiguest"
    next
end
```

4. Apply the MPSK profile to a VAP.

```
config wireless-controller vap
    edit "wifi"
        set ssid "FOS_81F_MPSK"
        set schedule "always"
        set mpsk-profile "wifi-guest"
        set dynamic-vlan enable
    next
end
```

5. Create a sub-VLAN interface (`wifi.v1`), set the VLAN ID to `100`, and apply the VAP interface you created (`wifi`).

```
config system interface
    edit "wifi.v1"
        set vdom "root"
        set ip 10.100.80.1 255.255.255.0
        set allowaccess ping
        set device-identification enable
        set role lan
        set snmp-index 47
        set ip-managed-by-fortiipam disable
        set interface "wifi"
        set vlanid 100
    next
end
```

6. Then apply the sub-VLAN interface you created to a DHCP server.

```
config system dhcp server
    edit 12
        set forticlient-on-net-status disable
        set dns-service default
        set default-gateway 10.100.80.1
        set netmask 255.255.255.0
        set interface "wifi.v1"
        config ip-range
            edit 1
                set start-ip 10.100.80.2
                set end-ip 10.100.80.254
            next
        end
    next
end
```

7. Create a firewall policy accepting traffic from "wifi.v1" to "wan1" and set the User Group to `wifi-guest-group`.

```
config firewall policy
    edit 4
        set name "wifi.v1"
        set srcintf "wifi.v1"
        set dstintf "wan1"
        set action accept
```

```
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
        set groups "wifi-guest-group"
    next
end
```

To verify the FortiGate can accept FortiGuest MPSK users and can apply firewall policies:

1. Connect the wireless client with the configured MPSK key from FortiGuest to the SSID and confirm the following:
   - The client can be assigned to VLAN ID 100, and get the correct IP(*10.100.80.2*) from the sub-VLAN interface (*wifi.v1*).
   - The client can receive the correct User Name (*test@fortinet.com*) and Group Name (*wifi-guest-group*) from FortiGuest
   - The MPSK name (*new-mpsk*) can be received from FortiGuest and is recorded in the wireless station list

```
diagnose wireless-controller wlac -d sta online

   vf=0 mpId=0 wtp=2 rId=2 wlan=wifi vlan_id=100(dynamic) ip=10.100.80.2 ip6=::
mac=f8:e4:e3:d8:5e:af vci= host=WiFi-Client-2 user=test@fortinet.com group=wifi-guest-group
signal=-45 noise=-95 idle=7 bw=2 use=5 chan=149 radio_type=11AX_5G security=wpa2_only_personal
mpsk=new-mpsk encrypt=aes cp_authed=no l3r=1,0 G=0.0.0.0:0,0.0.0.0:0-0-0 -- 0.0.0.0:0 0,0
online=yes mimo=2
```

2. Confirm that a firewall authentication session of the wireless station can be created.

```
diagnose firewall auth list

10.100.80.2, test@fortinet.com
    src_mac: f8:e4:e3:d8:5e:af
    type: wsso, id: 0, duration: 16, idled: 16
    expire: 1184, allow-idle: 1200
    packets: in 0 out 0, bytes: in 0 out 0
    group_id: 7
    group_name: wifi-guest-group
```

Switch controller

This section includes information about switch-controller-related new features:

- Add maximum power control for PoE ports on page 408
- Increase number of supported switches on page 409
- Advertise and control EEE using LLDP in the switch controller on page 409

- Support trusted host settings for managed switches on page 410
- Control when custom commands are pushed on page 412
- Provide private data encryption for managed switches on page 414
- Support MAC move for managed switches on page 415
- Enable advanced switching features in the GUI on page 420
- Control the number of 802.1X-authenticated clients per port on page 422
- Support IGMP-snooping static group configuration on page 422
- Dynamic MAB session handling on page 423
- Enhance NAC/DPP port control and QoS/PoE support on page 424
- Add global port-selection criteria for some switch models on page 426
- Using NAC with 802.1X authentication in the FortiOS GUI on page 427

Add maximum power control for PoE ports

You can now use the CLI to set the maximum amount of power on power over Ethernet (PoE) ports to 30 W, 60 W, or the maximum amount of power for that port.

To change the maximum amount of power for a PoE port:

```
config switch-controller managed-switch
   edit <FortiSwitch_serial_number>
      config ports
         edit <port_name>
            set poe-max-power-mode {30W | 60W | class-based}
         end
      end
```

Power mode	Description
30W	The maximum amount of power on this port is 30 W.
60W	The maximum amount of power on this port is 60 W.
class-based	The maximum amount of power on this port is based on the class. This power mode is the default.

For example, if you want port10 of a switch to use a maximum of 30 W:

```
config switch-controller managed-switch
   edit S548DF5018000776
      config ports
         edit port10
            set poe-max-power-mode 30W
         end
      end
```

Increase number of supported switches

The number of supported FortiSwitch devices has increased for some FortiGate, FortiWiFi, and FortiFirewall models. This change enhances scalability for larger deployments.

FortiGate, FortiWiFi, or FortiFirewall model	Number of supported FortiSwitch units
FG-50G, FG-50GP, FG-51G, FG-51GP, FG-50G-5G, FG-51G-5G, FG-50G-DSL, FG-50G-SFP, FWF-50G, FWF-51G, FWF-50G-SFP, FWF-50G-5G, FWF-51G-5G, FWF-50G-DSL	16 (from 8)
FG-200G, FG-201G	96 (from 64)
FG-2600F, FFW-2600F, FG-2601F	300 (from 196)

Advertise and control EEE using LLDP in the switch controller

You can now use the LLDP 802.3 TLV to advertise and control the energy-efficient Ethernet (EEE) configuration on managed FortiSwitch units, allowing EEE capabilities to be broadcast across the network.

To advertise the EEE configuration in the LLDP 802.3 TLV:

```
config switch-controller lldp-profile
       edit <profile_name>
              set 802.3-tlvs eee-negotiation
       next
end
```

For example:

```
config switch-controller lldp-profile
       edit <LLDP_EEEprofile>
              set 802.3-tlvs eee-negotiation
       next
end
```

You can also specify the number of microseconds that circuits are turned off to save power for EEE(with the `set eee-tx-idle-time <integer>` command) and the number of microseconds during which no data is transmitted while the circuits that were turned off are being restarted (with the `set eee-tx-wake-time <integer>` command). The range of values for both commands is 0-2,560 microseconds. The default value for the `set eee-tx-idle-time <integer>` command is 60 microseconds; the default value for the `set eee-tx-wake-time <integer>` command is 30 microseconds. You must enable EEE before configuring these two values.

When you change the `eee-tx-wake-time` value, the port resets, and the connection is lost briefly.

To configure EEE:

```
config switch-controller managed-switch
       edit <FortiSwitch_serial_number>
               config ports
                       edit <port_name>
                              set energy-efficient-ethernet enable
                              set eee-tx-idle-time <integer>
                              set eee-tx-wake-time <integer>
                       next
               end
       next
end
```

For example, to use EEE on port 7 on a managed FortiSwitch unit:

```
config switch-controller managed-switch
       edit S548DF5018000776
               config ports
                       edit port7
                              set energy-efficient-ethernet enable
                              set eee-tx-idle-time 500
                              set eee-tx-wake-time 200
                       next
               end
       next
end
```

To confirm that the EEE configuration has been pushed to the managed FortiSwitch unit, log in to the managed switch and check `config switch lldp profile` and `config switch physical-port`.

Support trusted host settings for managed switches

You can now specify trusted IPv4 and IPv6 hosts for admin accounts in FortiOS for managed FortiSwitch units. This feature enhances network security by restricting access for admins to specific IP addresses.

The trusted hosts are specified in a security policy for the admin of the managed FortiSwitch units. You can configure multiple admin security policies.

Use the `set auto` command to automatically assign the IP addresses that are connected to the FortiLink interface as trusted hosts. By default, the `set auto` command is disabled, and you need to specify the trusted hosts manually. When the `set auto` command is enabled, you can still add additional IP addresses for trusted hosts.

Notes

- If the path to the FortiSwitch unit uses NAT, the FortiGate device might not push the correct IP address when `set auto` is enabled. You might need to disable `set auto` and manually add the IP addresses for the trusted hosts.
- If you are using layer-2 FortiLink and `set auto` is enabled, the FortiGate FortiLink IPv4 address is pushed with a mask of /32, and the FortiGate FortiLink IPv6 address is pushed with a prefix of /128.

- If you are using layer-3 FortiLink, the `set auto` configuration is ignored, and you must specify the trusted hosts manually.

To configure trusted hosts:

1. Create a security policy for the FortiSwitch admin:

```
config switch-controller security-policy admin
   edit "<policy_name>"
      set auto {enable | disable}
      set trusthost1 <IPv4_address_netmask>
      set trusthost2 <IPv4_address_netmask>
      set trusthost3 <IPv4_address_netmask>
      set trusthost4 <IPv4_address_netmask>
      set trusthost5 <IPv4_address_netmask>
      set trusthost6 <IPv4_address_netmask>
      set trusthost7 <IPv4_address_netmask>
      set trusthost8 <IPv4_address_netmask>
      set trusthost9 <IPv4_address_netmask>
      set trusthost10 <IPv4_address_netmask>
      set ip6-trusthost1 <xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx>
      set ip6-trusthost2 <xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx>
      set ip6-trusthost3 <xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx>
      set ip6-trusthost4 <xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx>
      set ip6-trusthost5 <xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx>
      set ip6-trusthost6 <xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx>
      set ip6-trusthost7 <xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx>
      set ip6-trusthost8 <xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx>
      set ip6-trusthost9 <xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx>
      set ip6-trusthost10 <xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx>
   next
end
```

   For example:

```
config switch-controller security-policy admin
   edit "trusthost"
      set auto enable
      set trusthost2 1.1.1.0 255.255.255.0
   next
end
```

2. Associate the admin security policy with the FortiLink interface to apply the admin security policy to all managed FortiSwitch units using that FortiLink interface.

```
config switch-controller fortilink-settings
   edit <name_of_FortiLink_settings>
      set fortilink <FortiLink_interface>
      set admin-policy  <name_of_admin_security_policy?
   next
end
```

   For example:

```
config switch-controller fortilink-settings
   edit "FLsettings1"
      set fortilink "port1"
      set admin-policy “trusthost”
      config nac-ports
         set onboarding-vlan "onboarding"
         set lan-segment enabled
         set nac-lan-interface "nac_segment"
```

```
            set nac-segment-vlans "voice" "video"
        end
    next
end
```

3. Check the `switch-controller-fortilink-settings` value.

   The `switch-controller-fortilink-settings` value is set automatically during FortiLink discovery. The value cannot be changed while managed switches are connected to the FortiGate device. If the value is not set or set incorrectly, disconnect the managed switches from the FortiGate device and then reconnect them and rediscover them. You can also use the `factoryreset` command to establish FortiLink again.

```
config system interface
    edit <FortiLink_interface>
        set fortilink enable
        set switch-controller-fortilink-settings "<name_of_FortiLink_settings>"
    next
end
```

   For example:

```
config system interface
    edit "port1"
        set vdom "root"
        set fortilink enable
        set ip 10.255.1.1 255.255.255.0
        set allowaccess ping fabric
        set lldp-reception enable
        set lldp-transmission enable
        set snmp-index 26
        set auto-auth-extension-device enable
        set switch-controller-nac "FLsettings1"
        set switch-controller-dynamic "FLsettings1"
        set switch-controller-fortilink-settings "FLsettings1"
    next
end
```

4. Log in to the managed FortiSwitch unit to confirm the trusted host settings.

   In the following example, trusthost1 was automatically assigned, and trusthost2 was manually assigned.

```
config system admin
    edit "admin"
        set trusthost1 10.255.1.1 255.255.255.255
        set trusthost2 1.1.1.0 255.255.255.0
        set accprofile "super_admin"
        set comments "FortiGate switch controller configured password"
        set password ENC SH2JE5qDKCxoWJrF5iLDSKGmH8E4j2b0F5JTLou4tIffCq7va6K29uQQIedvbQ=
    next
end
```

Control when custom commands are pushed

You can now control when custom commands for a specific managed FortiSwitch unit are pushed from the FortiGate device to the managed FortiSwitch unit. Before FortiOS 8.0.0, custom commands for managed FortiSwitch units were pushed only when the full configuration was updated.

Starting in FortiOS 8.0.0, you can specify that custom commands for a specific managed FortiSwitch unit are pushed after any configuration change on the manged FortiSwitch unit. The custom commands are pushed last after the other configuration changes are pushed.

This new command affects only custom commands bound to a specific FortiSwitch unit.

To control when custom commands are pushed:

```
config switch-controller global
       set switch-custom-cmd {on-replay | on-any}
end
```

Variable	Description
on-replay	Custom commands bound to a specific managed FortiSwitch unit are pushed only when the full configuration is updated. This option is the default.
on-any	Custom commands bound to a specific managed FortiSwitch unit are pushed after any configuration change on the manged FortiSwitch unit. The custom commands are pushed last after the other configuration changes are pushed.

Example

1. Configure a custom command on the FortiGate device.

```
config switch-controller custom-command
       edit "change-auto-discovery-interval"
       set command "config switch interface
edit port1
set auto-discovery-fortilink-packet-interval 30
end
"
       next
end
```

2. Bind the custom command to a managed FortiSwitch unit.

```
FGT_A (vdom1) (Interim)# config switch-controller managed-switch
FGT_A (managed-switch) (Interim)# edit S524DN4K16000116
FGT_A (S524DN4K16000116) (Interim)# config custom-command
FGT_A (custom-command) (Interim)# show
config custom-command
       edit "change-auto-discovery-interval"
              set command-name "change-auto-discovery-interval"
       next
end
```

3. Configure the custom command to be pushed when any configuration on the managed FortiSwitch unit is updated.

```
config switch-controller global
       set switch-custom-cmd on-any
end
```

4. Change any configuration on the managed FortiSwitch unit that the custom command is bound to.

```
FGT_A (vdom1) (Interim)# config switch-controller managed-switch
FGT_A (managed-switch) (Interim)# edit S524DN4K16000116
FGT_A (S524DN4K16000116) (Interim)# config ports
FGT_A (ports) (Interim)# edit port1
FGT_A (port1) (Interim)# set vlan vlan12
FGT_A (port1) (Interim)# end
FGT_A (S524DN4K16000116) (Interim)# end
```

5. Verify that the custom command is pushed to the managed FortiSwitch unit.

```
0: config switch interface
0: edit port1
0: set auto-discovery-fortilink-packet-interval 30
0: end
```

Provide private data encryption for managed switches

Prior to FortiOS 8.0.0, you had to specify a private data encryption key on each FortiSwitch unit, either the same one for all switches or a different one for each switch. With FortiOS 8.0.0, you can now configure a private data encryption key for all managed switches on the FortiGate device. This centralized configuration simplifies network administration and reduces the chance of errors. Using private data encryption prevents exposing credentials in plain text when the FortiGate configuration is backed up.

Before FortiOS 8.0.0, each FortiSwitch unit used a different private data encryption key by default. When the CLI configuration for the encrypted RADIUS secret (`config user radius`) or encrypted TACACS+ key (`config user tacacs+`) was copied from one switch to another with the default private data encryption setting, authentication failed. After the private-data-encryption key is configured in FortiOS 8.0.0 and then pushed to all managed FortiSwitch units, you can use a custom command to push the RADIUS secret or TACACS+ key to all managed FortiSwitch units, and authentication succeeds.

Private data encryption uses an AES 128-bit key. By default, private data encryption is disabled in FortiOS.

To configure private data encryption:

```
config switch-controller switch-profile
   edit "<FortiSwitch_profile_name>"
      set private-data-encryption enable
      set private-data-encryption-key <32-digit hexadecimal number>
   next
end
```

Use case

In this example, the user uses a custom command to push the RADIUS secret from the FortiGate device to a managed FortiSwitch unit without revealing the secret in plain text when the FortiGate configuration is backed up.

1. Configure a private data encryption key on the FortiGate device.

```
config switch-controller switch-profile
   edit default
```

```
      set private-data-encryption enable
      set private-data-encryption-key <32-digit hexadecimal number>
   next
end
```

2. Configure RADIUS server on the managed FortiSwitch unit.

```
config user radius
   edit "RADIUS_Server"
      set secret ENC <encrypted_password>
      set server "172.18.60.203"
   next
end
```

3. Copy the encrypted password from the managed FortiSwitch unit.
4. Configure a custom command on the FortiGate device using the encrypted password.

```
config switch-controller custom-command
   edit "configure-radius"
      set command "config user radius
   edit RADIUS_Server
set server 172.18.60.203
set secret ENC <encrypted_password>
end
"
   next
end
```

5. Push the custom command from the FortiGate device to the managed FortiSwitch unit.

```
config switch-controller managed-switch
   edit S524DN4K16000116
      config custom-command
         edit 1
            set command-name "configure-radius"
         next
      end
   next
end
```

6. Verify that the RADIUS configuration is pushed down from the FortiGate device.

```
S524DN4K16000116 # show user radius
config user radius
   edit "RADIUS_Server"
      set secret ENC <encrypted_password>
      set server "172.18.60.203"
   next
end
```

7. Verify that RADUS authentication works on the managed FortiSwitch unit.

```
S524DN4K16000116 # diagnose test authserver radius Radius_Server pap test1 123456
authenticate 'test1' against 'pap' succeeded, server=primary assigned_rad_session_id=238157828
      session_timeout=0 secs!
Group membership(s) - group1
```

Support MAC move for managed switches

You can now move an 802.1X-authenticated client device between ports that are not directly connected to the FortiSwitch unit without having to delete the 802.1X session or make any other configuration changes. The

switch controller reauthenticates the client device that has been disconnected from one port and then connected to a different port, even if the client device is behind a hub or IP phone. For example, you can move an 802.1X client PC that connects through an IP phone to port1 of the FortiSwitch unit to a port of a third-party switch that connects to port2 of the FortiSwitch unit. MAC move improves flexibility and reliability in dynamic network environments.

This feature is available for 802.1X port-based authentication and 802.1X MAC-based authentication. You can have MAB enabled or disabled and EAP pass-through mode enabled or disabled.

When you are using the MAC move feature with EAP authentication, you need to disable `eap-egress-tagged` to force the switch to always use the untagged EAP response. By default, `eap-egress-tagged` is disabled.

To use MAC move:

1. Enable MAC move. By default, MAC move is enabled. **NOTE:** The location of the CLI commands depends on which FortiSwitch model you are using.

   If you are using FSR-124D, FSR-424F_POE, FS-108E, FS-108E-POE, FS-108E-FPOE, FS-108F, FS-108F-POE, FS-108F-FPOE, FS-110G-FPOE, FS-124E, FS-124E-POE, FS-124E-FPOE, FS-124G, FS-124G-FPOE, FS-148E, FS-148E-POE, 200 Series, FS-4xxE, 500 Series, FS-1024D, FS-1024E, FS-T1024E, FS-1048E, FS-1048G, FS-2048F, FS-3032D, FS-3032E, and FS-3032G

   ```
   config switch-controller security-policy 802-1X
          edit "<security_policy_name>"
                  set eap-egress-tagged {disable | enable}
                  set allow-mac-move enable
          next
   end
   ```

   If you are using FS-124F, FS-124F-POE, FS-124F-FPOE, FS-148F, FS-148F-POE, or FS-148F-FPOE:

   ```
   config switch-controller 802-1X-settings
          set eap-egress-tagged {disable | enable}
          set allow-mac-move enable
   end
   ```

2. Assign the 802.1X security policy to the ports of the managed switch that are used to communicate with the client device.

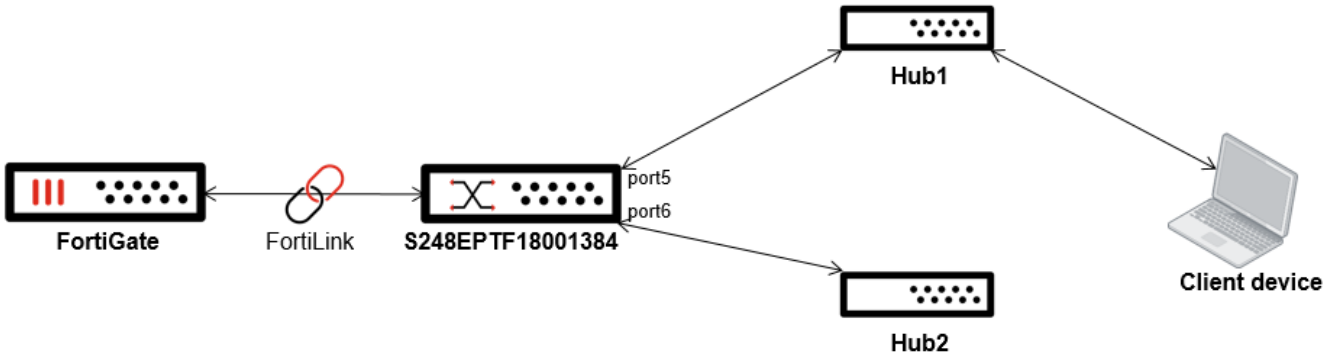
   ```
   config switch-controller managed-switch
          edit <FortiSwitch_serial_number>
                  config ports
                          edit <port_name>
                                  set port-security-policy <security_policy_name>
                          next
                          .
                          .
                          .
                  end
          next
   end
   ```

3. Disconnect the client device from the port used to communicate with the managed FortiSwitch unit.

4. Connect the client device to a different port that can communicate with the managed FortiSwitch unit.

Use case

In this example, the FortiGate device is managing one FortiSwitch unit, S248EPTF18001384. The FortiSwitch unit is connect to two hubs, Hub1 and Hub2. The network administrator disconnects the client device from Hub1 and then connects the client device to Hub2.



1. Configure 802.1x authentication on the FortiGate device and enable MAC move.

```
config switch-controller security-policy 802-1X
       edit "8021Xpolicy"
              set security-mode 802.1X
              set user-group "1X_USER_GROUP"
              set mac-auth-bypass disable
              set open-auth disable
              set eap-passthru disable
              set eap-auto-untagged-vlans enable
              set guest-vlan disable
              set guest-auth-delay 30
              set auth-fail-vlan disable
              set framevid-apply enable
              set radius-timeout-overwrite disable
              set policy-type 802.1X
              set authserver-timeout-vlan disable
              set dacl disable
              set eap-egress-tagged disable
              set allow-mac-move enable
       next
end
```

2. Assign the 802.1X security policy to port5, which communicates with Hub1, and port6, which communicates with Hub2.

```
config switch-controller managed-switch
       edit S248EPTF18001384
              config ports
                     edit port5
                            set port-security-policy "8021Xpolicy"
```

```
                        next
                        edit port6
                                set port-security-policy "8021Xpolicy"
                        next
                end
        next
end
```

3. On the client device, generate EAP traffic to trigger 802.1X authentication.
4. Verify that the 802.1X authentication is successful.

```
FGT_A (vdom1) (Interim)# diagnose switch-controller switch-info 802.1X
Managed Switch : S624FNTV23000204
Managed Switch : S248EPTF18001384

       port5 : Mode: port-based (mac-by-pass disable)
               Link: Link down
               Port State: unauthorized

       port6 : Mode: port-based (mac-by-pass disable)
               Link: Link up
               Port State: authorized: (  )
               Dynamic Authorized Vlan : 0
               Dynamic Allowed Vlan list:
               Dynamic Untagged Vlan list:
               EAP pass-through : Disable
               EAP egress-frame-tagged : Disable
               EAP auto-untagged-vlans : Enable
               Allow MAC Move : Enable
               Dynamic Access Control List : Disable
               Quarantine VLAN (4093) detection : Enable
               Native Vlan : 1
               Allowed Vlan list: 1,4093
               Untagged Vlan list: 4093
               Guest VLAN :
               Auth-Fail Vlan :
               AuthServer-Timeout Vlan :

               Sessions info:
               00:0c:29:d4:4f:3c     Type=802.1x,TTLS,state=AUTHENTICATED,etime=0,eap_cnt=6
params:reAuth=3600
               user="",security_grp="",radsec="disable":tunnel_id="0", fortinet_grp=""

Managed Switch : S248EPTF18001827
Managed Switch : S124EN5918003682
Managed Switch : S524DN4K16000116
```

5. Unplug the client device from Hub1 and then plug the client device into another port on Hub2, which connects to the same FortiSwitch unit.
6. On the client device, generate EAP traffic to trigger 802.1X authentication again.
7. Verify that the 802.1X authentication is successful on the new FortiSwitch port.

```
FGT_A (vdom1) (Interim)# diagnose switch-controller switch-info 802.1X
Managed Switch : S624FNTV23000204
Managed Switch : S248EPTF18001384

       port5 : Mode: port-based (mac-by-pass disable)
               Link: Link up
               Port State: authorized: (  )
               Dynamic Authorized Vlan : 0
               Dynamic Allowed Vlan list:
               Dynamic Untagged Vlan list:
               EAP pass-through : Disable
               EAP egress-frame-tagged : Disable
               EAP auto-untagged-vlans : Enable
               Allow MAC Move : Enable
               Dynamic Access Control List : Disable
               Quarantine VLAN (4093) detection : Enable
               Native Vlan : 1
               Allowed Vlan list: 1,4093
               Untagged Vlan list: 4093
               Guest VLAN :
               Auth-Fail Vlan :
               AuthServer-Timeout Vlan :

               Sessions info:
               00:0c:29:d4:4f:3c     Type=802.1x,TTLS,state=AUTHENTICATED,etime=6,eap_cnt=6
params:reAuth=3600
               user="",security_grp="",radsec="disable":tunnel_id="0", fortinet_grp=""

       port6 : Mode: port-based (mac-by-pass disable)
               Link: Link up
               Port State: unauthorized: (  )
               Dynamic Authorized Vlan : 0
               Dynamic Allowed Vlan list:
               Dynamic Untagged Vlan list:
               EAP pass-through : Disable
               EAP egress-frame-tagged : Disable
               EAP auto-untagged-vlans : Enable
               Allow MAC Move : Enable
               Dynamic Access Control List : Disable
               Quarantine VLAN (4093) detection : Enable
               Native Vlan : 1
               Allowed Vlan list: 1,4093
               Untagged Vlan list: 4093
               Guest VLAN :
               Auth-Fail Vlan :
               AuthServer-Timeout Vlan :

               Sessions info:
               00:09:0f:06:06:06     Type=802.1x,,state=AUTHENTICATING,etime=0,eap_cnt=0 params:reAuth=3600
               user="",security_grp="",radsec="disable":tunnel_id="0", fortinet_grp=""

Managed Switch : S248EPTF18001827
```

```
Managed Switch : S124EN5918003682
Managed Switch : S524DN4K16000116
```

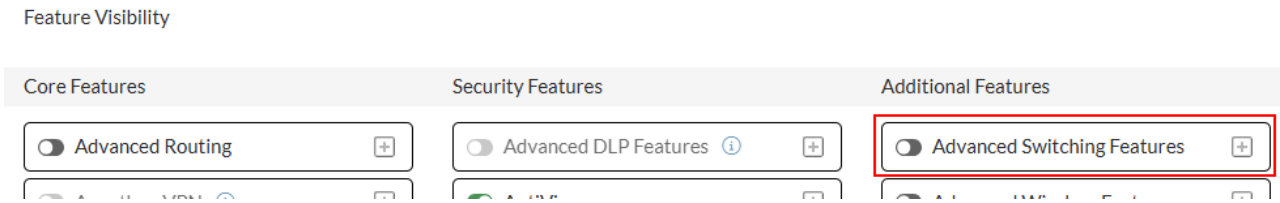
Enable advanced switching features in the GUI

The new toggle button, *Advanced Switching Features*, allows you to enable the following advanced switching features:

- On the *WiFi & Switch Controller* > *FortiSwitch Port Policies* page:
  - LLDP tab
  - QoS tab
  - VLAN policies tab
  - PTP tab
- On the *WiFi & Switch Controller* > *Managed FortiSwitches* page:
  - *MC-LAG Peer* column
  - *Config Sync Status* column
- On the *Diagnostics and Tools* pane:
  - *Config Sync Status* field (under the switch serial number)
- On the *WiFi & Switch Controller* > *FortiSwitch Ports* page:
  - *PTP* column (values are shown only when the FortiSwitch unit supports PTP)
  - *PTP Interface Policy* column (values are shown only when the FortiSwitch unit supports PTP)

To enable the advanced switching features:

1. Go to *System* > *Feature Visibility*.
2. Under *Additional Features*, enable *Advanced Switching Features*.



3. Click *Apply*.

To use the *LLDP* tab:

1. Go to *WiFi & Switch Controller* > *FortiSwitch Port Policies*.
2. Click the *LLDP* tab.
3. Under *Profiles*, create or edit an LLDP profile:
   - Click *Create new* to create a new LLDP profile and then click *OK* to save it.
   - Select the checkbox to the left of an already created LLDP profile name to edit, clone, or delete the profile.
4. Under *Settings*, configure the LLDP settings and then click *OK* to save them.

For more details about using LLDP, see Configuring LLDP-MED settings.

To use the *QoS* tab:

1. Go to *WiFi & Switch Controller* > *FortiSwitch Port Policies*.
2. Click the *QoS* tab.
3. Under *802.1p maps*, click *Create new*, configure a new 802.1p map, and then click *OK*.
4. Under *DSCP/IP precedence maps*, click *Create new*, configure a new DSCP/IP precedence map, and then click *OK*.
5. Under *Egress policies*, click *Create new*, configure a new QoS egress queue policy, and then click *OK*.
6. Under *QoS policies*, click *Create new* to configure the overall policy that will be applied to the switch ports.
   - From the *Egress queue profile* dropdown list, select the QoS egress queue policy you created in step 5.
   - Enable *Trust DSCP/IP precedence* and select the DSCP/IP precedence map you created in step 4.
   - Enable *Trust Dot1p* and select the 802.1p map you created in step 3.
7. Go to *WiFi & Switch Controller* > *FortiSwitch Ports*, select the checkbox for the port that you want to assign the QoS policy to, click the edit icon (looks like a pencil and paper) in the *QoS Policy* column, select the QoS policy you created in step 6, and click *Apply*.

For more details about using QoS, see Configuring QoS with managed FortiSwitch units.

To use the *VLAN policies* tab:

1. Go to *WiFi & Switch Controller* > *FortiSwitch Port Policies*.
2. Click the *VLAN policies* tab.
3. Click *Create new*.
4. In the *Name* field, enter a unique name.
5. [Optional] In the *Description* field, enter a description of the VLAN policy.
6. In the *FortiLink* dropdown list, select which FortiLink interface this VLAN policy applies to.
7. In the *Native VLAN* dropdown list, select the default VLAN for untagged incoming frames.
8. In the *Allowed VLANs* dropdown list, select which VLANs the port can transmit frames to or receive frames from.
9. In the *Untagged VLANs* dropdown list, select which VLANs the port will transmit frames to without the VLAN tag.

   Any VLAN in the untagged VLAN list must also be a member of the allowed VLAN list.
10. If you want to discard tagged or untagged frames, enable *Discard frames* and then select *Untagged frames* or *Tagged frames*.
11. Click *OK* to save the VLAN policy.

For more details about using VLANs, see Configuring VLANs.

To use the *PTP* tab:

1. Go to *WiFi & Switch Controller* > *FortiSwitch Port Policies*.
2. Click the *PTP* tab.
3. Under *Profiles*, click *Create new*, configure a new PTP profile, and then click *OK*.
4. Configure the PTP settings in the CLI.

   By default, PTP is disabled. Enable PTP and select which PTP profile will use these PTP settings. The default profile is automatically selected. If you have multiple PTP profiles, each managed switch can use a different PTP profile.

5. Under *Interface policies*, click *Create new*, configure a new PTP interface policy, and then click *OK*.
6. Go to *WiFi & Switch Controller* > *FortiSwitch Ports*, select the checkbox for the port that you want to assign the PTP interface policy to, click the edit icon (looks like a pencil and paper) in the *PTP Interface Policy* column, select the PTP interface policy you created in step 4, and click *Apply Changes*.

For more details about using PTP, see Configuring PTP transparent-clock mode.

Control the number of 802.1X-authenticated clients per port

You can now limit the number of 802.1X-authenticated sessions allowed on a port. Limiting the number of devices or PCs per port helps increase the security of the network. The default number of sessions allowed per port is 20, and you can configure 2-20 sessions per port.

This feature requires MAC-based 802.1x authentication (`set security-mode 802.1X-mac-based`).

To configure the maximum number of 802.1X-authenticated sessions per port:

```
config switch-controller security-policy 802-1X
   edit <policy_name>
      set security-mode 802.1X-mac-based
      set client-limit <2-20>
      set user-group <user_group_name>
   next
end
```

For example:

```
config switch-controller security-policy 802-1X
   edit "8021Xpolicy"
      set security-mode 802.1X-mac-based
      set client-limit 2
      set user-group "1X_RADIUS_GROUP"
   next
end
```

Support IGMP-snooping static group configuration

You can now define static groups for particular multicast addresses in a VLAN that has IGMP snooping enabled. You can specify multiple ports in the static group, separated by a space. The trunk interface can also be included in a static group.

You need to enable IGMP snooping for the VLAN interface before creating the IGMP-snooping static group for that VLAN.

Use the `set ignore-reports enable` command to configure an IGMP static group to ignore requests from other ports to become members. Preventing other ports from joining means that administrators control which ports receive traffic. This option is disabled by default, which allows other ports to dynamically join.

To configure an IGMP-snooping static group:

```
config switch-controller igmp-snooping-static-group
       edit <IGMP-snooping static group name>
        set description <IGMP-snooping static group description>
        set mcast-addr <IGMP-snooping static group multicast IP address>
        set vlan <VLAN name>
        set ignore-reports {enable | disable}
        config ports
               edit <identifier>
                       set switch-id <FortiSwitch_serial_number>
                       set ports <one or more port names>
               next
        end
       next
end
```

For example:

```
config switch-controller igmp-snooping-static-group
       edit "1"
               set description "static group for S424EFTF19000003"
               set mcast-addr 238.2.5.8
               set vlan "vlan22"
               set ignore-reports enable
               config ports
                       edit 1
                               set switch-id "S424EFTF19000003"
                               set ports "port6"
                       next
               end
       next
end
```

Dynamic MAB session handling

You can now configure in the CLI how long MAC authentication bypass (MAB) sessions are kept:

- In static mode, MAB sessions are kept until the link goes down or the MAB sessions are manually deleted with the CLI. Static mode is the default.
- In dynamic mode, MAB sessions are treated the same way as dynamically learned MAC addresses.

  In dynamic mode, you specify how long MAB sessions are kept with the `set mac-aging-interval <10-1,000,000 seconds>` command (under `config switch-controller global`). By default, the `mac-aging-interval` is set to 300 seconds.

> ⚠ Whenever you change the `mac-aging-interval` or `mab-entry-as` setting, all MAB sessions are deleted, but the EAP sessions are unchanged.

This feature is supported on all FortiSwitch models.

To configure how long MAB sessions are kept:

```
config switch-controller 802-1X-settings
   set mab-entry-as {dynamic | static}
end
```

Enhance NAC/DPP port control and QoS/PoE support

Several enhancements have been made to the FortiSwitch network access control (NAC) and dynamic port policy (DPP). These changes improve port security and provide flexibility in NAC deployments.

- You can now use the CLI to specify how many hours that the NAC policies and DPPs keep matched devices. Previously, you could only specify the number of days to keep matched devices. The maximum amount of time is still 120 days (3,072 hours). Set the `match-period` to 0 to always keep the matched devices.

  To set the match period in the NAC policy:

  ```
  config user nac-policy
         edit <policy_name>
                 set match-type override
                 set match-period <0-3072 hours>
         next
  end
  ```

  For example:

  ```
  config user nac-policy
         edit NACpolicy1
                 set match-type override
                 set match-period 3000
         next
  end
  ```

  To set the match period in the DPP:

  ```
  config switch-controller dynamic-port-policy
         edit <dynamic_port_policy_name>
                 set fortilink <FortiLink_interface_name>
                 config policy
                         edit <policy_name>
                                 set match-type override
                                 set match-period <0-3072 hours>
                         next
  ```

```
                end
        next
end
```

For example:

```
config switch-controller dynamic-port-policy
        edit DPP1
                set fortilink port11
                config policy
                        edit 1
                                set match-type override
                                set match-period 2048
                        next
                end
        next
end
```

- When NAC is enabled on a managed switch port, you can now configure the Power over Ethernet (PoE) settings.

  For example:

```
config switch-controller managed-switch
        edit S248EPTF18001384
                config ports
                        edit port6
                                set access-mode nac
                        next
                end
        next
end
config switch-controller managed-switch
        edit S248EPTF18001384
                config ports
                        edit port6
                                set poe-pre-standard-detection enable
                                set poe-max-power-mode 30W
                                set poe-port-priority high-priority
                                set poe-status disable
                        next
                end
        next
end
```

- You can now specify a QoS policy that is applied if the device that matches the NAC policy is the only device on the port.

  To specify a QoS policy for a port using NAC:

```
config user nac-policy
        edit <policy_name>
```

```
                set port-setting-override enable
                set qos-policy <name of QoS policy>
        next
end
```

For example:

```
config user nac-policy
        edit "38731"
                set mac "00:0c:29:d4:4f:3c"
                set switch-fortilink "port11"
                set switch-mac-policy "38731"
                set port-setting-override enable
                set qos-policy "38731-QoS"
        next
end
```

Add global port-selection criteria for some switch models

For the FS-6xxF, FSR-216F-POE, FSR-108F, and FSR-112F-POE models, you now configure the port-selection criteria at the global switch-controller level. For all other FortiSwitch models, the port-selection criteria is configured at the trunk level.

For the FS-6xxF, FSR-216F-POE, FSR-108F, and FSR-112F-POE models:

```
config switch-controller managed-switch
    edit <FortiSwitch_serial_number>
        set port-selection-criteria {src-mac | dst-mac | src-dst-mac | src-ip | dst-ip | src-dst-
ip}
    next
end
```

Option	Description
`port-selection-criteria {src-mac | dst-mac | src-dst-mac | src-ip | dst-ip | src-dst-ip}`	Algorithm for aggregate port selection: • `src-mac`: Source MAC address. • `dst-mac`: Destination MAC address. • `src-dst-mac`: Source and destination MAC address. • `dst-mac`: Destination MAC address. • `src-ip`: Source IP address. • `dst-ip`: Destination IP address. • `src-dst-ip`: Source and destination IP address. This is the default setting.

For all other FortiSwitch models:

```
config switch-controller managed-switch
    edit <FortiSwitch_serial_number>
        config ports
            edit <trunk_name>
                set port-selection-criteria {src-mac | dst-mac | src-dst-mac | src-ip | dst-ip |
src-dst-ip}
            next
        end
    next
end
```

Using NAC with 802.1X authentication in the FortiOS GUI

The FortiOS GUI now supports using both FortiSwitch network access control (NAC) and 802.1X authentication on the same switch port. Previously, this feature was supported only in the FortiOS CLI.

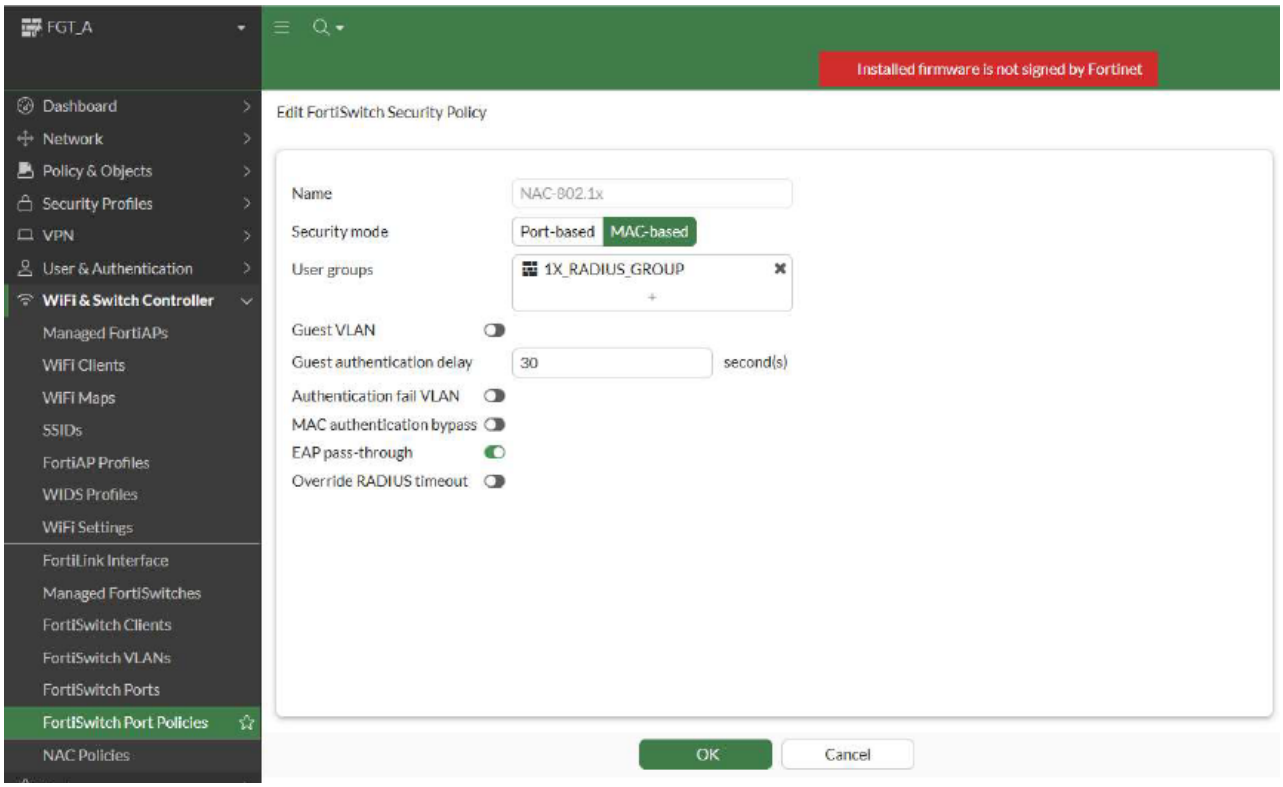
To use NAC with 802.1 authentication, you must specify 802.1X MAC-based authentication. 802.1X port-based authentication is not supported.

The following are the prerequisites for this feature:

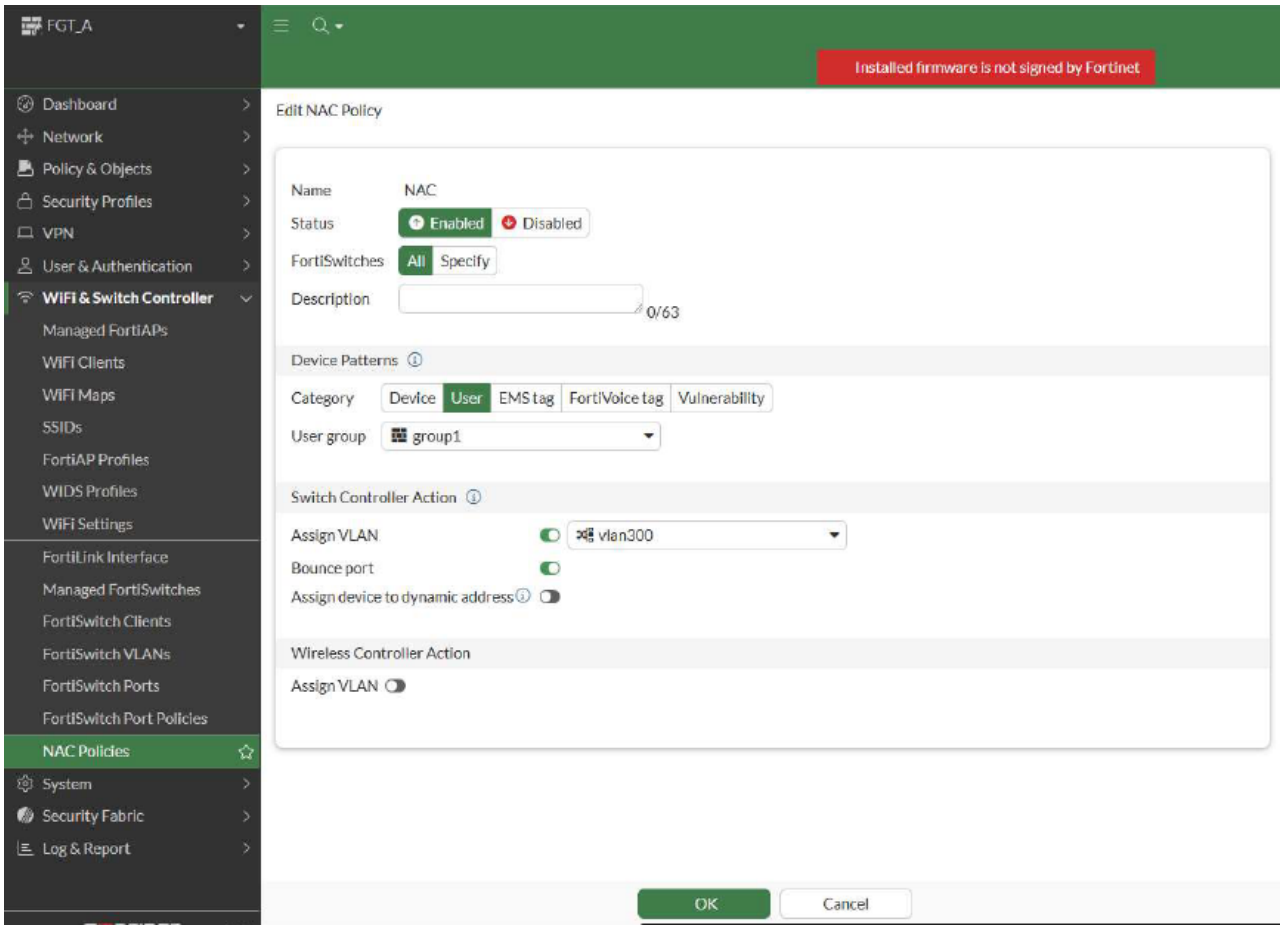
- The RADIUS server must return the Fortinet-Group-Name RADIUS attribute with the user group information.
- The FortiGate device must have a user group matching the Fortinet-Group-Name RADIUS attribute, and the RADIUS server can be added as a member to the user group.

Using the GUI:

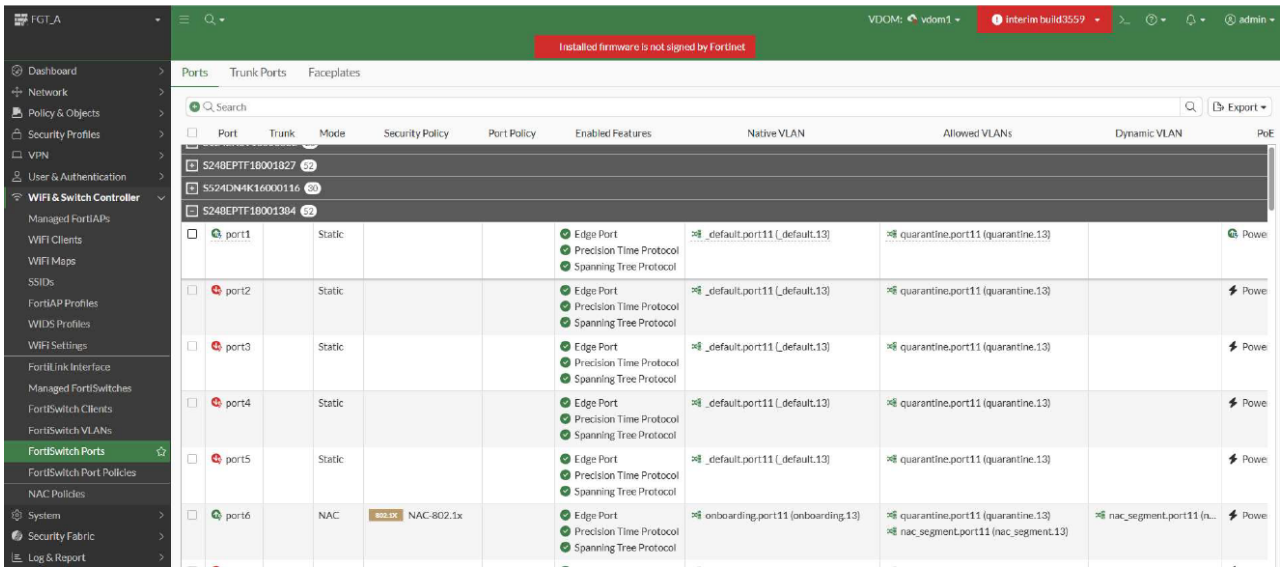
1. Configure an 802.1X MAC-based security policy.



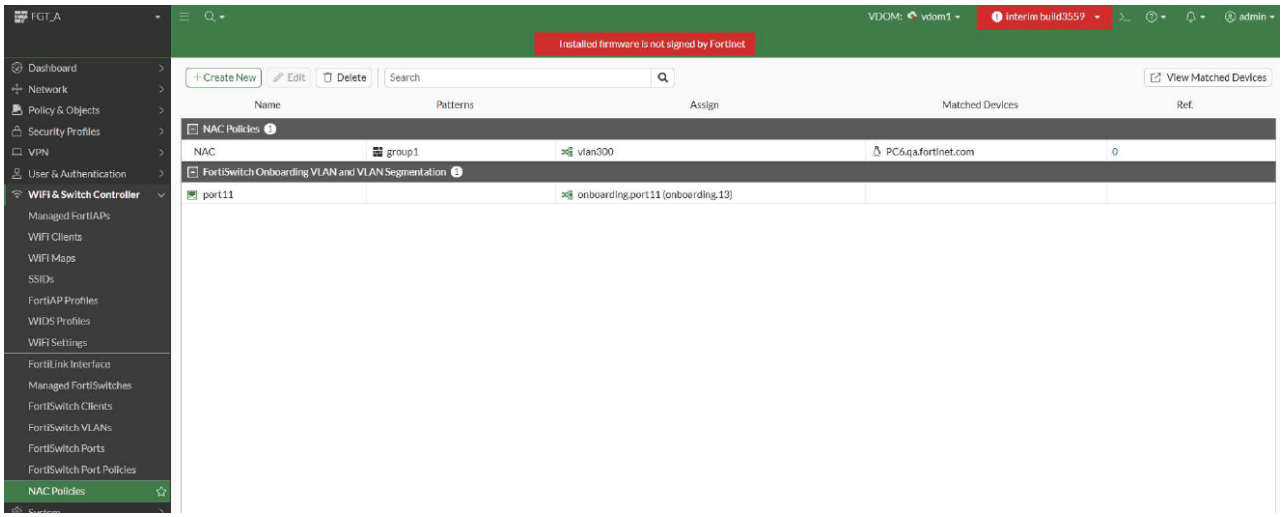
2. Configure a NAC user policy on the same user group.



3. Apply the NAC user policy and 802.1X security policy to the same FortiSwitch port.



4. Perform 802.1X authentication on the client device.
5. After thesuccessful authentication, go to the *WiFi & Switch Controller > NAC Policies* page to check that the client device was matched by the NAC policy.


FGT_A
VDOM: vdom1
Interim build3559
admin
Installed firmware is not signed by Fortinet
Dashboard
Network
Policy & Objects
Security Profiles
VPN
User & Authentication
WiFi & Switch Controller
Managed FortiAPs
WiFi Clients
WiFi Maps
SSIDs
FortiAP Profiles
WIDS Profiles
WiFi Settings
FortiLink Interface
Managed FortiSwitches
FortiSwitch Clients
FortiSwitch VLANs
FortiSwitch Ports
FortiSwitch Port Policies
NAC Policies
Create New
Edit
Delete
Search
View Matched Devices
Name
Patterns
Assign
Matched Devices
Ref.
NAC Policies 1
NAC
group1
vlan300
PC6.qa.fortinet.com
0
FortiSwitch Onboarding VLAN and VLAN Segmentation 1
port11
onboarding.port11 (onboarding,13)

System

This section includes information about system related new features:

- General on page 431
- High availability on page 467
- SNMP on page 492
- FortiGuard on page 494
- API on page 506

General

This section includes information about general system related new features:

- Configurable timeout for log file system check on page 431
- Administrator access hardening on page 433
- SCP file transfers on page 434
- VM license grace period monitoring enhancement on page 437
- Configurable FQDN Host header for FortiGate proxy communication on page 438
- Additional commands supported before device registration on page 440
- Legal third party disclosure panel on page 440
- Per VDOM replacement message customization on page 442
- Configuration revisions and logout backup default changes on page 444
- FortiAI assistant and CLI Code Lab on page 446
- Enhancements to required upgrade when firmware license is invalid or device is EOES on page 458

Configurable timeout for log file system check

FortiGate now supports a configurable timeout for log file system check:

```
config system global
    set log-fsck-timeout <integer>
end
```

Option	Description
`log-fsck-timeout <integer>`	Configure the maximum the number of seconds the FortiGate unit waits while the file system check is in progress before allowing the boot process to complete and the system fully is operational. Zero seconds means the FortiGate unit waits until the file system check is complete (0 - 3600, default = 300).

When an administrator starts a log file system check after an unexpected shutdown, FortiGate reboots and starts the scan during bootup. If the timeout period expires before the scan completes, FortiGate continues booting up. After successfully booting up, the scan runs in the background, and administrators can log in to FortiOS.

Scope and limitations

While background disk scanning is in progress, the following features are affected because the disk is unavailable:

- Log feature: no logs are recorded to disk. Logs in memory can be viewed, and logs can be forwarded to remote devices, such as FortiAnalyzer and FortiGate Cloud.
- IPS feature: the IPS engine does not launch.
- Web filter flow mode: Web filter is paused because of no IPSE worker in the system. FortiGate bypasses all the URL access.
- SCIM feature: no data from SCIM client is received.
- Federated upgrade feature (upgrading all devices): Federated upgrade is suspended.

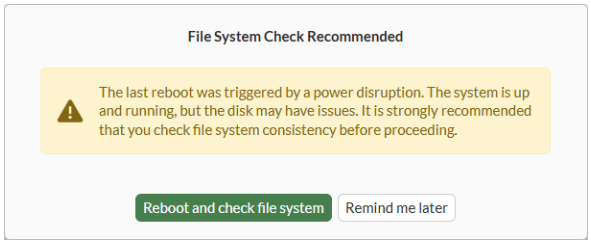
The features are no longer affected after the disk scanning completes.

Example

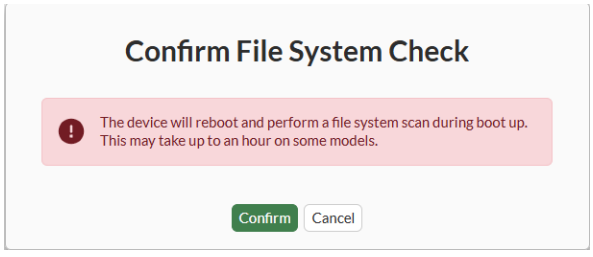
In this example, an administrator selects *Reboot and check file system* in the GUI after a power outage. FortiGate reboots and initiates scan during bootup. However, the configured timeout (`log-fsck-timeout`) expires before the scan completes. As a result, FortiGate continues booting up while performing the scan in the background, allowing administrators to log in to the GUI and perform most tasks. The GUI displays the file system scan progress in the banner.

To initiate a file system check in the GUI:

1. After a power outage, the following message is displayed:



2. Click *Reboot and check file system*.
   A confirmation dialog box is displayed:



3. Click *Confirm* to start the file system scan.

After rebooting, FortiGate starts the file system check and attempts to finish the scan during bootup.

However the configured timeout expires before the scan completes. FortiGate continues booting up and continues the scan in the background.

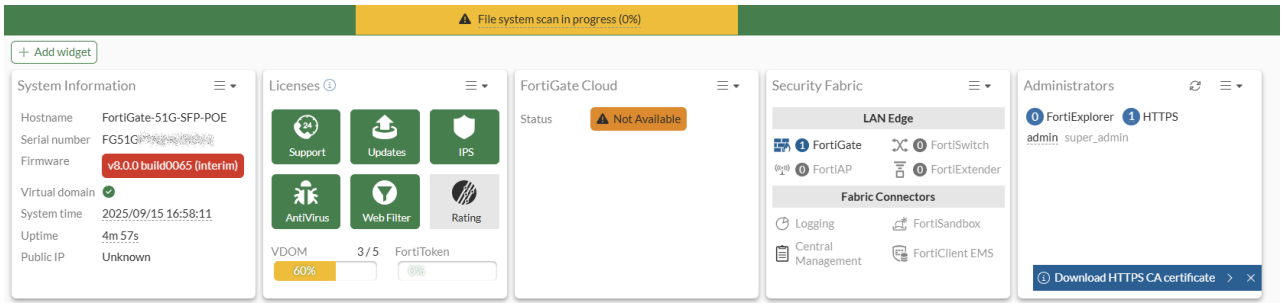
```
Booting OS...
Initializing firewall...

System is starting...
System file integrity init check failed!
Scanning /dev/sda1 timeout.


FortiGate-101F login: Background scanning on disk.
```

A log is added:

```
1: date=2025-08-14 time=17:15:33 devname="FortiGate-101F" devid="FG101FTK20000146"
eventtime=1755216932098384640 tz="-0700" logid="0100020212" type="event" subtype="system"
level="alert" vd="root" logdesc="Disk scan is needed" msg="Automatical disk scan for /dev/sda1
was timeout. It will be executed in background after system is started!"
```

4. While the scan runs in the background, administrators can log in to the GUI.

   A *File system scan in progress (<number>%)* message is displayed in the banner to report scan progress:



Administrator access hardening

The `disallowed-login-methods` configuration setting allows administrators to explicitly block specific access methods for logging into the FortiGate, such as console, GUI, SSH, or Telnet. Specifying the GUI access method will prohibit the ability of the administrator to log into the FortiGate through HTTP and HTTPS.

This feature introduces granular control over login channels, enhancing security by enabling organizations to disable unwanted or less secure access methods.

CLI syntax

The following command can be implemented to block login access through specific methods:

```
config sys admin
    edit <administrator>
        set disallowed-login-methods {console | gui | ssh | telnet}
```

```
    next
end
```

To enable administrator access hardening:

1. Configure the administrator and block access to SSH login attempts:

```
config sys admin
    edit "admin1"
        set disallowed-login-methods ssh
        set accprofile "prof_admin"
        set vdom "root"
        set password ENC xxxxxxxx
    next
end
```

2. Enter the current administrator password when prompted.
3. Attempt to log into the FortiGate by SSH as the administrator. The login attempt is blocked.
4. Review the logs:

```
# execute log display
9: date=2025-07-21 time=14:26:59 eventtime=1753133219677192147 tz="-0700" logid="0100032002"
type="event" subtype="system" level="alert" vd="root" logdesc="Admin login failed" sn="0"
user="admin1" ui="ssh(172.16.200.55)" method="ssh" srcip=172.16.200.55 dstip=172.16.200.1
action="login" status="failed" reason="none" msg="Administrator admin1 login failed from ssh
(172.16.200.55)"
```

SCP file transfers

SCP can be used for uploading and exporting configuration, firmware, license, and certificate files. This provides a more secure and flexible alternative to FTP and TFTP.

To backup or restore a configuration file using SCP:

```
execute {backup | restore} config scp <filename> <scp server>[:scp port] <user> <SCP password>
[password]
```

Option	Description
<filename>	Configure file name(path) on the remote SCP server.
<scp server>[:scp port]	SCP server address (IPv4, [IPv6], or FQDN - port optional). Use square brackets when specifying a port with an IPv6 address or IPv6-resolving FQDN.
<user>	SCP username.
<SCP password>	SCP password.
[password]	Password may be needed to restore/backup file.

For example:

```
# execute backup config scp 101f.conf 172.16.200.55 root sys@qa123456
Please wait...
Elapsed time: 0.797648100 seconds
Send config file to scp server via vdom root OK.
```

```
# execute restore config scp 101f.conf 172.16.200.55 root sys@qa123456
This operation will overwrite the current setting and could possibly reboot the system!
Do you want to continue? (y/n)y

Elapsed time: 0.628775620 seconds
File check OK.

FortiGate-101F (Interim)#
System is rebooting...


The system is going down NOW !!
```

To restore an image using SCP:

```
execute restore image scp <filename> <scp server>[:scp port] <user> <SCP password>
```

For example, to downgrade from version 8.0.0 to 7.6.3:

```
# execute restore  image  scp  FGT_101F-v7.6.3.F-build3510-FORTINET.out 172.16.200.55 root
sys@qa123456
This operation will replace the current firmware version and reboot the system!
Do you want to continue? (y/n)y

Elapsed time: 4.579736944 seconds
Verifying the signature of the firmware image.
Image verification OK!
Warning: Installing image v7.6.3 from v8.0.0 is not a recommended upgrade path. Continuing with
the upgrade may result in loss of configuration. Do you want to proceed?
Do you want to continue? (y/n)y

Warning: Downgrading to an image with Feature maturity notation.
Image file uploaded is marked as a Feature image, are you sure you want to downgrade?
Do you want to continue? (y/n)y

Please confirm again. Are you sure you want to downgrade using uploaded file?
Do you want to continue? (y/n)y

This operation will downgrade the current firmware version.
Do you want to continue? (y/n)y

Upgrade report is saved successfully.


Firmware downgrade in progress ...
Done.
```

```
The system is going down NOW !!
```

To import or export a certificate using SCP:

```
execute vpn certificate local import scp <scp server>[:scp port] <user> <SCP password> <filename>
<type> [password]
```

```
execute vpn certificate local export scp <name> <type> <filename> <scp server>[:scp port] <user>
<SCP password>
```

Options	
<name>	Local certificate name.
<type>	Certificate file type ('cer'|'p12'|'csr').
<filename>	File name on the SCP server.
<scp server>[:scp port]	SCP server address (IPv4, [IPv6], or FQDN - port optional). Use square brackets when specifying a port with an IPv6 address or IPv6-resolving FQDN.
<user>	SCP username.
<SCP password>	SCP password.
[password]	Password for PKCS12 file.

For example:

```
# execute vpn certificate local export scp FGT182-subca-san p12 /usr/local/share/backup/FGT182-
subca-san.p12 10.1.100.41 oliver test123
Elapsed time: 0.788176176 seconds
Done.
```

```
# execute vpn certificate local import scp 10.1.100.41 oliver test123
/usr/local/share/backup/FGT182-subca-san.p12 p12 123456
Elapsed time: 0.555388134 seconds
Done.
```

To upload a license using SCP:

```
execute restore manual-license scp <filename> <scp server>[:scp port] <user> <SCP password>
```

Option	Description
<filename>	Configure file name(path) on the remote SCP server.
<scp server>[:scp port]	SCP server address (IPv4, [IPv6], or FQDN - port optional).

Option	Description
	Use square brackets when specifying a port with an IPv6 address or IPv6-resolving FQDN.
<user>	SCP username.
<SCP password>	SCP password.

For example:

```
# execute restore  manual-license  scp  FG101FTK20000146Off-NetworkLicenseFile.lic 172.16.200.55
root sys@qa123456
Elapsed time: 0.540073900 seconds
FortiGate Manual License is updated
```

VM license grace period monitoring enhancement

FortiGate VM now displays `Last Check` and `Check Due` fields, allowing administrators to track remaining time in the license grace period before shutdown. This is especially valuable in air-gapped or DDIL (denied, degraded, intermittent, or limited) environments, where connectivity to FortiGuard or FortiManager may be limited, and the new fields help prevent unexpected service disruption.

After FortiGuard validates the VM license, the new fields are shown under `diagnose debug vm-print-license`:

```
# diagnose debug vm-print-license
SerialNumber: FGVMSLTM25007716
CreateDate: Fri Jun 13 23:25:44 2025
License expires: Mon Jun 15 17:00:00 2026
Last Check: Tue Sep 23 09:45:00 2025
Check Due: Thu Oct 23 09:45:00 2025
Key: yes
Cert: yes
Key2: yes
Cert2: yes
Signature: yes
Model: SL (18)
CPU: 4 (subscription:4)
MEM: 2147483647
VDOM license:
  permanent: 2
  subscription: 0
```

Option	Description
`Last Check`	Time of last successful license check.
`Check Due`	Maximum graceful time from last check (30 days)

Use the `get system status` command to view the license expiration date:

```
# get system status
Version: FortiGate-VM64-KVM v8.0.0,build0067,250918 (interim)
First GA patch build date: 240724
Current Security Level: Low
Firmware Signature: not-certified
Virus-DB: 1.00000(2018-04-09 18:07)
Extended DB: 1.00000(2018-04-09 18:07)
Extreme DB: 1.00000(2018-04-09 18:07)
AV AI/ML Model: 0.00000(2001-01-01 00:00)
IPS-DB: 6.00741(2015-12-01 02:30)
IPS-ETDB: 6.00741(2015-12-01 02:30)
IPS-MLDB: 0.00000(2001-01-01 00:00)
APP-DB: 6.00741(2015-12-01 02:30)
AIAP-DB: 0.00000(2001-01-01 00:00)
Proxy-IPS-DB: 6.00741(2015-12-01 02:30)
Proxy-IPS-ETDB: 6.00741(2015-12-01 02:30)
Proxy-APP-DB: 6.00741(2015-12-01 02:30)
FMWP-DB: 0.00000(2001-01-01 00:00)
IPS Malicious URL Database: 1.00001(2015-01-01 01:01)
IoT-Detect: 0.00000(2022-08-17 17:31)
OT-Detect-DB: 0.00000(2001-01-01 00:00)
OT-Patch-DB: 0.00000(2001-01-01 00:00)
OT-Threat-DB: 6.00741(2015-12-01 02:30)
IPS-Engine: 7.01157(2025-08-29 19:17)
Timezone DB Version: 1.003
Timezone DB IANA Version: 2024a
Serial-Number: FGVMSLTM25XXXXXX
License Status: Valid
License Expiration Date: 2026-06-15
VM Resources: 1 CPU/4 allowed, 3946 MB RAM
Log hard disk: Available
Hostname: FGVMSLTM25XXXXXX
```

Configurable FQDN Host header for FortiGate proxy communication

When FortiGate is behind an upstream proxy, FortiGuard communications requests to FortiGuard servers may be blocked unless explicit exceptions are configured.

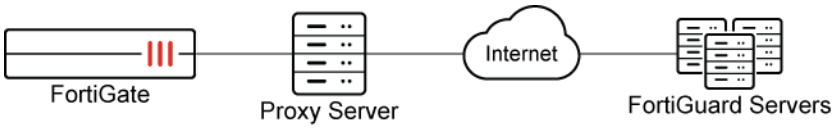
This feature adds a configurable option to prefer using the FortiGuard server FQDN as the `Host` header in `CONNECT` requests sent through the proxy. This allows administrators to create rules on the upstream proxy that match the FQDN, which is more stable and easier to manage than other identifiers, enabling FortiGuard traffic to pass through reliably.

```
config system fortiguard
    set proxy-fqdn-host {enable | disable}
end
```

Command	Description
`proxy-fqdn-host {enable | disable}`	Enable/disable preference of the FQDN as host header, if resolved by DNS, in the CONNECT request to FortiGuard servers.

Example

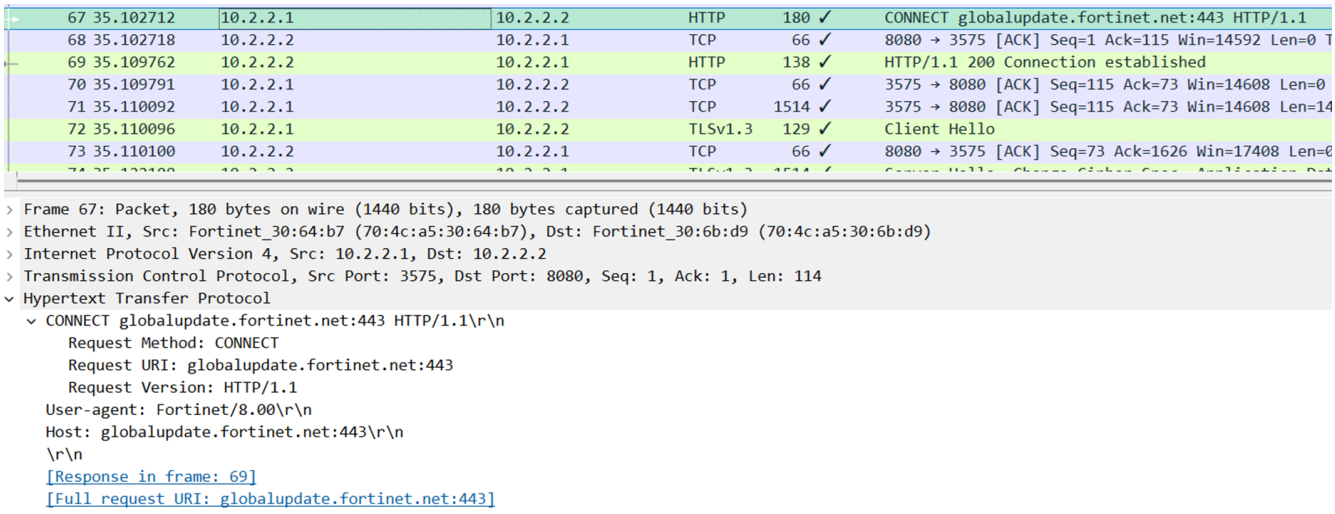
In this example, `proxy-fqdn-host` is enabled on the FortiGate and a proxy server is used to connect to the FortiGuard Distribution Network (FDN).



To configure a proxy server:

```
config system fortiguard
    set proxy-fqdn-host enable
    set proxy-server-ip <proxy_address>
    set proxy-server-port <proxy_port>
end
```

The output from the proxy server shows that the FortiGate sends a `CONNECT` request with the FortiGuard server FQDN in the `Host` header. This allows the upstream proxy to apply rules based on the FQDN, ensuring FortiGuard traffic passes through reliably.



The FQDN will not be included in the `Host` header if the FortiGate is configured to reach FortiGuard using unicast through a proxy server or when using FortiManager as a local FortiGuard server.

Additional commands supported before device registration

Additional commands are allowed before device registration to accommodate users that require configuring the device for central management, zero-touch provisioning (ZTP), and low-touch provisioning (LTP).

Commands added:

- `config firewall`
  - `policy`: Configure IPv4/IPv6 policies.
- `config router`
  - `setting`: Configure router settings.
  - `static`: Configure IPv4 static routing tables.
  - `static6`: Configure IPv6 static routing tables.
- `config system`
  - `admin`: Configure admin users.
  - `central-management`: Configure central management.
  - `dns`: Configure DNS.
  - `interface`: Configure interfaces.
  - `pppoe-interface`: Configure the PPPoE interfaces.
  - `settings`: Configure VDOM settings.

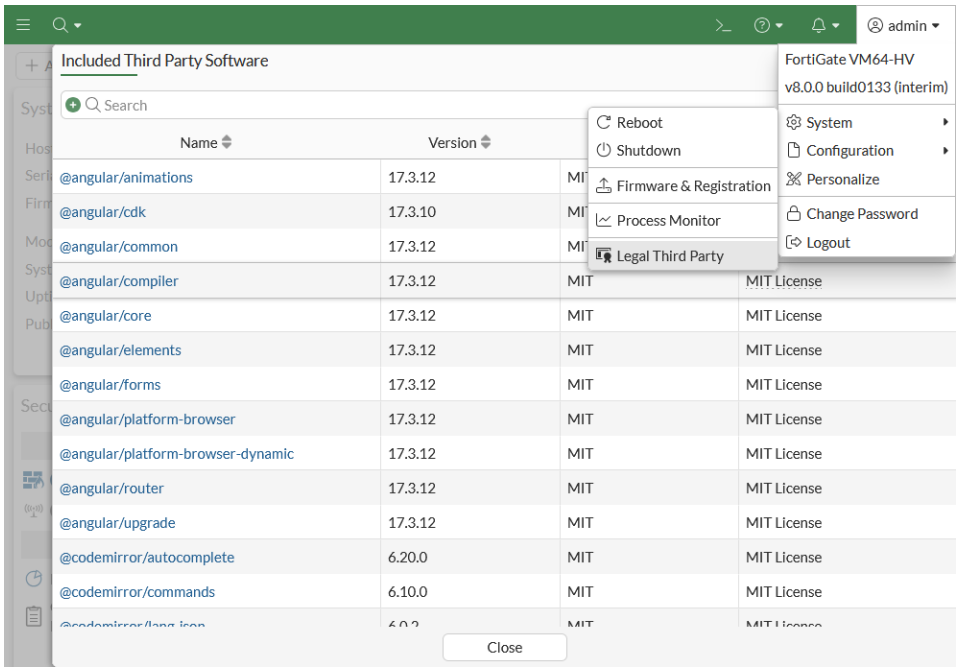
See also Enforce FortiCare registration with read-only CLI.

Legal third party disclosure panel

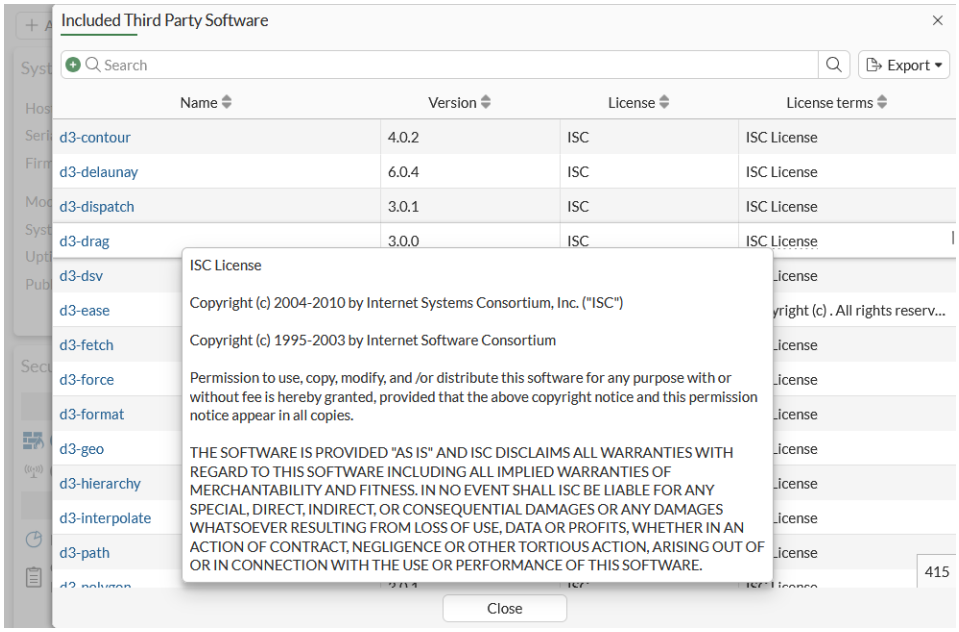
A *Legal Third Party* panel is added to the GUI, providing a searchable and exportable list of all third-party software used in FortiOS, along with their required licenses, license terms, and version information. This centralizes all third-party licensing details in one single, easily accessible location.

To access and use the third-party software list:

1. Select the *user name* > *System* > *Legal Third Party* to open the *Included Third Party Software* list.



2. Hover over the License terms field to display additional information about the license terms.

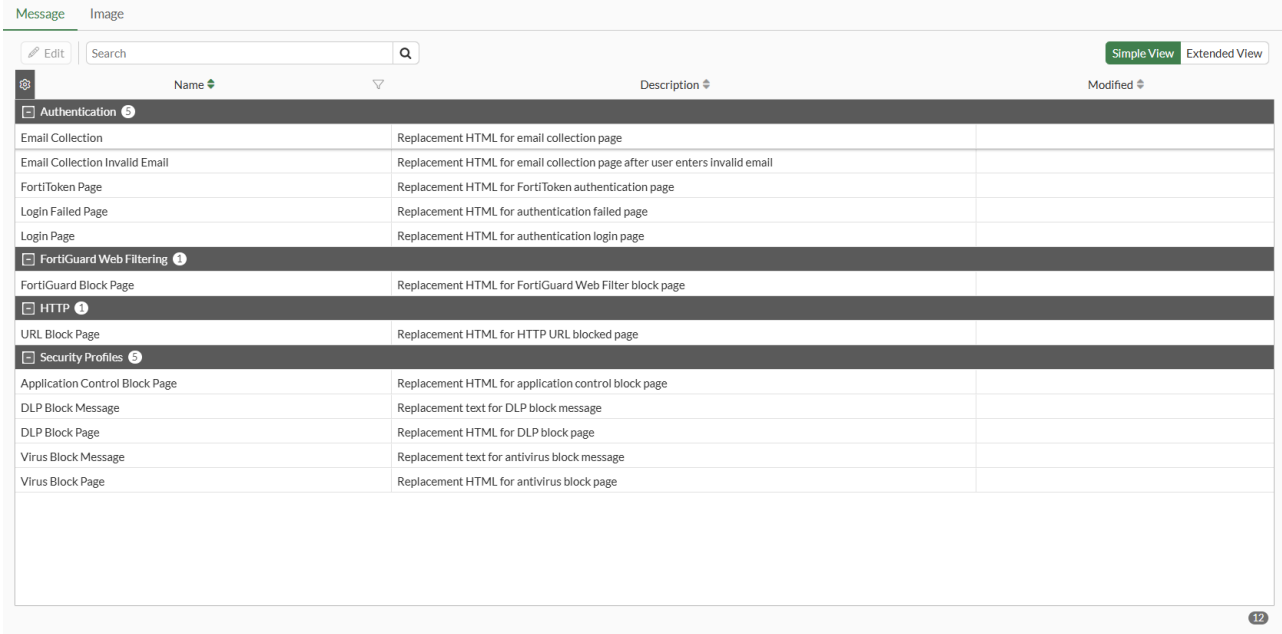


3. Click *Export* to export the list in CSV or JSON format.
4. Click on the third party software name to open that product's information page in your browser.

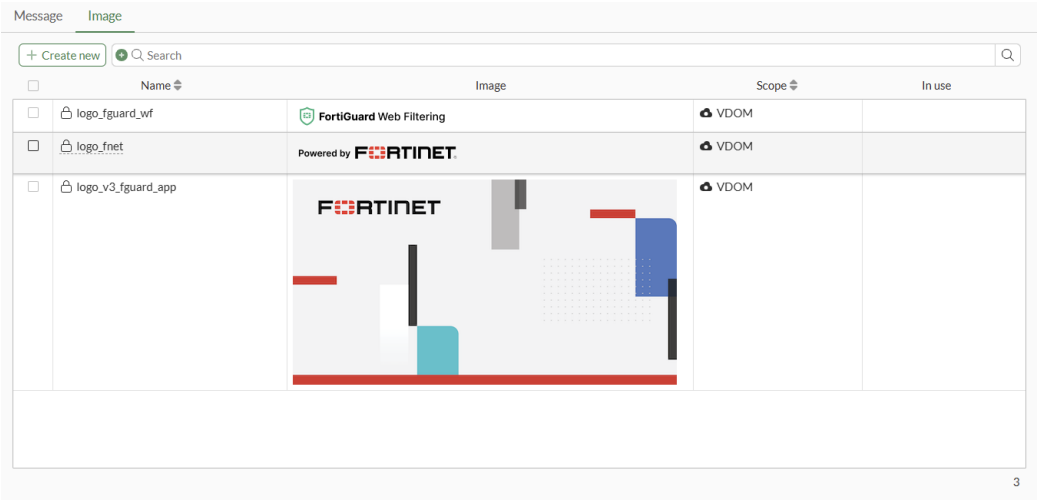
Per VDOM replacement message customization

Replacement messages and images are supported globally and per VDOM, allowing administrators to tailor message templates separately for each VDOM and improving control in multi-tenant deployments.

The replacement message GUI has been reorganized into two tabs: *Message* and *Image*.

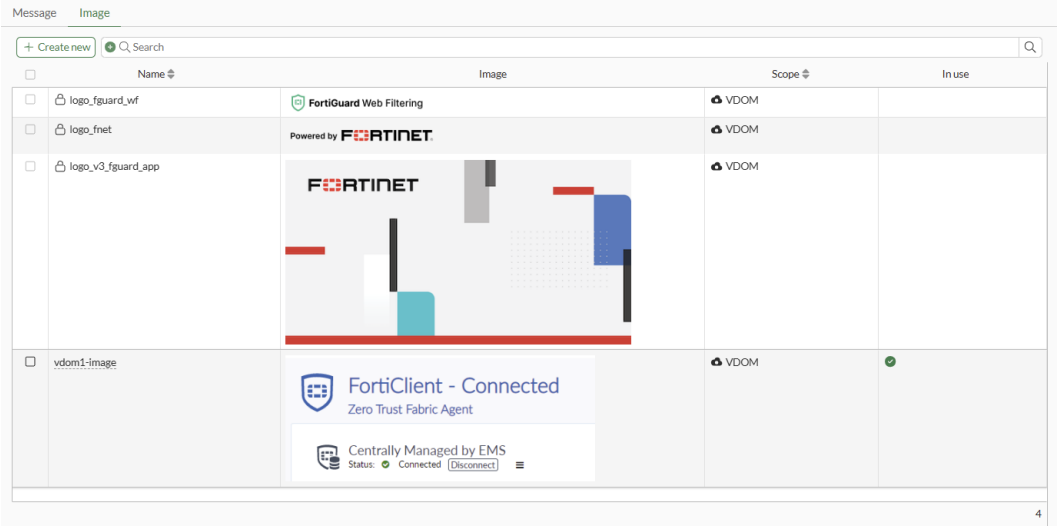


Replacement message images are available both globally and in individual VDOMs. When a new VDOM is created, a default *replacemsg-image* profile is automatically added.

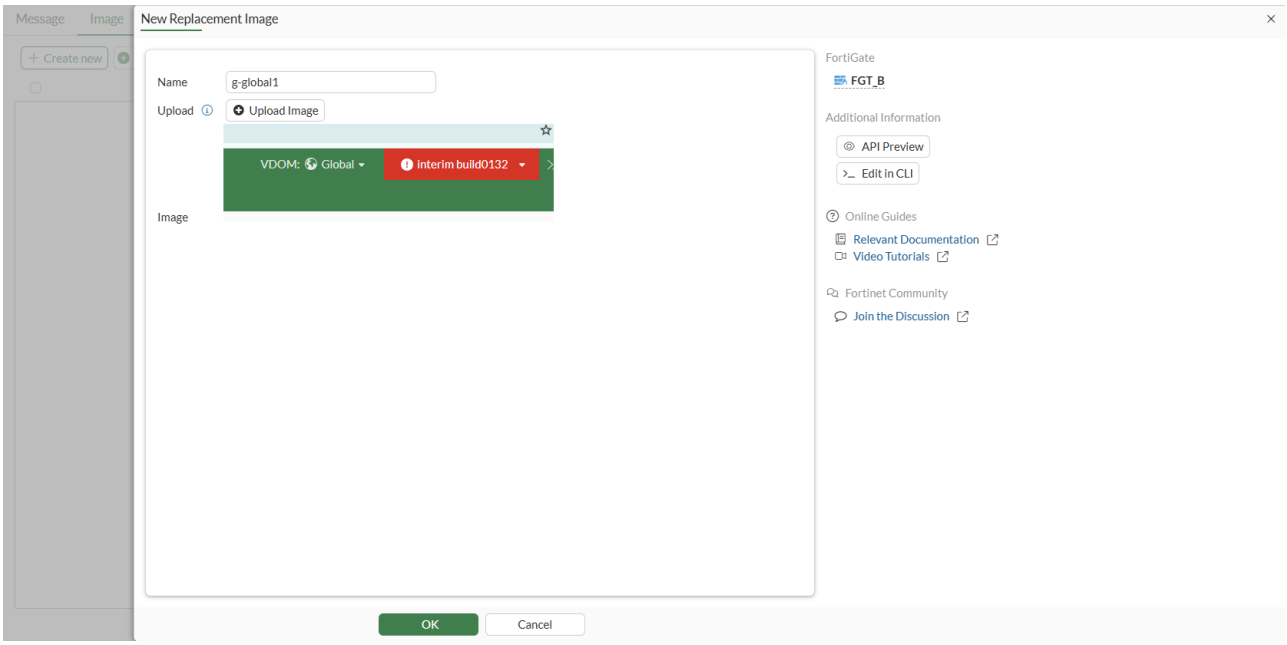


In the image list view, two columns have been added: *Scope* and *In Use*.

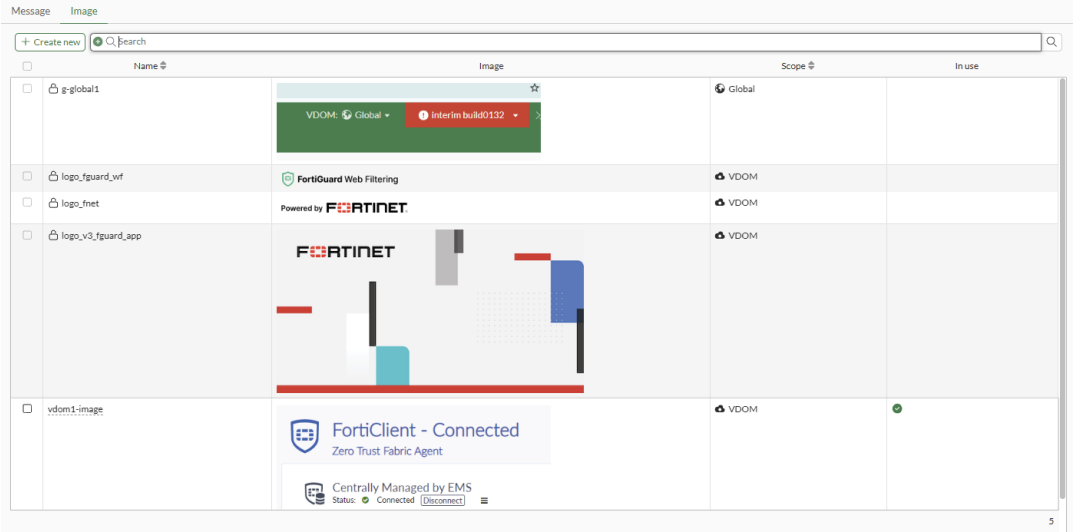
- *Scope* indicates whether an image belongs to the Global configuration or a specific VDOM.
- *In Use* shows whether the image is currently referenced by any replacement message.



Global-scope image names must begin with the *g-* prefix.



Global entries are read-only when viewed from within a VDOM. Objects inherited from Global are displayed with a lock icon, and the *Edit* and *Delete* buttons are disabled. This prevents accidental modification of Global configurations in a VDOM.



In the CLI, the `config system replacemsg-image` command is no longer Global only, and can be configured per VDOM.

```
config vdom
    edit vdom1
        config system replacemsg-image
            edit "newreplace-image"
                set image-type gif
                set image-base64 "R0lGODlhAQABAIAAAAAAAP///ywAAAAAQABAAACAUwAOw=="
            next
        end
    next
end
```

Configuration revisions and logout backup default changes

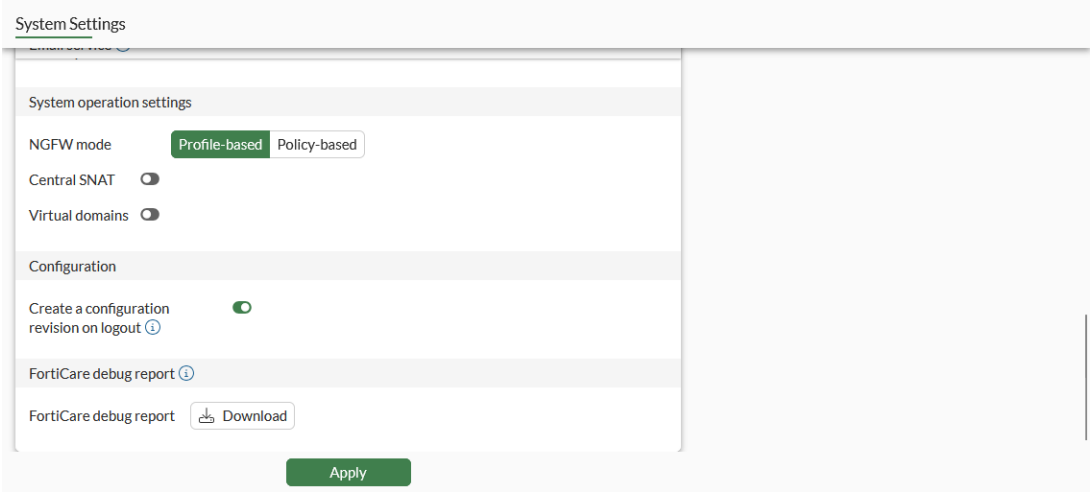
Configuration revisions are now only available on FortiGate models that have a hard disk. See Configuration revision for more details.

The `revision-backup-on-logout` option is enabled by default (on previous versions it was disabled).

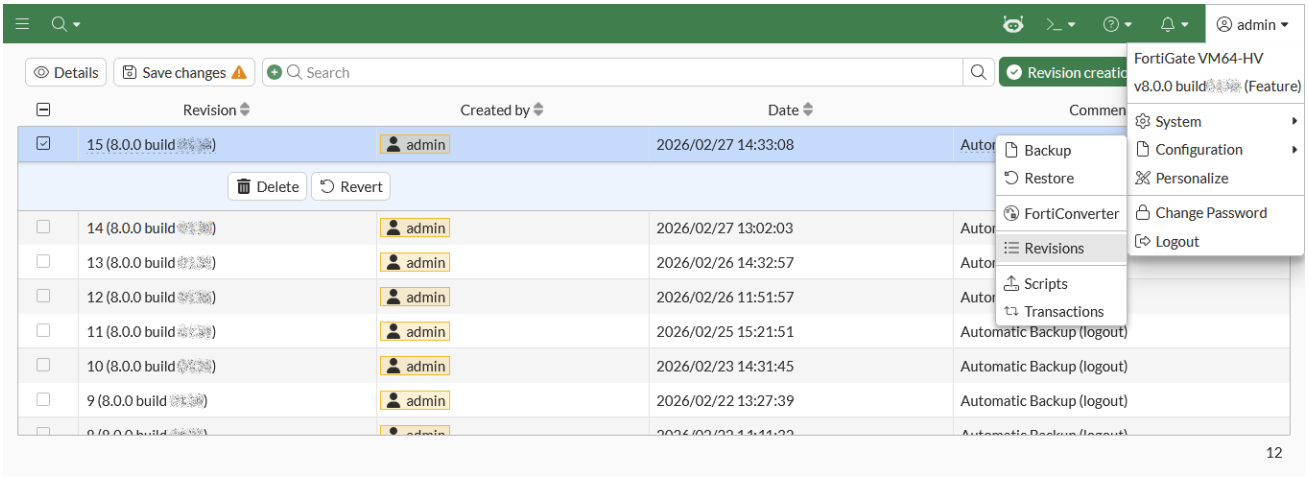
```
config system global
    set revision-backup-on-logout enable
end
```

A new GUI option called *Create a configuration revision on logout* is added under *System* > *Settings*, corresponding to the `revision-backup-on-logout` option in the CLI. This means that, by default, the configuration will automatically be saved when an administrator logs out.

Go to *System* > *Settings* and select the *General* tab to configure creating a revision automatically on logout on devices that have a hard disk

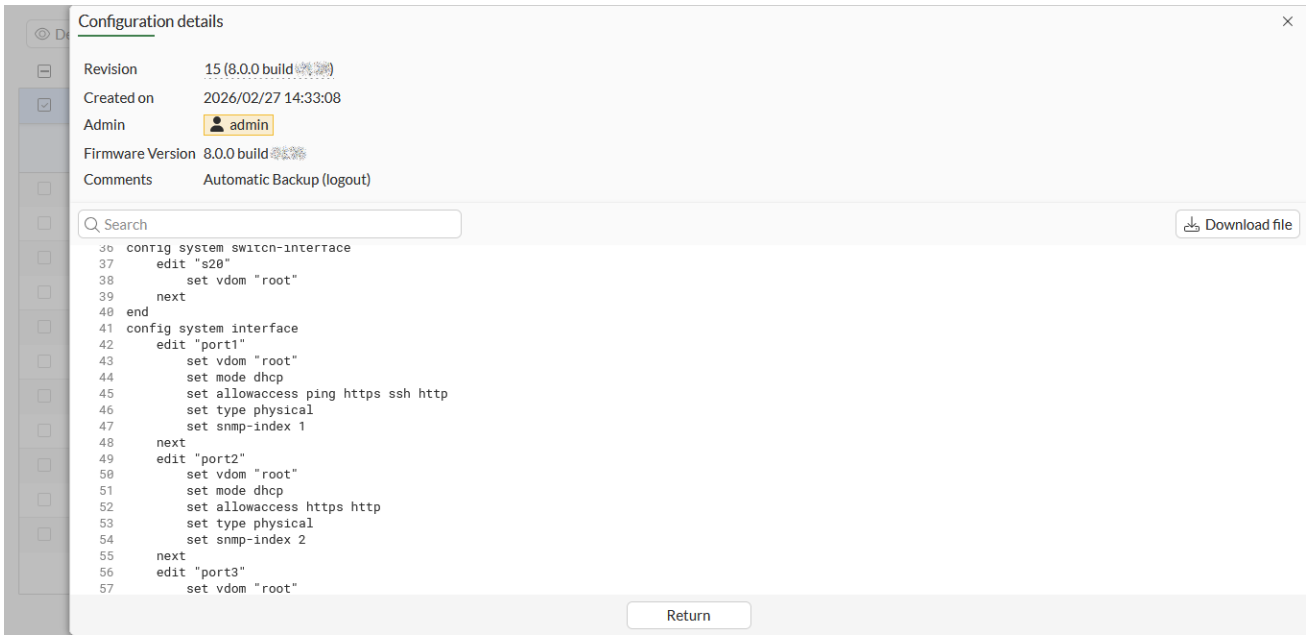


To view and manage revisions, open the administrator name drop-down and select *Configuration* > *Revisions*.



If changes have been made since the last revision was created, the *Save changes* button will show a warning icon.

Select a revision in the table to delete or revert to it. Click *Details* to show detailed information about that revision and download the configuration file.



On devices that do not have a disk, the *Revisions* page in the GUI and the option to create a configuration revisions automatically on logout are both not available. The configuration also cannot be backed up are restored from flash in the CLI, as the following commands are not available:

```
# execute backup config flash
# execute restore config flash
# execute revision delete
# execute revision list config
# execute upload config
```

FortiAI assistant and CLI Code Lab

This information is also available in the FortiOS 8.0 Administration Guide:

- FortiAI assistant and CLI Code Lab

FortiAI-Assist is now embedded in FortiOS, and FortiOS includes the FortiAI assistant and CLI Code Lab tool to provide RAG-enhanced documentation support, automated diagnostic analysis, and CLI script execution.

FortiAI can answer technical questions by using the FortiOS documentation and troubleshoot issues by reading logs directly from FortiGate or by analyzing debug outputs provided by administrators. The CLI Code Lab tool helps administrators use natural language prompts to generate and execute complex FortiGate configurations.

Administrators can choose between FortiAI and OpenAI as the AI provider for FortiAI-Assist. With FortiAI and the required subscription, a monthly allotment of tokens is available for use, and additional tokens can be purchased as needed. With OpenAI, administrators can configure FortiAI-Assist to work directly with OpenAI for billing and tokens.

Administrator profiles control administrator access to FortiAI-Assist.

This topic includes the following sections to introduce the FortiAI assistant and the CLI Code Lab tool:

- Requirements and tokens on page 447
- Getting started with FortiAI on page 448
- Example on page 454
- CLI syntax on page 457

Requirements and tokens

FortiAI-Assist is available on the following FortiGate models with internet access:

FortiGate model	Required subscription	Available tokens
FortiGate hardware with more than 2 GB RAM	FortiCare Premium support	Includes 2,000,000 starter tokens each month per device Additional tokens can be purchased and are shared among all FortiGate devices registered under the same FortiCare account.
FortiGate-VM S-series	FortiCare Premium support or one of the following: • Enterprise bundle • UTP bundle • ATP protection bundle	

FortiGate models with 2 GB RAM or less do not support FortiAI-Assist.

The available tokens and billing depend on the AI provider for FortiAI:

- With FortiAI as the AI provider, a monthly number of tokens is provided per device. Additional tokens can be purchased from Fortinet, and purchased tokens are shared among all FortiGate devices registered under the same FortiCare account. You can view token usage in FortiAI. See Checking token usage for FortiAI on page 453.
- With OpenAI as the AI provider, tokens are purchased from and billed to OpenAI. FortiAI does not provide any information about OpenAI token usage.

How token usage is calculated

Tokens are used in large language models (LLMs) to process text and quantify usage. Token usage is calculated using the following guidelines:

- When you use the FortiAI assistant, the text in both the prompt (input) and the response (output) is processed as tokens.
- While there is not a one-to-one relationship between words or characters and tokens, in general, more text in the query and response means using more tokens.
- Because the FortiAI assistant uses session history to inform its responses, queries that are a part of a long session will use more tokens than new conversations.

Best practices for managing token usage

To ensure you are using your monthly allocation of tokens effectively, consider implementing best practices for FortiAI users. For example:

- Make your prompts concise and specific. In terms of token usage, the prompt `Can you please help me create a firewall address for 10.0.0.1 and another one for the domain awesome-domain.com?` is less effective than `Create firewall addresses for 10.0.0.1 and awesome-domain.com`.
- Use filters in your prompts to receive concise and specific responses. For example, say that you want to create a site-to-site VPN based on an uploaded topology image.
- Use words that relate to functions existing in FortiOS. For example, using `quarantine device` concisely tells the FortiAI assistant what action is required.
- Reference details in the existing thread when possible. This reduces redundancy and allows you to be concise and specific as you build upon previous prompts. However, note that the FortiAI assistant will not remember previous threads.

Getting started with FortiAI

This section describes how to enable FortiAI with FortiAI or OpenAI as the AI provider. It also includes some basic procedures:

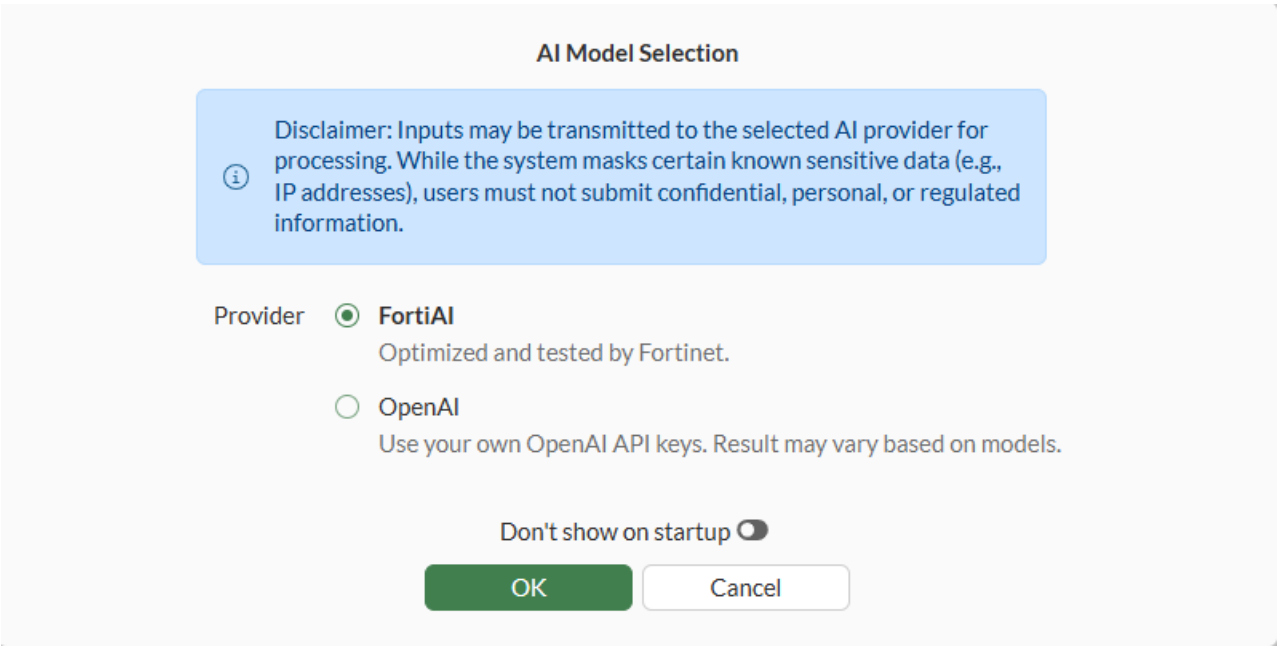
- Enabling FortiAI with FortiAI as AI provider on page 448
- Enabling FortiAI with OpenAI as AI provider on page 449
- Accessing CLI Code Lab on page 451
- Managing administrator access to FortiAI in administrator profiles on page 451
- Masking sensitive data in chat on page 452
- Accessing FortiAI chat history on page 452
- Setting chat message history on page 453
- Exporting FortiAI chats and threads on page 453
- Checking token usage for FortiAI on page 453

Enabling FortiAI with FortiAI as AI provider

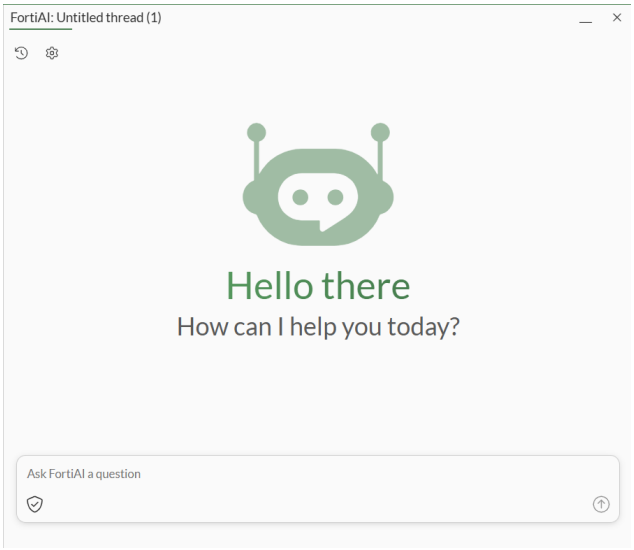
FortiAI can be enabled with FortiAI as the AI provider. This configuration includes a monthly allotment of tokens.

To enable FortiAI in the GUI:

1. In the banner, click the *FortiAI* icon. The *AI Model Selection* dialog box appears.



2. Select *FortiAI* and click *OK*. FortiAI is displayed on the right side of the GUI.

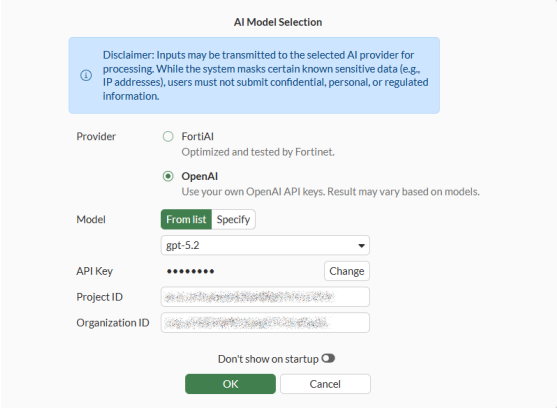


Enabling FortiAI with OpenAI as AI provider

FortiAI can be enabled with OpenAI as the AI provider. This option configures billing with OpenAI for tokens.

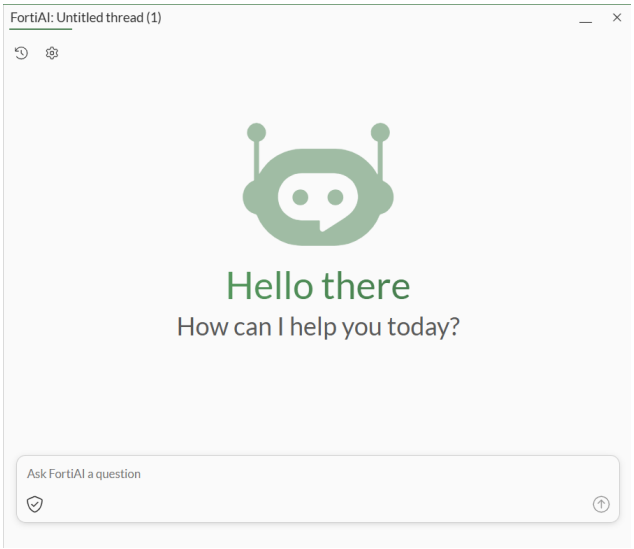
To configure OpenAI as AI provider in the GUI:

1. In the banner, click *FortiAI*. The *AI Model Selection* dialog box appears.
2. Select *OpenAI* and complete the options.



3. Click *OK*. FortiAI is displayed on the right side of the GUI powered by OpenAI instead of FortiAI.

> The GUI always displays FortiAI regardless of the selected LLM.



To configure OpenAI as AI provider in the CLI:

```
config system admin
    edit "admin"
        set vdom "root"
        set accprofile "super_admin"
        set gui-llm-provider openai
        set openai-api-key xxx
        set openai-api-key-part2 yyy
        set openai-model "gpt-5.2"
        set openai-project-id "<project ID>"
        set openai-org-id "<organization ID>"
    next
end
```

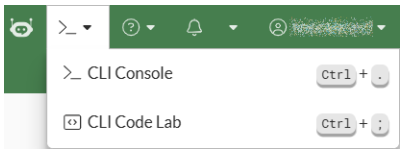
Accessing CLI Code Lab

The CLI Code Lab is available from the FortiOS GUI banner and from FortiAI, and it can be used to generate, edit, and execute CLI commands for configuring FortiGate.

When FortiAI uses FortiAI as the AI provider, the following FortiOS documents are used to provide answers: FortiOS Administration Guide, FortiOS CLI Reference. FortiAI also checks its answers against the FortiGate to ensure the results are suitable for the specific model.

To access CLI Code Lab:

1. Use one of the following methods:
   - From the FortiOS GUI banner, click *CLI Console > CLI Code Lab*.

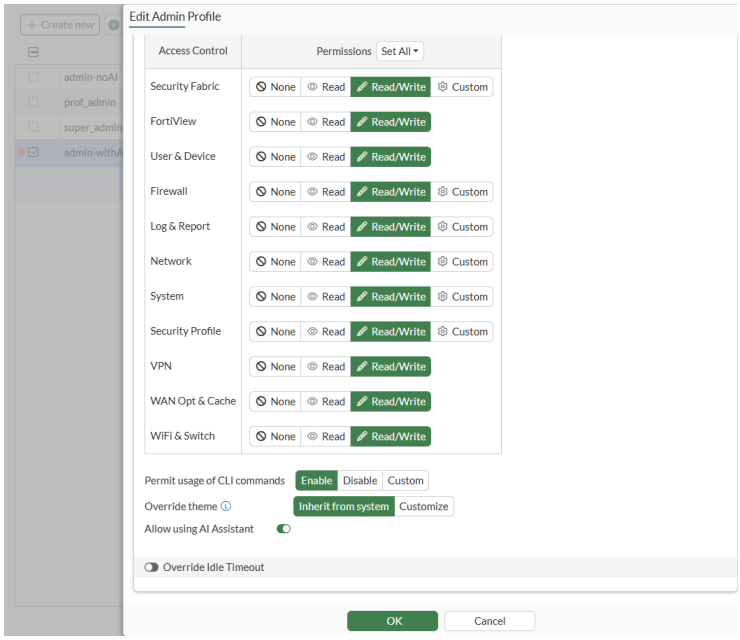
   - When FortiAI returns commands, an *Edit* button is available to open the commands in CLI Code Lab for editing.

Managing administrator access to FortiAI in administrator profiles

Administrator profiles control administrator access to FortiAI. FortiAI calls the FortiOS REST APIs with the same permissions as the logged in administrator.

To configure FortiAI access in administrator profiles in the GUI:

1. Go to *System > Admin Profiles*, and double-click an administrator profile to open it for editing.
   Of the default administrator profiles provided with FortiOS, you can edit the setting in the *prof_admin* profile but not the *super_admin* profile.



2. Enable/disable *Allow using AI Assistant* and click *OK*.
   When disabled, the FortiAI icon remains visible in the FortiOS banner for administrators; however, administrators cannot submit queries and receive answers.

To configure FortiAI access in admin profiles in the CLI:

```
config system accprofile
    edit "admin-withAI"
        set comments ''
        set secfabgrp read-write
        ...
        set gui-ai-assistant enable
    next
end
```

Masking sensitive data in chat

Sensitive data can be masked in FortiAI before it is sent to the AI provider. When data masking is enabled, a placeholder replaces sensitive data, such as IP addresses, MAC addresses, email addresses, phone numbers, URLs, and so on, before the data is sent to the AI provider.

A checkmark displays in the FortiAI chat window when data masking is enabled:

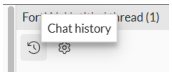
An X displays in the FortiAI chat window when data masking is disabled:

Click the icon to enable/disable data masking.

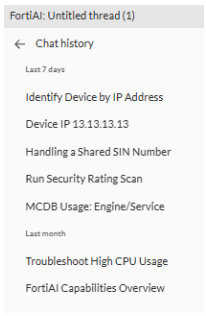
Accessing FortiAI chat history

The *Chat history* button lets you view a list of chat history. You can click any entry in the list to view the thread.

To use chat history:

1. In FortiAI, click the *Chat history* button.
   

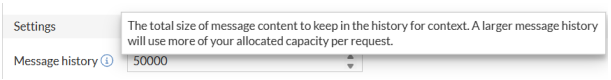
   A list of chat history is displayed.



2. Click any item in the list to view the thread.

Setting chat message history

You can specify how much message history FortiAI sends with each new query in the same thread to the AI provider. In FortiAI settings, the *Message history* field specifies the maximum number of words.



When you enter a query in FortiAI, the entire thread history is sent to the AI provider as context. When the thread history reaches the *Message history* limit, the history is compressed. Compressed history loses some information and is therefore less accurate.

A large history uses more allocated capacity for each query, and a small history may reduce the accuracy of the response because it provides too little context. It is recommended to start with the default setting.

Exporting FortiAI chats and threads

You can export the following history from FortiAI to a JSON file:

- Chat: the current conversation
- Thread: the current conversation plus details, such as timestamps, message IDs, and so on

To export chats and threads:

1. In FortiAI, click the *Settings (gear icon)* > *Export chat*, and select either *Export chat* or *Export thread*.
   A JSON file is downloaded to your local device.

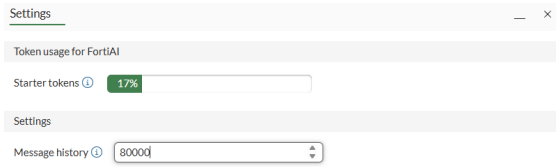
Checking token usage for FortiAI

When FortiAI is the AI provider, you can check token usage in FortiAI settings.

When monthly tokens are fully consumed, FortiAI access is temporarily suspended until the next monthly cycle begins. See also How token usage is calculated on page 447.

To check FortiAI token usage:

1. In FortiAI, click the *Settings (gear icon)*, and select *Settings*. The *Settings* are displayed.



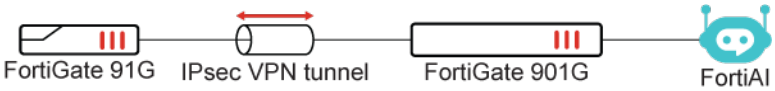
2. Under *Token usage for FortiAI*, the *Starter tokens* box displays the percentage of used tokens.
3. Click *OK* to close the window.

Example

This section includes the following examples:

- Example 1: use FortiAI to troubleshoot FortiGate on page 454
- Example 2: use OpenAI to configure FortiGate on page 455

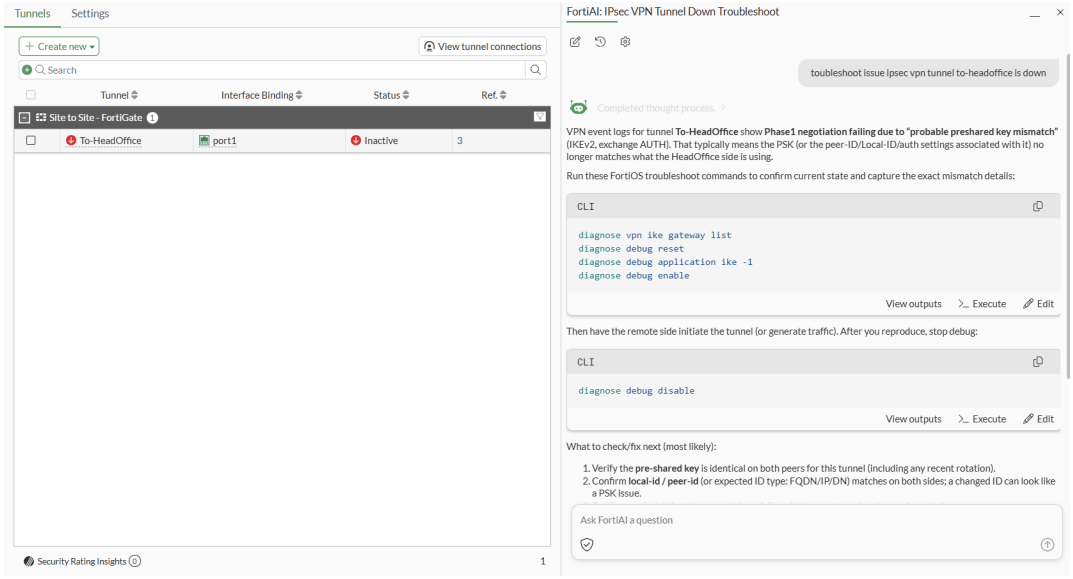
Example 1: use FortiAI to troubleshoot FortiGate



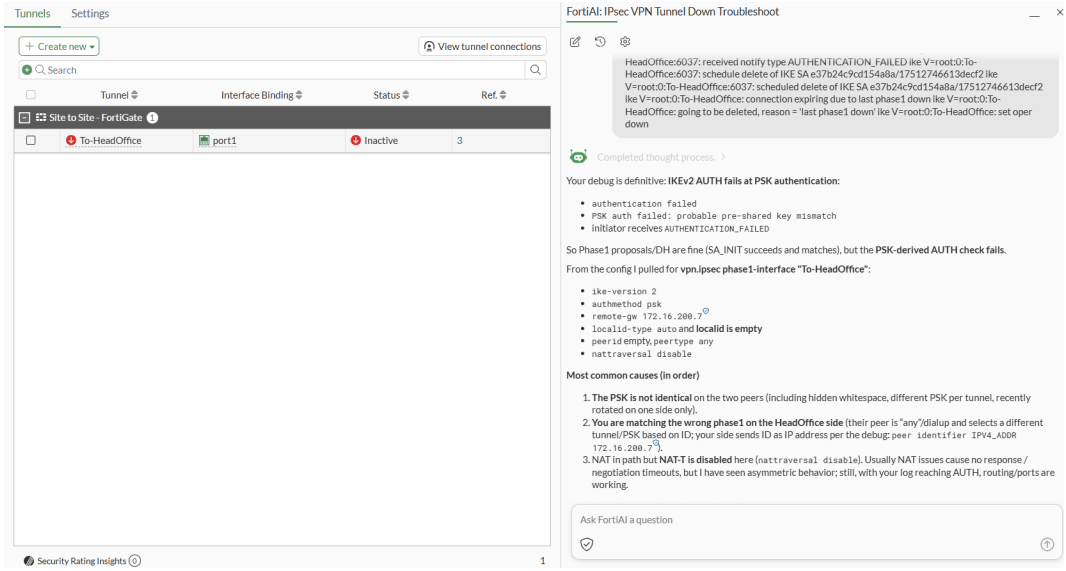
In this example, IPsec site-to-site VPN is configured between two FortiGates, but the tunnel is down. You can use FortiAI to help troubleshoot the issue.

To use FortiAI to troubleshoot:

1. Enable FortiAI as the AI provider. See Enabling FortiAI with FortiAI as AI provider on page 448.
2. In the FortiAI chat window, enter the request: `toubleshoot issue ipsec vpn tunnel to-headoffice is down`.
3. FortiAI returns CLI commands to collect IKE debug logs as well as some suggested solutions.



4. In a separate SSH session, run the commands, collect the log output, and paste the output in the chat window.
5. FortiAI analyzes the log and returns the results in order with the most common cause listed first:
   Your debug is definitive: IKEv2 AUTH fails at PSK authentication:
   - authentication failed
   - PSK auth failed: probable pre-shared key mismatch
   - initiator receives AUTHENTICATION_FAILED

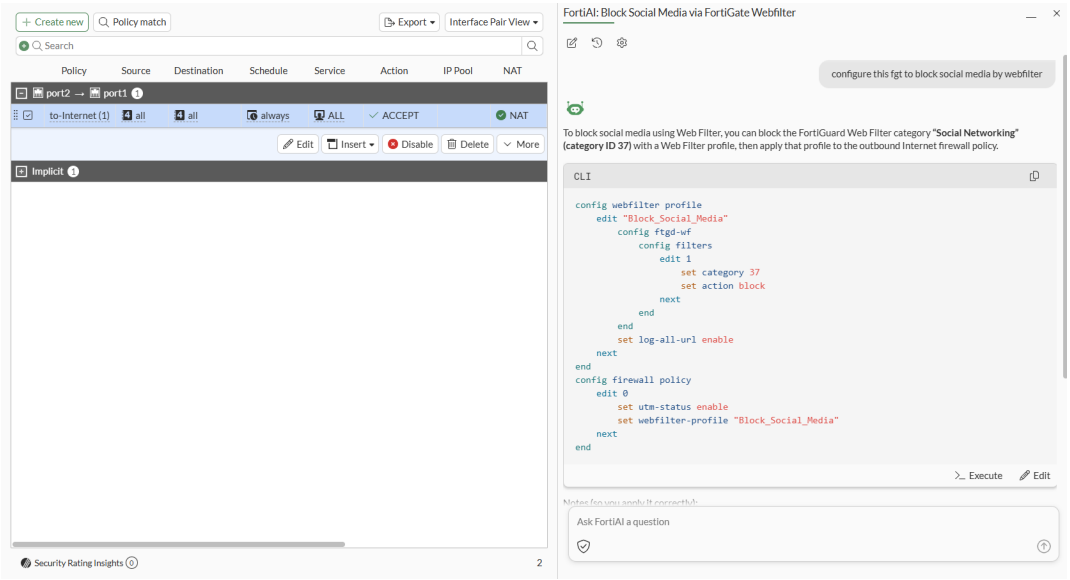


Example 2: use OpenAI to configure FortiGate

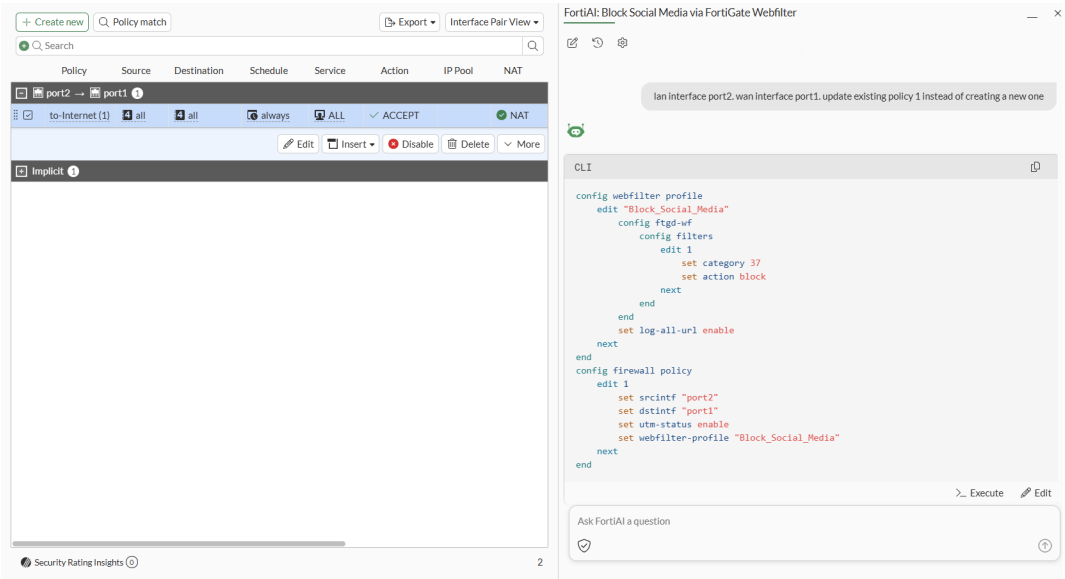
In this example, FortiAI is configured to use OpenAI as the AI provider. FortiAI is used to configure FortiGate to block social media by creating a web filter profile and updating a firewall policy to use the web filter profile.

To use OpenAI to configure FortiGate:

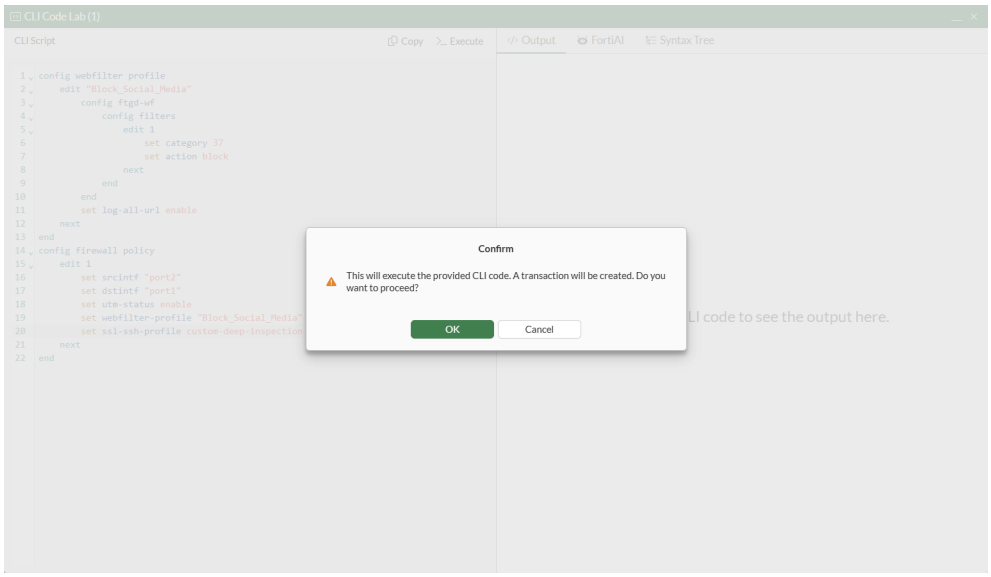
1. Enable OpenAI as the AI provider. See Enabling FortiAI with OpenAI as AI provider on page 449.
2. In the FortiAI chat window, enter the request: `configure this fgt to block social media by webfilter`.
3. FortiAI returns CLI commands for a web filter profile and firewall policy configuration.



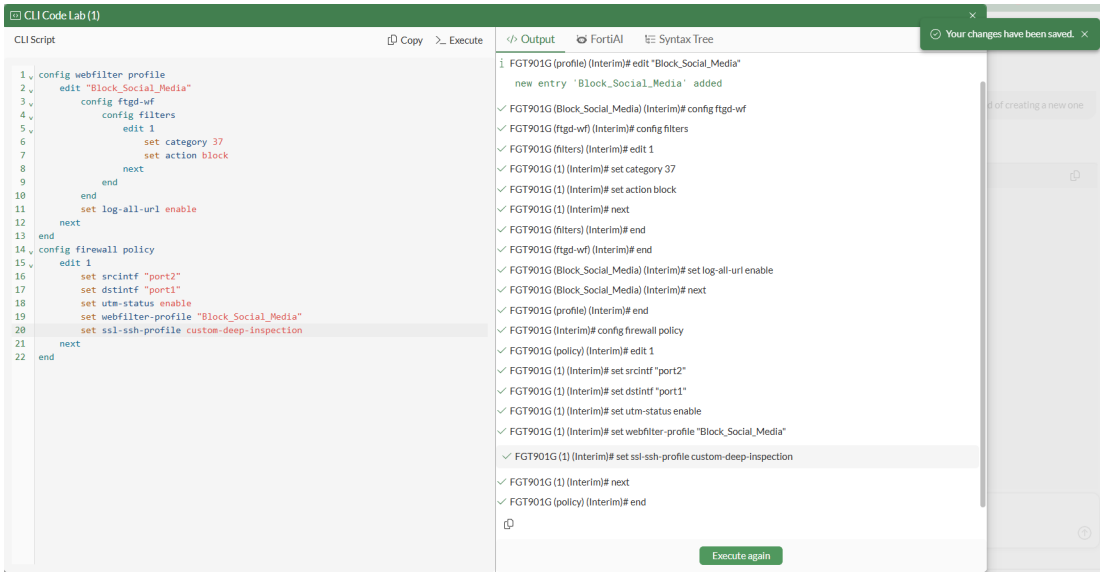
4. Enter more information, such as the LAN interface name or existing policy ID.
   For example, enter: `lan interface port2. wan interface port1. update existing policy 1 instead of creating a new one.`
   FortiAI uses the additional information to return tuned CLI commands.



5. On the bottom-right, click *Edit* to open the *CLI Code Lab*.
   In the left pane of the CLI Code Lab, edit the CLI commands as needed.
6. Click *Execute*. A *Confirm* dialog box is displayed.



7. Click *OK* to execute the commands. The output is printed in the right pane of the CLI Code Lab.
8. Click *Commit changes* to save the configuration changes.



CLI syntax

A new option is available:

```
config system accprofile
    edit <name>
        set gui-ai-assistant {enable | disable}
    next
end
```

Option	Description
`gui-ai-assistant {enable | disable}`	Enable/disable permission to use AI assistant (default = disable): • `enable`: Enable permission to use AI assistant. • `disable`: Disable permission to use AI assistant.

New options are available to enable OpenAI as the LLM provider:

```
config system admin
    edit <name>
        set accprofile <name>
        set gui-llm-provider {fortiai | openai}
        set openai-api-key <string>
        set openai-api-key-part2 <string>
        set openai-model <string>
        set openai-project-id <string>
        set openai-org-id <string>
    next
endd
```

Option	Description
`gui-llm-provider {fortiai | openai}`	Select the LLM provider: • `fortiai`: Use FortiAI. • `openai`: Use OpenAI with your own OpenAI key.
`openai-api-key <string>`	Input the OpenAI password value.
`openai-api-key-part2 <string>`	OpenAI API key part 2 for excess password value length.
`openai-model <string>`	OpenAI model name.
`openai-project-id <string>`	OpenAI project ID.
`openai-org-id <string>`	OpenAI organization ID.

Enhancements to required upgrade when firmware license is invalid or device is EOES

This information is also available in the FortiOS 8.0 Administration Guide:

- Required firmware upgrades for FortiGate with invalid support contracts or that have reached EOES

Since FortiOS 7.4.8 and 7.6.4, FortiGates are required to upgrade to the latest patch when:

- FortiGate has an invalid firmware license.
- FortiGate is running a minor FortiOS version that has reached end of engineering support (EOES).

In this enhancement, the FortiGate FortiGuard communication protocol (FCPC) is enhanced to accept a new `ForcedUpdate` flag as well as the <major>.<minor>.<patch>-<build> version from FortiGate. When a FortiGate observes its firmware license is invalid, it sends FortiGuard a firmware upgrade message with the `ForcedUpdate` flag and its version, such as `7.4.10-2878`.

In turn, FortiGuard server will ignore license check for that device and parse its firmware version. If the major and minor version on the upgrade-from and upgrade-to firmware are the same, the upgrade will be allowed.

Required upgrade does not apply to special builds.

Furthermore logs, notifications, and automation stitches are improved to provide clearer messaging about auto-upgrade and required-upgrade.

This topic contains the following sections:

- Example 1: FortiGate is required to upgrade while firmware license is invalid on page 459
- Example 2: FortiGate is required to upgrade because EOES is reached on page 460
- Automation stitches on page 462
- Logs on page 466
- Notification methods on page 467

Example 1: FortiGate is required to upgrade while firmware license is invalid

In this hypothetical scenario, a FortiGate is running FortiOS 7.6.7, and its firmware license expired on Friday, January 2, 2026:

```
System contracts:
    FMWR,Fri Jan  2 2026
```

Checking its auto-upgrade status, the `Support Contract Status` is marked `Expired`:

```
# execute auto-upgrade status
Current Image Reached End of Life: no
Is current firmware a special build: No
Support Contract Status: Expired
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled.
        New image information may be fetched.
        New image installation may be cancelled by the user.
        Next new image info fetch scheduled at (local time) Wed Apr  1 21:07:59 2026
```

A new firmware is available on the upgrade path on FortiGuard. FortiGate schedules an upgrade.

```
# execute auto-upgrade status
Current Image Reached End of Life: no
Is current firmware a special build: No
Support Contract Status: Expired
```

```
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled (Forced).
        New image information may be fetched.
        New image installation will be forced.
        Next new image info fetch scheduled at (local time) Thu Apr  2 21:19:00 2026
        New image 7.6.8b3800(07006000FIMG0025906008) installation is scheduled to:
                start at Sat Apr  4 14:08:36 2026
                end by Sat Apr  4 15:00:00 2026
        Last new image info fetch executed at (local time) Wed Apr  1 21:07:59 2026
```

When the scheduled time is reached, FortiGate will request the download from the FortiGuard server. When FortiGuard server identifies this is a `Required-Upgrade`, it will bypass license check and allow the firmware image to be downloaded.

FortiGate will install the new image, reboot, and send an email notification to the administrator.

Example 2: FortiGate is required to upgrade because EOES is reached

In this hypothetical scenario, a FortiGate has a valid firmware license but the current firmware has reached the EOES date.

The current firmware version is lower than the patch level image released on FortiGuard server.

Current firmware:

```
# get system status | grep Version
Version: FortiGate-40F v7.6.7,build3679,260330
```

Available firmware on FortiGuard:

```
# diagnose fdsm image-list
Last update: 895 secs ago. Total: 5
... ...
07006000FIMG0025906008  v7.6 MR6-GA-F P8 b3800 (upgrade)
07002000FIMG0025902012  v7.2 MR2-GA-F P12 b1800 (downgrade)
07002000FIMG0025902011  v7.2 MR2-NA- b1737 (downgrade)
```

Upgrade path:

```
# diagnose fdsm image-upgrade-matrix
Last update: 1013 secs ago. Total: 315
v8.0.0.(b150) -> v8.0.1.F(b1234)        (id:08000000FIMG0025900001)
v7.6.7.(b3679) -> v7.6.8.F(b3800)       (id:07006000FIMG0025906008)
```

Checking its auto-upgrade status, the end of life status is set to `yes`.

```
FortiGate-40F (Interim)# execute auto-upgrade status
Current Image Reached End of Life: yes
Is current firmware a special build: No
```

```
Support Contract Status: Active
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled.
        New image information may be fetched.
        New image installation may be cancelled by the user.
        Next new image info fetch scheduled at (local time) Thu Apr  2 14:33:55 2026
```

To force FortiGate to trigger a manual update-check, run this command:

```
# execute auto-upgrade check-for-new-image
Manually triggering auto-upgrade new image check...
Auto-upgrade new image check triggered.
Firmware Update from Management Service
Upgrade to image 7.6.8b3800-F (07006000FIMG0025906008) has been scheduled between Sat Apr  4
14:21:30 2026 and Sat Apr  4 15:00:00 2026.
This upgrade is forced and cannot be cancelled, to delay this upgrade, please use "execute auto-
upgrade delay-installation".
```

The FortiGate is now scheduled to perform the upgrade:

```
# execute auto-upgrade status
Current Image Reached End of Life: yes
Is current firmware a special build: No
Support Contract Status: Active
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled (Forced).
        New image information may be fetched.
        New image installation will be forced.
        Next new image info fetch scheduled at (local time) Thu Apr  2 14:51:45 2026
        New image 7.6.8b3800(07006000FIMG0025906008) installation is scheduled to:
                start at Sat Apr  4 14:21:30 2026
                end by Sat Apr  4 15:00:00 2026
        Last new image info fetch executed at (local time) Wed Apr  1 17:41:10 2026
```

Upon the completion of the upgrade, FortiGate will trigger an automation stitch to send the registered FortiCare account owner an email stating:

```
An automated firmware update has completed successfully. The device FortiGate-40F, serial number
      FGT40FTK19004135, was updated to firmware version 7.6.8 on 2026-4-4 15:00:06.
This is an informational notification and was generated by automation stitch configuration on
      FortiGate-40F.
```

Administrators can delay an upgrade for one (1) week using the command:

```
# execute auto-upgrade delay-installation
Postponing auto-upgrade image installation to a week later...
Auto-upgrade image installation rescheduled to: start at local time Sat Apr 11 10:12:08
2026
        end by local time Sat Apr 11 11:00:00 2026
```

Automation stitches

Wording in the following default automation stitches has been improved for firmware upgrade:

```
show system automation-stitch ?
name    Name.
...
Firmware Upgrade Cancelled
Firmware Upgrade Complete
Firmware Upgrade Notification
Firmware Upgrade Scheduled
```

This section shows the settings for the following automation stitches:

- Firmware upgrade canceled on page 462
- Firmware upgrade complete on page 463
- Firmware upgrade notification on page 464
- Firmware upgrade scheduled on page 465

Firmware upgrade canceled

This section shows the settings, trigger, and action for the following default automation stitch: `Firmware Upgrade Cancelled`. The automation stitch is used when an automatic firmware upgrade is canceled.

- View the default automation stitch for `Firmware Upgrade Cancelled`:

```
show system automation-stitch "Firmware Upgrade Cancelled"
config system automation-stitch
    edit "Firmware Upgrade Cancelled"
        set description "Automatic firmware upgrade cancelled notification."
        set trigger "Auto Firmware Upgrade Cancelled"
        config actions
            edit 1
                set action "Auto Upgrade Cancelled Email Notification"
            next
        end
    next
end
```

- View the trigger:

  The `"Auto Firmware Upgrade Cancelled"` is the event, and `32325` is the new logid for `"Automatic firmware upgrade image installation was cancelled"`.

```
config system automation-trigger
    edit "Auto Firmware Upgrade Cancelled"
        set description "Automatic firmware upgrade cancelled."
        set event-type event-log
        set logid 32325
    next
end
```

- View the action:

By default, an email notification is sent to the email address for the FortiCare account. The default email subject and content is shown.

```
config system automation-action
    edit "Auto Upgrade Cancelled Email Notification"
        set description "Send auto upgrade cancelled email notification to the FortiCare email
address registered on this device."
        set action-type email
        set forticare-email enable
        set email-to "<name>@<company>.com"
        set email-subject "Firmware Update Cancelled Notification : %%log.devname%%"
        set message "Date: %%log.date%% Time: %%log.time%%
An automated firmware update has been cancelled on device: %%log.devname%%, serial number
%%log.devid%%.

This is an informational notification and was generated by automation stitch configuration on
%%log.devname%%."
    next
end
```

Firmware upgrade complete

This section shows the settings, trigger, and action for the following default automation stitch: `Firmware Upgrade Complete`. The automation stitch is used when an automatic firmware upgrade completes.

- View the default automation stitch for `Firmware Upgrade Complete`:

```
edit "Firmware Upgrade Complete"
        set description "Automatic firmware upgrade complete notification."
        set trigger "Auto Firmware Upgrade Complete"
        config actions
            edit 1
                set action "Auto Upgrade Complete Email Notification"
            next
        end
    next
end
```

- View the trigger:

  The `"Auto Firmware Upgrade Complete"` is the event, and `22096` is the new logid for `"Automatic firmware upgrade complete successfully"`.

```
config system automation-trigger
    edit "Auto Firmware Upgrade Complete"
        set description "Automatic firmware upgrade complete successfully."
        set event-type event-log
        set logid 22096 22094
    next
end
```

- View the action:

By default, an email notification is sent to the email address for the FortiCare account. The default email subject and content is shown.

```
edit "Auto Upgrade Complete Email Notification"
        set description "Send auto upgrade complete email notification to the FortiCare email
address registered on this device."
        set action-type email
        set forticare-email enable
        set email-to "<name>@<company>.com"
        set email-subject "Firmware Update Notification : %%log.devname%%"
        set message "An automated firmware update has completed successfully. The device
%%log.devname%%, serial number %%log.devid%%, was updated to firmware version %%log.version%%
on %%log.date%% %%log.time%%.

This is an informational notification and was generated by automation stitch configuration on
%%log.devname%%."
    next
end
```

For example, when an upgrade completes, it is captured in the logs:

```
183: date=2026-03-06 time=07:16:18 eventtime=1772738177229032043 tz="+1200" logid="0100022096"
type="event" subtype="system" level="information" vd="root" logdesc="An automatic firmware upgrade
was completed" msg="Automatic upgrade complete" upgradesource="auto-firmware-upgrade"
version="7.6.5" nodecount=1 fgtcount=1
```

Triggered by the default automation stitch, FortiGate sends an email notification about the completed upgrade:

```
Firmware Update Scheduled Notification : 245
Date: 2026-03-05 Time: 10:30:05
An automated firmware update has completed successfully. The device 245, serial number
FG9H1GTB25900162, was updated to firmware version 7.6.5 on Mar 5 10:30:05 2026

This is an informational notification and was generated by automation stitch configuration on 245.

2026-03-05 10:30:05 mail_info:
        from:fortinet-notifications.com user:DoNotReply@fortinet-notifications.com
2026-03-05 10:30:05 mail_info:
        reverse path:DoNotReply@fortinet-notifications.com
        user name:DoNotReply
2026-03-05 10:30:05 to[0]:<name>@<company>.com
2026-03-05 10:30:05 <==_init_mail_info
2026-03-05 10:30:05 create session
```

Firmware upgrade notification

This section shows the settings, trigger, and action for the following default automation stitch: `Firmware Upgrade Notification`. The automation stitch notifies administrators when an automatic firmware upgrade fails.

- View the default automation stitch for `Firmware Upgrade Notification`:
  By default, the notification is disabled. The `"Auto Firmware Upgrade"` trigger is for failed upgrade events.

```
edit "Firmware Upgrade Notification"
        set description "Automatic firmware upgrade notification."
        set status disable
        set trigger "Auto Firmware Upgrade"
        config actions
            edit 1
                set action "Email Notification"
            next
        end
    next
end
```

- View the trigger:

  The `"Auto Firmware Upgrade"` is the event, and `22097` is the logid for `"Automatic firmware upgrade failed"`.

```
config system automation-trigger
    edit "Auto Firmware Upgrade"
        set description "Automatic firmware upgrade."
        set event-type event-log
        set logid 22095 22097
    next
end
```

- View the action:

  By default, an email notification is sent to the email address for the FortiCare account. The default email subject is the log.

```
edit "Email Notification"
        set description "Send a custom email to the specified recipient(s)."
        set action-type email
        set forticare-email enable
        set email-subject "%%log.logdesc%%"
    next
end
```

Firmware upgrade scheduled

This section shows the settings, trigger, and action for the following default automation stitch: Firmware Upgrade Scheduled. The automation stitch is to notify administrators when an automatic firmware upgrade is scheduled.

- View the default automation stitch for Firmware Upgrade Scheduled:

```
config system automation-stitch
    edit "Firmware Upgrade Scheduled"
        set description "Automatic firmware upgrade scheduled notification."
        set trigger "Auto Firmware Upgrade Scheduled"
        config actions
            edit 1
                set action "Auto Upgrade Scheduled Email Notification"
            next
        end
```

```
    next
end
```

- View the trigger:

  The "Auto Firmware Upgrade Scheduled" is the event, and 32263 is the logid for the system event "Automatic firmware image installation is (re)scheduled".

```
config system automation-trigger
    edit "Auto Firmware Upgrade Scheduled"
        set description "Automatic firmware upgrade scheduled."
        set event-type event-log
        set logid 32263
    next
end
```

- View the action:

  By default, an email notification is sent to the email address for the FortiCare account. The default email subject and content is shown.

```
config system automation-action
    edit "Auto Upgrade Scheduled Email Notification"
        set description "Send scheduled auto upgrade email notification to the FortiCare email
address registered on this device."
        set action-type email
        set forticare-email enable
        set email-to "<name>@<company>.com"
        set email-subject "Firmware Update Scheduled Notification : %%log.devname%%"
        set message "Date: %%log.date%% Time: %%log.time%%
An automated firmware update has been scheduled on device: %%log.devname%%, serial number
%%log.devid%%.
%%log.msg%%

This is an informational notification and was generated by automation stitch configuration on
%%log.devname%%."
    next
end
```

Logs

The following log IDs are used with automation stitches for automatic and required upgrades to trigger email notifications:

Log ID	Description
32325	Log ID for the Automatic firmware upgrade image installation was cancelled trigger.
22096	Log ID for the Automatic firmware upgrade complete successfully trigger.
32263	Log ID for the Automatic firmware image installation is (re)scheduled system event.

Notification methods

Several methods are used to notify administrators about a required upgrade.

Before a required upgrade starts, the FortiOS GUI banner shows a new firmware version is available, for example, *A new firmware version is available: v7.6.8*.

After FortiOS confirms a valid upgrade path for the required upgrade:

- On the *System > Firmware & Registration* page:
  - The *Upgrade Status* column displays *Upgrade to v7.6.8 shortly*.
  - The *Cancel Fabric Upgrade* button is grayed out because forced upgrades cannot be canceled.
- The Console prints the following information when scheduled to install the new image:

```
Auto-upgrade new image check triggered.

Firmware Update from Management Service
Upgrade to image 7.6.8b3800-F (07006000FIMG0025906008) has been scheduled between Sat Apr  4
14:08: 36 2026 and Sat Apr  4 15:00:00 2026.
This upgrade is forced and cannot be cancelled, to delay this upgrade, please use "execute
auto-upgrade delay-installation".
```

- A default automation stitch triggers FortiGate to send an email notification:
  - FortiGate sends an email notification to the email address for the registered FortiCare account before and after upgrade:

```
An automated firmware update has completed successfully. The device FortiGate-40F, serial
number FGT40FTK<number>, was updated to firmware version 7.6.8 on 2026-4-4 15:00:06.

This is an informational notification and was generated by automation stitch configuration
on FortiGate-40F.
```

High availability

This section includes information about HA related new features:

- IPv6 support for in-band management IP on page 468
- Support IPv6 multicast route synchronization in HA on page 469
- FGSP source IP and interface selection on page 470
- HA monitoring support for software switch member interfaces on page 472
- FGCP and FGSP synchronizes full-cone expectation sessions when session-pickup-expectation enabled on page 475
- Enhanced FGCP monitoring with interface group awareness on page 481
- GUI support for HA actions, health status, and config diff on page 485
- FGSP synchronizes firewall-authenticated users on page 487
- HA role shown in CLI prompt on page 492

IPv6 support for in-band management IP

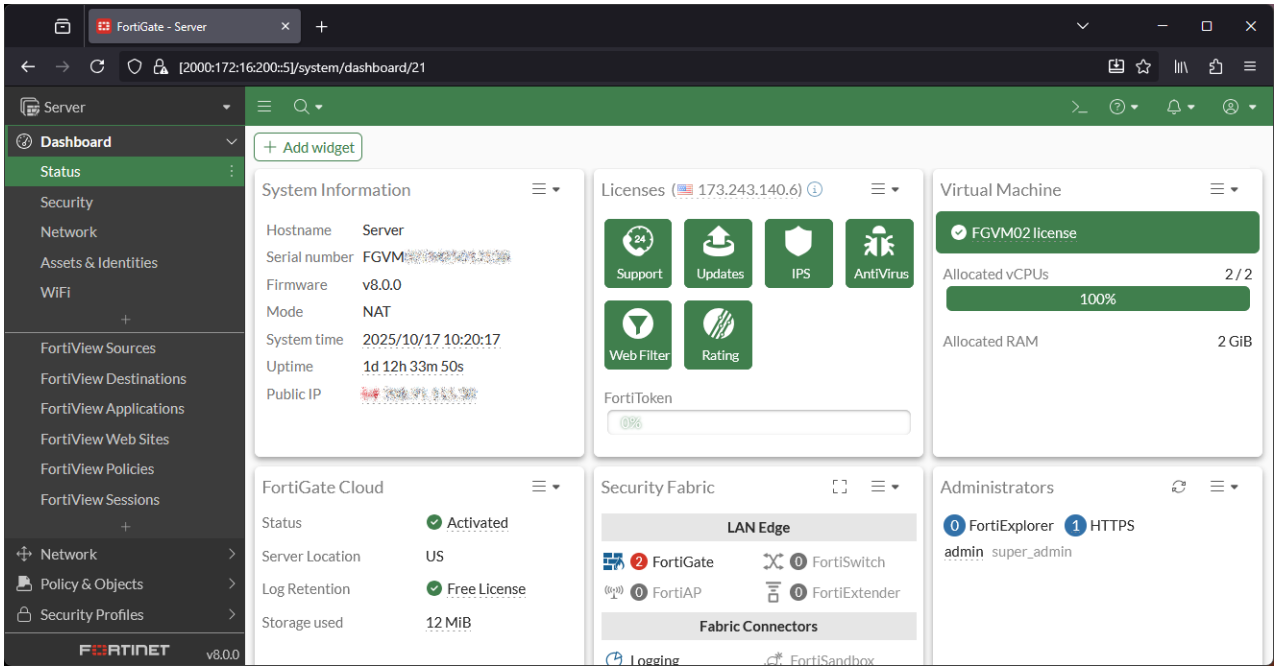
IPv6 support has been added to the in-band management IP feature, enabling HA members to be accessed via IPv6. This provides greater flexibility and future-proofing for networks transitioning to IPv6. In-band management IP addresses serve as an alternative to dedicated HA management interfaces. See In-band management for more information.

```
config system interface
    edit <interface>
        config ipv6
            set ip6-mgmt-address <IPv6 address>
        end
    next
end
```

For example, if the IPv6 management IP address is 2000:172:16:200::5, it could be configured as follows:

```
config system interface
    edit "wan1"
        config ipv6
            set ip6-address 2000:172:16:200::1/64
            set ip6-mgmt-address 2000:172:16:200::5/64
            set ip6-allowaccess ping https ssh snmp http telnet fgfm
        end
    next
end
```

The GUI dashboard could then be accessed at *https://[2000:172:16:200::5]/system/dashboard/21*.

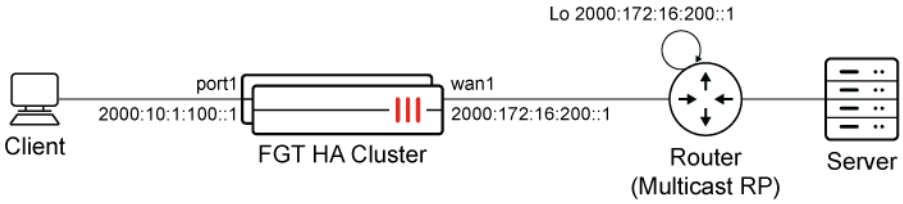


Support IPv6 multicast route synchronization in HA

HA now supports synchronizing IPv6 multicast routes from primary to secondary, ensuring that IPv6 multicast traffic has minimal disruption during an HA failover.

Example

The following example illustrates how to confirm IPv6 multicast routes are synchronized from primary to secondary FortiGates in an HA cluster.



To verify IPv6 :

1. On the primary FortiGate in the HA cluster, configure multicast routing:

```
config router multicast6
    set multicast-routing enable
    config interface
        edit "wan1"
        next
        edit "port1"
        next
    end
    config pim-sm-global
        config rp-address
            edit 1
                set ip6-address 2000:172:16:210::1
            next
        end
    end
end
```

2. Set up multicast configurations and RP on the router.
   The FortiGate HA cluster and router will have multicast IPv6 routes.
3. Check the primary and secondary FortiGates in the cluster to confirm IPv6 multicast route synchronization.
   Synchronization for the multicast IPv6 route is identified by `flags=(0x10000005)`.
   Check the primary FortiGate in the cluster:

```
FGT-A# diagnose ipv6 multicast mroute
grp=ff0e::10 src=2000:10:1:100::11 intf=11 flags=(0x10000000)[  ] status=resolved
        last_assert=6241506 bytes=865648 pkt=826 wrong_if=0 num_ifs=1
        index(ttl)=[7(1),]
```

   Check the secondary FortiGate in the cluster and note the synchronization mark:

```
FGT-B# diagnose ipv6 multicast mroute
grp=ff0e::10 src=2000:10:1:100::11 intf=11 flags=(0x10000005)[S ] status=resolved
        last_assert=6243961 bytes=0 pkt=0 wrong_if=0 num_ifs=1
        index(ttl)=[7(1),]
```

FGSP source IP and interface selection

A specific source IP or interface can be selected for FGSP heartbeat/sync traffic over L3, preventing packet loss caused by ECMP path divergence.

```
config system standalone-cluster
    config cluster-peer
        edit <sync-id>
            set source-ip <class_ip>
            set interface <interface>
        next
    end
end
```

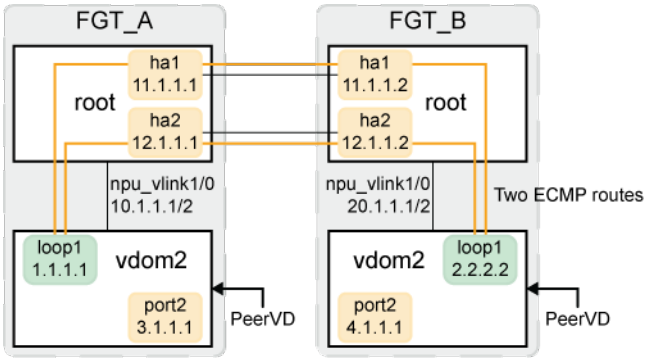
Command	Description
`source-ip <class_ip>`	Source IP address to use for peer connections.
`interface <interface>`	Outgoing interface for peer connections.

When configure the interface, always specify the outgoing (egress) interface, and not the source interface.

Example

In this example, FGSP is configured between FGT_A and FGT_B. Peering is over Layer3 network through two ECMP routes.

FGSP peers could be setup between two loopback interfaces, port2, or both.



To use the source IP for FGSP heartbeat/sync traffic:

- Configure FGT_A:

```
config system standalone-cluster
    set standalone-group-id 20
    set group-member-id 3
    config cluster-peer
        edit 1
            set peervd "v2"
            set peerip 2.2.2.2
            set syncvd "root"
            set source-ip 1.1.1.1
        next
    end
end
```

- Configure FGT_B:

```
config system standalone-cluster
    set standalone-group-id 20
    set group-member-id 4
    config cluster-peer
        edit 1
            set peervd "v2"
            set peerip 1.1.1.1
            set syncvd "root"
            set source-ip 2.2.2.2
        next
    end
end
```

To use the interface for FGSP heartbeat/sync traffic:

In this example, synchronization traffic originates on `loopback1` but must exit the device through `npu-vlink1` to reach its peer, so the interface must be set to `npu-vlink1`. Always select the port that serves as the traffic's exit point to ensure successful delivery.

- Configure FGT_A:

```
config system standalone-cluster
    set standalone-group-id 20
    set group-member-id 3
    config cluster-peer
        edit 1
            set peervd "v2"
            set peerip 2.2.2.2
            set syncvd "root"
            set interface npu-vlink1
        next
    end
end
```

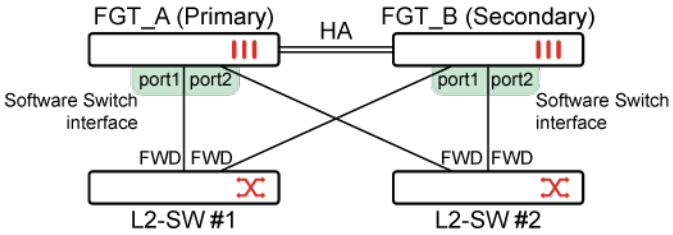
- Configure FGT_B:

```
config system standalone-cluster
    set standalone-group-id 20
    set group-member-id 4
    config cluster-peer
        edit 1
            set peervd "v2"
            set peerip 1.1.1.1
            set syncvd "root"
            set interface npu-vlink1
        next
    end
end
```

HA monitoring support for software switch member interfaces

Software switch member interfaces can be used as HA monitor interfaces. Previously, these interfaces could not be monitored. This enhancement improves configuration flexibility and enables more reliable failover using existing interface layouts.

Example



To check that the software switch member interfaces are used as HA monitor interfaces:

1. Configure a software switch interface:

```
config system switch-interface
    edit "sw1"
        set vdom "root"
        set member "port1" "port2"
    next
end
```

2. Configure HA to monitor the software switch member interfaces:

```
config system ha
    set group-id 240
```

```
    set group-name "FG_HA"
    set mode a-p
    set password ENC 5BgrSGI4E
    set hbdev "ha1" 50 "ha2" 100
    set session-pickup enable
    set override enable
    set priority 200
    set monitor "port1" "port2" "wan1" "wan2"
end
```

3. Check the interface status:

```
FGT_A # get system ha status
HA Health Status: OK
Model: FortiGate-101F
Mode: HA A-P
Group Name: FG_HA
Group ID: 240
Debug: 0
Cluster Uptime: 0 days 0h:1m:19s
Cluster state change time: 2026-02-03 11:05:52
Primary selected using:
    <2026/02/03 11:05:52> vcluster-1: FG101FTK19003737 is selected as the primary because its
override priority is larger than peer member FG101FTK19002788.
ses_pickup: enable, ses_pickup_delay=disable
override: enable
Configuration Status:
    FG101FTK19003737(updated 3 seconds ago): in-sync
    FG101FTK19003737 chksum dump: 75 36 6e e1 b5 e5 d4 c9 9f 06 04 ff e4 d1 5c c9
    FG101FTK19002788(updated 4 seconds ago): in-sync
    FG101FTK19002788 chksum dump: 75 36 6e e1 b5 e5 d4 c9 9f 06 04 ff e4 d1 5c c9
System Usage stats:
    FG101FTK19003737(updated 3 seconds ago):
        sessions=5, average-cpu-user/nice/system/idle=0%/0%/0%/99%, memory=35%
    FG101FTK19002788(updated 4 seconds ago):
        sessions=3, average-cpu-user/nice/system/idle=0%/0%/0%/99%, memory=34%
HBDEV stats:
    FG101FTK19003737(updated 3 seconds ago):
        ha1: physical/1000auto, up, rx-bytes/packets/dropped/errors=207719/438/0/0,
tx=173193/337/0/0
        ha2: physical/1000auto, up, rx-bytes/packets/dropped/errors=762549/2018/0/0,
tx=365824/982/0/0
    FG101FTK19002788(updated 4 seconds ago):
        ha1: physical/1000auto, up, rx-bytes/packets/dropped/errors=177032/379/0/0,
tx=199336/391/0/0
        ha2: physical/1000auto, up, rx-bytes/packets/dropped/errors=857970/1991/0/0,
tx=318297/1107/0/0
MONDEV stats:
    FG101FTK19003737(updated 3 seconds ago):
        port1: physical/1000auto, up, rx-bytes/packets/dropped/errors=9714/136/0/0,
tx=1795/13/0/0
        port2: physical/1000auto, up, rx-bytes/packets/dropped/errors=4374/47/0/0,
```

```
tx=5777/81/0/0
        wan1: physical/1000auto, up, rx-bytes/packets/dropped/errors=135184/614/0/0,
tx=191326/490/0/0
        wan2: physical/1000auto, up, rx-bytes/packets/dropped/errors=2368/14/0/0,
tx=1770/3/0/0
    FG101FTK19002788(updated 4 seconds ago):
        port1: physical/1000auto, up, rx-bytes/packets/dropped/errors=11196/157/0/0,
tx=1337/7/0/0
        port2: physical/1000auto, up, rx-bytes/packets/dropped/errors=5016/54/0/0,
tx=1337/7/0/0
        wan1: physical/1000auto, up, rx-bytes/packets/dropped/errors=12847/156/0/0, tx=0/0/0/0
        wan2: physical/1000auto, up, rx-bytes/packets/dropped/errors=1838/11/0/0, tx=0/0/0/0
number of member: 2
FGT_A           , FG101FTK19003737, HA cluster index = 0
FGT_B           , FG101FTK19002788, HA cluster index = 1
number of vcluster: 1
vcluster 1: work 169.254.0.1
Primary: FG101FTK19003737, HA operating index = 0
Secondary: FG101FTK19002788, HA operating index = 1
```

4. Shutdown the port2 connection on the switch side and confirm that the link is down:

```
FGT_A # diagnose hardware deviceinfo nic port2
Description         :FortiASIC NP6XLITE Adapter
Driver Name         :FortiASIC NP6XLITE Driver
Current_HWaddr       04:d5:90:04:f8:a1
Permanent_HWaddr     04:d5:90:04:f8:a1
========== Link Status ==========
Speed               :N/A
Duplex              :N/A
link_status         :Down
============ Counters ===========
Rx Pkts             :212
Rx Bytes            :16716
```

5. Check that failover occurred and FGT_A became the secondary after the failover:

```
FGT_A # get system ha status
HA Health Status:
    WARNING: FG101FTK19003737 has mondev down;
Model: FortiGate-101F
Mode: HA A-P
Group Name: FG_HA
Group ID: 240
Debug: 0
Cluster Uptime: 0 days 0h:9m:41s
Cluster state change time: 2026-02-03 11:12:21
Primary selected using:
    <2026/02/03 11:12:21> vcluster-1: FG101FTK19002788 is selected as the primary because the
value of link-failure + pingsvr-failure is less than peer member FG101FTK19003737.
    <2026/02/03 11:05:52> vcluster-1: FG101FTK19003737 is selected as the primary because its
override priority is larger than peer member FG101FTK19002788.
```

```
ses_pickup: enable, ses_pickup_delay=disable
override: enable

MONDEV stats:
    FG101FTK19003737(updated 4 seconds ago):
        port1: physical/1000auto, up, rx-bytes/packets/dropped/errors=61541/932/0/0,
tx=5960/34/0/0
        port2: physical/00, down, rx-bytes/packets/dropped/errors=16716/212/0/0,
tx=28485/435/0/0
        wan1: physical/1000auto, up, rx-bytes/packets/dropped/errors=686500/3018/0/0,
tx=1007397/2263/0/0
        wan2: physical/1000auto, up, rx-bytes/packets/dropped/errors=9146/47/0/0,
tx=7080/12/0/0
    FG101FTK19002788(updated 5 seconds ago):
        port1: physical/1000auto, up, rx-bytes/packets/dropped/errors=62501/948/0/0,
tx=5597/32/0/0
        port2: physical/1000auto, up, rx-bytes/packets/dropped/errors=23868/309/0/0,
tx=17183/223/0/0
        wan1: physical/1000auto, up, rx-bytes/packets/dropped/errors=330856/1711/0/0,
tx=433062/950/0/0
        wan2: physical/1000auto, up, rx-bytes/packets/dropped/errors=8616/44/0/0,
tx=4720/8/0/0
number of member: 2
FGT_A           , FG101FTK19003737, HA cluster index = 0
FGT_B           , FG101FTK19002788, HA cluster index = 1
number of vcluster: 1
vcluster 1: standby 169.254.0.2
Secondary: FG101FTK19003737, HA operating index = 1
Primary: FG101FTK19002788, HA operating index = 0
```

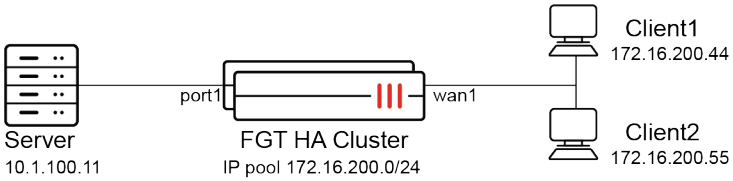
FGCP and FGSP synchronizes full-cone expectation sessions when session-pickup-expectation enabled

FGCP or FGSP HA configured to synchronize expectation sessions, now also synchronizes full cone NAT UDP expectation sessions. For information about full cone NAT, see Full cone NAT for fixed port range IP pools.

HA configuration to synchronize UDP and expectation sessions:

```
config system ha
    set session-pickup enable
    set session-pickup-connectionless enable
    set session-pickup-expectation enable
end
```

Example configuration



Example HA configuration with `session-pickup`, `session-pickup-connectionless`, and `session-pickup-expectation` enabled.

```
config system ha
    set group-id 240
    set group-name FG_HA
    set mode a-p
    set password <password>
    set hbdev ha1 50 ha2 100
    set session-pickup enable
    set session-pickup-connectionless enable
    set session-pickup-expectation enable
    set override enable
    set priority 255
end
```

Example Full cone NAT IP pool:

```
config firewall ippool
    edit full_cone_nat
        set type fixed-port-range
        set startip 172.16.200.155
        set endip 172.16.200.165
        set startport 5000
        set endport 5087
        set source-startip 10.1.100.10
        set source-endip 10.1.100.23
        set port-per-user 32
        set permit-any-host enable
    end
```

Example full cone NAT Firewall policy:

```
config firewall policy
    edit 10
        set name FW
        set uuid a2efc630-015a-51f1-a27b-aa72f6406517
        set srcintf port1
        set dstintf wan1
        set action accept
        set srcaddr all
        set dstaddr all
```

```
        set schedule always
        set service ALL
        set auto-asic-offload disable
        set nat enable
        set ippool enable
        set poolname full_cone_nat
    end
```

In the example, the server connects to Client2 and this opens up full cone NAT ports on the FortiGate cluster wan1 interface that allow CLient1 and Client2 to connect through the FortiGate to the server.

When the server connects to Client2, the following session is created on the primary FortiGate in the HA cluster:

```
diagnose sys session list

session info: proto=17 proto_state=00 duration=218 expire=179 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=5
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty synced f16
statistic(bytes/packets/allow_err): org=29145088/19456/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 133569/1068 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=172.16.200.55/0.0.0.0
hook=post dir=org act=snat 10.1.100.11:5050->172.16.200.55:5001(172.16.200.155:5050)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.155:5050(10.1.100.11:5050)
misc=0 policy_id=10 pol_uuid_idx=667 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
hrx info: valid=1/0, qid=7/0, npuid=1/0, sublink=0/0
```

Because of full cone NAT, the following expectation sessions are also created on the primary FortiGate:

```
diagnose sys session list expectation

session info: proto=17 proto_state=00 duration=251 expire=-221 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=2
origin-shaper=
reply-shaper=
per_ip_shaper=
master session: proto=17 10.1.100.11:5050->172.16.200.55:5001
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=255/255
state=new f31
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=dnat 10.1.100.11:0->172.16.200.155:5050(10.1.100.11:0)
hook=pre dir=org act=noop 0.0.0.0:0->0.0.0.0:0(0.0.0.0:0)
misc=0 policy_id=10 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
```

```
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x4000000
no_ofld_reason:  new
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0

session info: proto=17 proto_state=00 duration=251 expire=-221 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=2
origin-shaper=
reply-shaper=
per_ip_shaper=
master session: proto=17 10.1.100.11:5050->172.16.200.55:5001
class_id=0 ha_id=0 policy_dir=1 tunnel=/ vlan_cos=255/255
state=new f31
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=0->11/11->0 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=dnat 0.0.0.0:0->172.16.200.155:5050(10.1.100.11:5050)
hook=pre dir=org act=noop 0.0.0.0:0->0.0.0.0:0(0.0.0.0:0)
misc=0 policy_id=10 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x4000000
no_ofld_reason:  new
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0
```

HA synchronizes the following session to the backup or secondary FortiGate:

```
diagnose sys session list

session info: proto=17 proto_state=00 duration=338 expire=111 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=5
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=dirty may_dirty syn_ses
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.11:5050->172.16.200.55:5001(172.16.200.155:5050)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.155:5050(10.1.100.11:5050)
misc=0 policy_id=10 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x4000000
no_ofld_reason:  npu-flag-off
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0
```

As well, HA synchronizes the following expectation sessions to the backup or secondary FortiGate:

```
diagnose sys session list expectation

session info: proto=17 proto_state=00 duration=107 expire=-77 timeout=0 refresh_dir=both
```

```
flags=00000000 socktype=0 sockport=0 av_idx=0 use=2
origin-shaper=
reply-shaper=
per_ip_shaper=
master session: proto=17 10.1.100.11:5050->172.16.200.55:5001
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=255/255
state=new synced f31
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=dnat 10.1.100.11:0->172.16.200.155:5050(10.1.100.11:0)
hook=pre dir=org act=noop 0.0.0.0:0->0.0.0.0:0(0.0.0.0:0)
misc=0 policy_id=10 pol_uuid_idx=665 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x000001 no_offload
no_ofld_reason:  new disabled-by-policy
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0

session info: proto=17 proto_state=00 duration=107 expire=-77 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=2
origin-shaper=
reply-shaper=
per_ip_shaper=
master session: proto=17 10.1.100.11:5050->172.16.200.55:5001
class_id=0 ha_id=0 policy_dir=1 tunnel=/ vlan_cos=255/255
state=new synced f31
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=0->11/11->0 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=dnat 0.0.0.0:0->172.16.200.155:5050(10.1.100.11:5050)
hook=pre dir=org act=noop 0.0.0.0:0->0.0.0.0:0(0.0.0.0:0)
misc=0 policy_id=10 pol_uuid_idx=665 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x000001 no_offload
no_ofld_reason:  new disabled-by-policy
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0
```

Full cone NAT also allows Client1 to connect to the server through the FortiGate cluster. After Client1 connects, the primary FortiGate includes the following sessions:

```
diagnose sys session list

session info: proto=17 proto_state=00 duration=454 expire=1724 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
master session: proto=17 10.1.100.11:5050->172.16.200.55:5001
class_id=0 ha_id=0 policy_dir=1 tunnel=/ vlan_cos=255/255
state=synced intree
statistic(bytes/packets/allow_err): org=22717170/15165/1 reply=0/0/0 tuples=2
```

```
tx speed(Bps/kbps): 55604/444 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=7->11/11->7 gwy=10.1.100.11/172.16.200.44
hook=pre dir=org act=dnat 172.16.200.44:5020->172.16.200.155:5050(10.1.100.11:5050)
hook=post dir=reply act=snat 10.1.100.11:5050->172.16.200.44:5020(172.16.200.155:5050)
misc=0 policy_id=10 pol_uuid_idx=665 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
hrx info: valid=1/1, qid=4/6, npuid=1/1, sublink=0/0

session info: proto=17 proto_state=00 duration=454 expire=104 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=6
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=dirty may_dirty synced f16
statistic(bytes/packets/allow_err): org=263648/176/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=172.16.200.55/0.0.0.0
hook=post dir=org act=snat 10.1.100.11:5050->172.16.200.55:5001(172.16.200.155:5050)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.155:5050(10.1.100.11:5050)
misc=0 policy_id=10 pol_uuid_idx=665 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
hrx info: valid=1/0, qid=7/0, npuid=1/0, sublink=0/0
total session: 2
```

And these sessions are synchronized to the backup or secondary FortiGate:

```
diag sys session list


session info: proto=17 proto_state=00 duration=443 expire=1463 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
master session: proto=17 10.1.100.11:5050->172.16.200.55:5001
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=255/255
state=syn_ses intree
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=7->11/11->7 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=dnat 172.16.200.44:5020->172.16.200.155:5050(10.1.100.11:5050)
hook=post dir=reply act=snat 10.1.100.11:5050->172.16.200.44:5020(172.16.200.155:5050)
misc=0 policy_id=10 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x4000000
```

```
no_ofld_reason:  npu-flag-off
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0

session info: proto=17 proto_state=00 duration=443 expire=6 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=5
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=dirty may_dirty syn_ses
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.11:5050->172.16.200.55:5001(172.16.200.155:5050)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.155:5050(10.1.100.11:5050)
misc=0 policy_id=10 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x4000000
no_ofld_reason:  npu-flag-off
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0
total session: 2
```

Because these sessions are synchronized, after an HA failover, Client1 and Client2 can continue communicating with the server.

Enhanced FGCP monitoring with interface group awareness

Enhances HA monitoring by allowing interfaces to be grouped and monitored collectively. Administrators can now define failover behavior based on group status instead of individual interfaces, improving stability in complex topologies and reducing unnecessary failover events.

New options are available:

```
config system ha
    config link-group
        edit <name>
            set *member <string>
            set min-members <integer>
        next
    end
    set link-group-monitor <string>
end
```

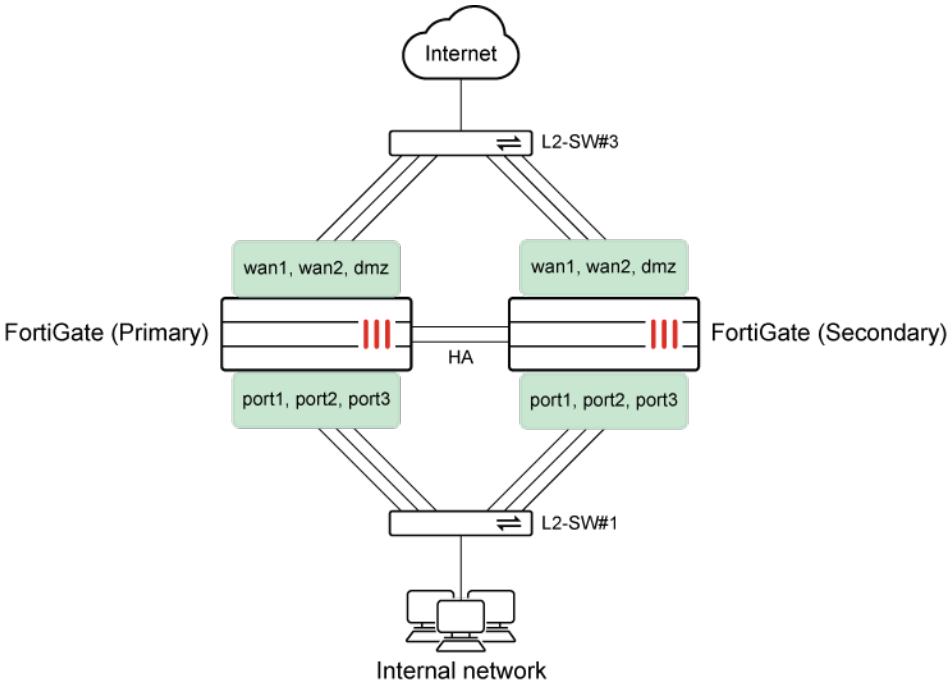
Option	Description
`config link-group`	Configure link group table.

Option	Description
`edit <name>`	Name of the link group.
`*member <string>`	Member interfaces in this link group.
`min-members <integer>`	Minimum number of members that must be up before this link group is considered up. Enter an integer value from 1 to 64 (default = 1).
`link-group-monitor <string>`	Link groups to check for port monitoring.

```
config system ha
    set vcluster-status enable
    config vcluster
        edit <id>
            set link-group-monitor <group_1>, ... [group_n]
        next
    end
end
```

Option	Description
`link-group-monitor <group_1>, ... [group_n]`	Enter the name of the link groups to check for port monitoring.

Example



In this HA example, two link monitor groups are created with multiple group members:

- The internal group contains the following members: port1, port2, port3, and a minimum of one member must be up by default for the group to be up.
- The outgoing group contains the following members: wan1, wan2 and dmz, and a minimum of two members must be up for the group to be up.

The groups are monitored. When one member of each group goes down, no failover occurs because enough group members are up for the groups to be up. However, when two members of the outgoing group go down, the minimum number of up members is not met, and FortiGate fails over to the secondary unit in the cluster.

To configure link group monitoring:

1. On the primary FortiGate in the HA cluster, configure link groups, and enable link group monitoring:

   Two link monitor groups are created: internal and outgoing.

   The internal group is up when at least one group member is up, which is the default setting. The outgoing group is up when at least two group members are up (`set min-members 2`).

   ```
   config system ha
       config link-group
           edit "internal"
               set member "port1" "port2" "port3"
           next
           edit "outgoing"
               set member "wan1" "wan2" "dmz"
               set min-members 2
           next
       end
       set override enable
       set priority 200
       set link-group-monitor "internal" "outgoing"
   end
   ```

2. On the primary FortiGate, view the vcluster information:

   ```
   (Primary)# diagnose sys ha dump-by vcluster
              HA information.

   vcluster_nr=1
   vcluster-1: state/o/chg_time=2(work)/1(helo)/1774297178(2026-03-23 13:19:38)
           pingsvr_flip_timeout/expire=3600s/0s
           link group:
           internal(prio=50,status=up), member status: port1(up) port2(up) port3(down)
           outgoing(prio=50,status=up), member status: wan1(up) wan2(up) dmz(down)
           'FG101FTK19003069': ha_prio/o=1/1, link_failure=0, pingsvr_failure=0, flag=0x00000000,
   mem_failover=0, uptime/reset_cnt=7/1
           'FG101FTK19003370': ha_prio/o=0/0, link_failure=0, pingsvr_failure=0, flag=0x00000001,
   mem_failover=0, uptime/reset_cnt=0/1
   ```

   In the `internal` group, two of the three ports are up (port1 and port2), and port3 is down. The link group is up because it meets the requirements of at least one up group member.

   In the `outgoing` group, two of the three ports are up (wan1 and wan2), and dmz is down. The link group is up because it meets the requirements of at least two up group members.

3. On the secondary FortiGate, view the vcluster information after wan2 and port2 go down:

```
(Secondary)# diagnose sys ha dump-by vcluster
            HA information.

vcluster_nr=1
vcluster-1: state/o/chg_time=3(standby)/2(work)/1774308016(2026-03-23 16:20:16)
        pingsvr_flip_timeout/expire=3600s/3561s
        link group:
        internal(prio=50,status=up), member status: port1(up) port2(down) port3(down)
        outgoing(prio=50,status=down), member status: wan1(up) wan2(down) dmz(down)
        'FG101FTK19003069': ha_prio/o=0/0, link_failure=0, pingsvr_failure=0, flag=0x00000001,
mem_failover=0, uptime/reset_cnt=10845/1
        'FG101FTK19003370': ha_prio/o=1/1, link_failure=50, pingsvr_failure=0,
flag=0x00000000, mem_failover=0, uptime/reset_cnt=0/2
```

In the `internal` link monitor group, two of the three ports are down (port2 and port3), and port1 is up. The link group is up because it meets the requirements of at least one up group member.

In the `outgoing` link monitor group, two of the three ports are down (wan2 and dmz), and wan1 is up. The link group is down because only one group member is up, which fails to meet the minimum requirement of at least two up group members.

The HA cluster fails over to the secondary unit.

4. On the secondary FortiGate, view the HA status:

```
(Secondary)# get sys ha status
HA Health Status: OK
Model: FortiGate-101F
Mode: HA A-P
Group Name: mmmlll
Group ID: 300
Debug: 0
Cluster Uptime: 0 days 3h:4m:16s
Cluster state change time: 2026-03-23 16:20:16
Primary selected using:
    <2026/03/23 16:20:16> vcluster-1: FG101FTK19003069 is selected as the primary because the
value of link-failure + pingsvr-failure is less than peer member FG101FTK19003370.
    <2026/03/23 13:19:38> vcluster-1: FG101FTK19003370 is selected as the primary because its
override priority is larger than peer member FG101FTK19003069.
ses_pickup: enable, ses_pickup_delay=disable
override: enable
Configuration Status:
    FG101FTK19003370(updated 0 seconds ago): in-sync
    FG101FTK19003370 chksum dump: 89 0f 35 97 80 c3 b0 74 62 eb 25 ab 43 25 eb dd
    FG101FTK19003069(updated 2 seconds ago): in-sync
    FG101FTK19003069 chksum dump: 89 0f 35 97 80 c3 b0 74 62 eb 25 ab 43 25 eb dd
System Usage stats:
    FG101FTK19003370(updated 0 seconds ago):
        sessions=0, average-cpu-user/nice/system/idle=4%/0%/0%/95%, memory=34%
    FG101FTK19003069(updated 2 seconds ago):
        sessions=6, average-cpu-user/nice/system/idle=5%/0%/7%/87%, memory=36%
HBDEV stats:
    FG101FTK19003370(updated 0 seconds ago):
        ha1: physical/1000auto, up, rx-bytes/packets/dropped/errors=705942/5298/0/0,
```

```
tx=28233056/55261/0/0
        ha2: physical/1000auto, up, rx-bytes/packets/dropped/errors=88653370/195779/0/0,
tx=40464030/84449/0/0
    FG101FTK19003069(updated 2 seconds ago):
        ha1: physical/1000auto, up, rx-bytes/packets/dropped/errors=465634559/867811/0/0,
tx=28092032/55289/0/0
        ha2: physical/1000auto, up, rx-bytes/packets/dropped/errors=97389244/200257/0/0,
tx=31745955/80008/0/0
number of member: 2
EEEEEEEEEEEEEEEEEE, FG101FTK19003370, HA cluster index = 0
FFFFF           , FG101FTK19003069, HA cluster index = 1
number of vcluster: 1
vcluster 1: standby 169.254.0.2
Secondary: FG101FTK19003370, HA operating index = 1
Primary: FG101FTK19003069, HA operating index = 0
2
```

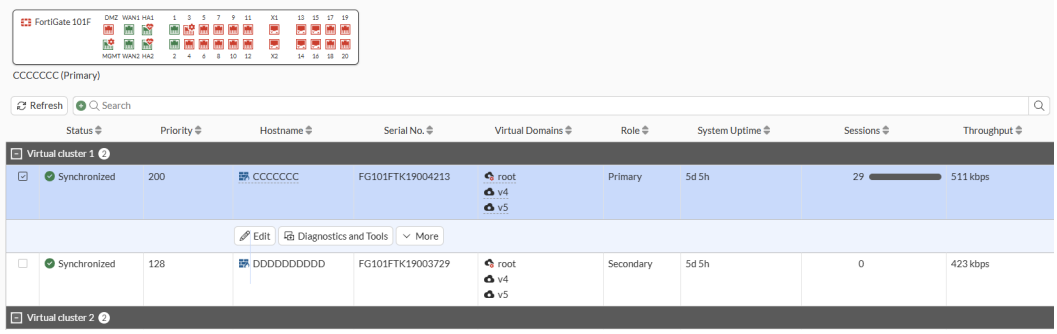
GUI support for HA actions, health status, and config diff

The FortiOS GUI now supports HA operations that were previously available only through the CLI, including resync, failover, and diagnostics.

The GUI also provides visual HA health details and a config diff view, making it easier to monitor status and identify configuration mismatches from the GUI.

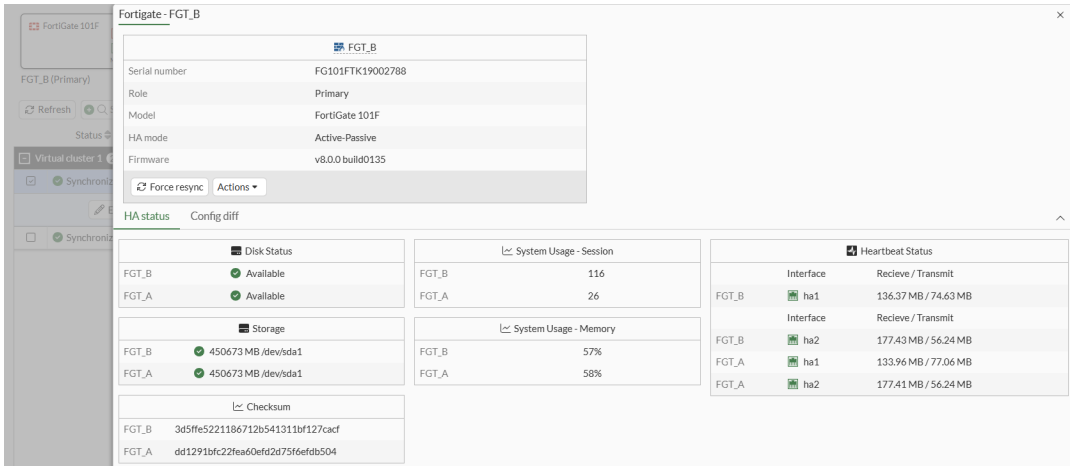
To view the GUI changes:

1. Go to the *System* > *HA* pane, and select a member. The *Diagnostics and Tools* button is displayed.

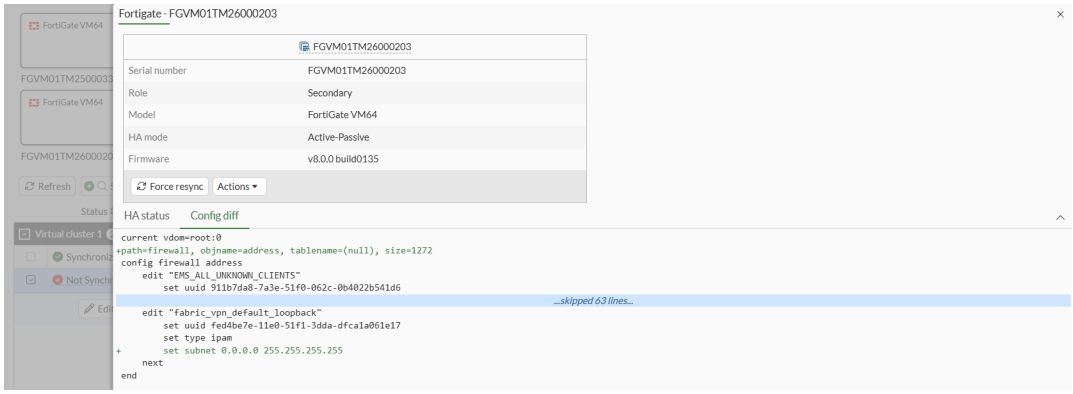


2. Click *Diagnostics and Tools* to display a slide-in pane.
   - At the top of the pane is a box of information about the device, such as serial number, role, model, HA mode, and firmware version.
   - A new *Force resync* button is available when the cluster is not synchronized. Click *Force resync* to synchronize the cluster. When the cluster is synchronized, the *Force resync* button is grayed out.
   - Under *Actions*, a new *Force HA failover* button is available to force a failover. See To force failover in the GUI: on page 486 for details about this process.
   - The following tabs of information are also available: *HA status* and *Config diff*. The *HA status* tab displays the following widgets of information:

- Disk Status
- Storage
- System Usage - Session
- System Usage - Memory
- Heartbeat Status



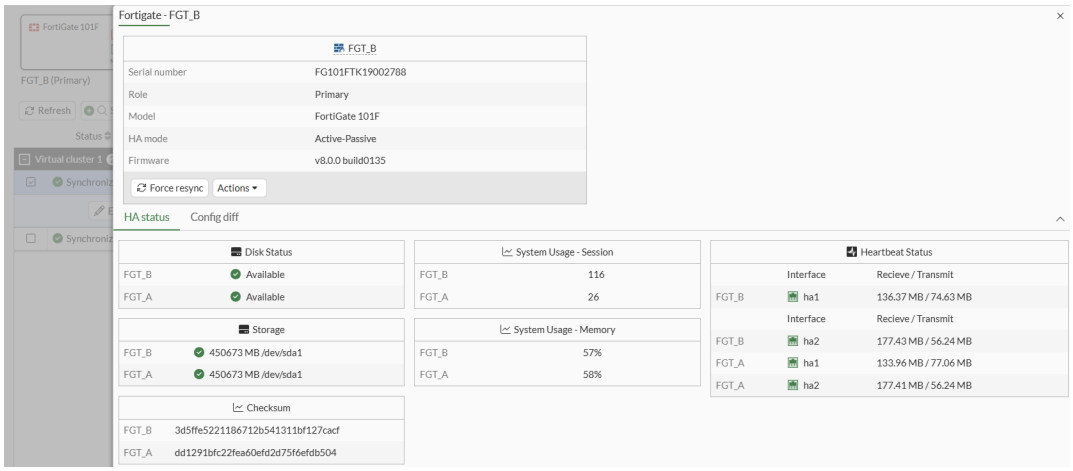
3. Click the *Config diff* tab to view the configuration difference between the secondary and primary FortiGates.



To force failover in the GUI:

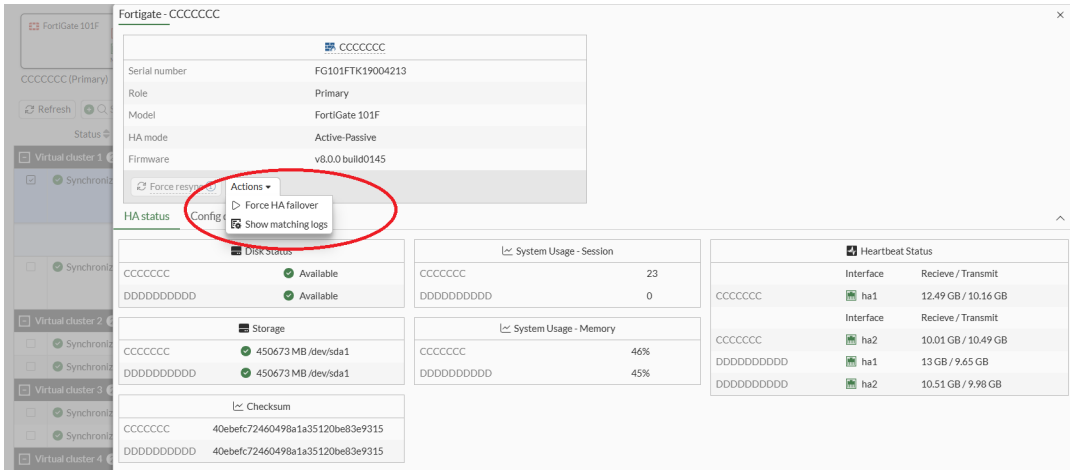
> It is recommended to only force failover during testing, troubleshooting, maintenance, and demonstrations. Do not force failover a live production environment outside of an active maintenance window. See also Force HA failover for testing and demonstrations for more information.

1. On the primary FortiGate, go to the *System > HA* pane, and select a member. The *Diagnostics and Tools* button is displayed.
2. Click *Diagnostics and Tools* to display a slide-in pane.



3. Click *Actions* and select *Force HA failover*.

   The cluster fails over to the secondary unit.



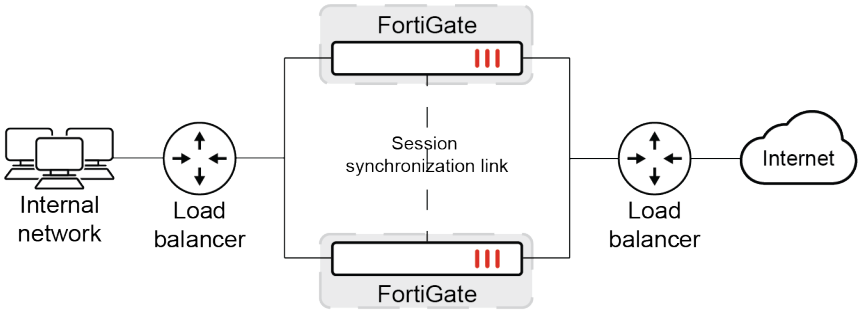
4. On the secondary unit (the new primary unit), log in to FortiOS, and run `execute ha failover unset <cluster_id>` to resume normal HA election behavior.

FGSP synchronizes firewall-authenticated users

FGSP synchronizes firewall-authenticated users between FGSP peers. If authenticated users sessions change to a different FGSP peer they do not have to re-authenticate. No special FGSP configuration is required to support this synchronization.

A firewall authenticated user is a user who has been authenticated by a user group that has been added to a firewall policy (see User groups.)

Example configuration



First FGSP peer configuration

This simple example includes setting up a basic FCSP configuration with two FGSP peers. The configuration uses local firewall users as the example authentication configuration.

```
config system standalone-cluster
    set standalone-group-id 222
    set group-member-id 8
    config cluster-peer
        edit 1
            set peerip 10.2.2.2
        next
    end
end
```

```
diagnose sys ha fgsp-zone
Local standalone-member-id: 8
Primary FGSP peer_num = 1, ver=3
        peer[1]: standalone-member-id=9, IP=10.2.2.2,vd=root, serial_no=FG101FTK19002788, prio=1,
rsso_sync_done=0
```

Second FGSP peer configuration

```
config system standalone-cluster
    set standalone-group-id 222
    set group-member-id 9
    config cluster-peer
        edit 1
            set peerip 10.2.2.1
        next
    end
end
```

```
diagnose sys ha fgsp-zone
Local standalone-member-id: 9
Primary FGSP peer_num = 1, ver=6
```

```
       peer[1]: standalone-member-id=8, IP=10.2.2.1,vd=root, serial_no=FG101FTK19003737, prio=1,
rsso_sync_done=0
```

Local user and firewall configuration

The following local user configuration must be the same for all FGSP peers:

```
config user local
    edit "tt"
        set type password
        set passwd *
    next
end
```

```
config user group
    edit "g1"
        set member "tt"
    next
end
```

```
config firewall policy
    edit 1
        set srcintf "port1"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set groups "g1" <-------------------- local user group
    next
```

Generate HTTP traffic

The peer FortiGate that processes the traffic could have the following session information:

```
diagnose firewall auth list

10.1.100.24, tt
        src_mac: 00:0c:29:a7:3e:3b
        type: fw, id: 0, duration: 4984, idled: 8
        expire: 292, allow-idle: 300
        server: tt
        packets: in 2610 out 4784, bytes: in 321123 out 304220
        user_id: 16777218
        group_id: 2
        group_name: g1

----- 1 listed, 0 filtered ------
```

```
diagnose sys session list


session info: proto=6 proto_state=01 duration=9 expire=3590 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=8:0 policy_dir=0 tunnel=/ vlan_cos=0/255
user=tt auth_server=tt state=may_dirty npu authed synced acct-ext
statistic(bytes/packets/allow_err): org=92/2/1 reply=52/1/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=172.16.200.55/10.1.100.24
hook=post dir=org act=snat 10.1.100.24:50851->172.16.200.55:80(172.16.200.1:50851)
hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.1:50851(10.1.100.24:50851)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=4072 auth_info=2 chk_client_info=0 vd=0
serial=00005500 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=66/70, ipid=70/66, vlan=0x0000/0x0000
vlifid=70/66, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=4/7, ha_divert=0/0
hrx info: valid=1/1, qid=5/8, npuid=1/1, sublink=0/0
total session: 1
```

Similar sessions appear on the other peer FortiGate:

```
diagnose firewall auth list

10.1.100.24, tt
        src_mac: 00:0c:29:a7:3e:3b
        type: fw, id: 0, duration: 81, idled: 88
        expire: 212, allow-idle: 300
        flag(400): ha
        server: tt
        packets: in 0 out 0, bytes: in 0 out 0
        user_id: 16777218
        group_id:
        group_name:

----- 1 listed, 0 filtered ------
```

```
diagnose sys session list

session info: proto=6 proto_state=01 duration=14 expire=3585 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=8:0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=dirty may_dirty npu syn_ses
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
```

```
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.24:50851->172.16.200.55:80(172.16.200.1:50851)
hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.1:50851(10.1.100.24:50851)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=00005500 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=00000000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0, ha_divert=0/0
no_ofld_reason:
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0
total session: 1
```

After a failover

You can simulate a failover by restarting the peer FortiGate that is processing traffic. The authenticated user can continue to send traffic through the second peer FortiGate without re-authenticating.

```
diagnose firewall auth list

10.1.100.24, tt
        src_mac: 00:0c:29:a7:3e:3b
        type: fw, id: 0, duration: 81, idled: 88
        expire: 212, allow-idle: 300
        flag(400): ha
        server: tt
        packets: in 0 out 0, bytes: in 0 out 0
        user_id: 16777218
        group_id:
        group_name:

----- 1 listed, 0 filtered ------
```

```
diagnose sys session list

session info: proto=6 proto_state=01 duration=14 expire=3585 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=8:0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=dirty may_dirty npu syn_ses
statistic(bytes/packets/allow_err): org=230/5/0 reply=52/1/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.24:50851->172.16.200.55:80(172.16.200.1:50851)
hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.1:50851(10.1.100.24:50851)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=00005500 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
```

```
npu_state=00000000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0, ha_divert=0/0
no_ofld_reason:
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0
total session: 1
```

HA role shown in CLI prompt

The CLI prompt displays the current HA role of the device, such as Primary or Secondary, removing the need to check the status commands. The display automatically updates after failover, making role identification faster and more intuitive for users.

```
FGT_A(Primary) login: admin
Password:
Verifying password...

Welcome!

FGT_A(Primary) #
```

```
FGT_B(Secondary) login: admin
Password:
Verifying password...

Welcome!

FGT_B(Secondary) #
```

SNMP

This section includes information about SNMP related new features:

- SNMP support for new IP pool statistics OIDs on page 492

SNMP support for new IP pool statistics OIDs

SNMP support added for five new OIDs to monitor IP pool block size, port range, and client IP range, enabling enhanced visibility into IP pool configurations through SNMP queries.

To verify the supported OIDs:

1. Configure IP pools:

```
config firewall ippool
    edit "test-pba"
        set type port-block-allocation
        set startip 10.30.1.1
        set endip 10.30.100.254
        set block-size 256
        set num-blocks-per-user 128
        set permit-any-host enable
    next
    edit "fixed"
        set type fixed-port-range
        set startip 115.0.4.0
        set endip 115.0.4.255
        set startport 19000
        set endport 20000
        set source-startip 5.0.6.1
        set source-endip 5.0.6.100
    next
end
```

2. Configure SNMP:
   a. Configure interface access:

```
config system interface
    edit port1
        append allowaccess snmp
    next
end
```

   b. Configure the SNMP agent:

```
config system snmp sysinfo
    set status enable
    set description Branch
    set contact-info Jane Doe
    set location Burnaby
end
```

   c. Configure an SNMP Community:

```
config system snmp community
    edit 1
        set name "REGR-SYS"
        config hosts
            edit 1
                set ip 172.16.200.55 255.255.255.255
            next
        end
```

```
    next
end
```

3. List the IP pools:

```
# diagnose firewall ippool list
list ippool info:(vf=vdom1)
ippool test-pba: id=1, block-sz=256, num-block=128, fixed-port=no, use=2
        nat ip-range=10.30.1.1-10.30.100.254, port-range=5117-65533, num-pba-per-ip=236
        TCP/UDP/ICMP-session-quota=0/0/0
        clients=0, inuse-NAT-IPs=0, failed-clients=0
        total-PBAs=6041128, inuse-PBAs=0, expiring-PBAs=0, free-PBAs=100.00%
        allocate-PBA-times=0, reuse-PBA-times=0
ippool fixed: id=2, block-sz=1001, num-block=1, fixed-port=no, use=2
        nat ip-range=115.0.4.0-115.0.4.255, port-range=19000-20000, num-pba-per-ip=1
        source ip-range=5.0.6.1-5.0.6.100  deterministic NAT
        TCP/UDP/ICMP-session-quota=0/0/0
        clients=0, inuse-NAT-IPs=0, failed-clients=0
        total-PBAs=256, inuse-PBAs=0, expiring-PBAs=0, free-PBAs=100.00%
        allocate-PBA-times=0, reuse-PBA-times=0
```

4. Run SNMP queries from a client PC and confirm that they show the same values as the FortiGate.

```
pc05:~$ snmpwalk -v1 -c REGR-SYS 172.16.200.1 fgFwIppStatsBlockSize
FORTINET-FORTIGATE-MIB::fgFwIppStatsBlockSize.10.30.1.1.10.30.100.254.2 = Gauge32: 256

pc05:~$ snmpwalk -v1 -c REGR-SYS 172.16.200.1 fgFwIppStatsPortStart
FORTINET-FORTIGATE-MIB::fgFwIppStatsPortStart.10.30.1.1.10.30.100.254.2 = Gauge32: 5117

pc05:~$ snmpwalk -v1 -c REGR-SYS 172.16.200.1 fgFwIppStatsPortEnd
FORTINET-FORTIGATE-MIB::fgFwIppStatsPortEnd.10.30.1.1.10.30.100.254.2 = Gauge32: 65533

pc05:~$ snmpwalk -v1 -c REGR-SYS 172.16.200.1 fgFwIppStatsStartClientIP
FORTINET-FORTIGATE-MIB::fgFwIppStatsStartClientIP.115.0.4.0.115.0.4.255.2 = IpAddress: 5.0.6.1

pc05:~$ snmpwalk -v1 -c REGR-SYS 172.16.200.1 fgFwIppStatsEndClientIP
FORTINET-FORTIGATE-MIB::fgFwIppStatsEndClientIP.115.0.4.0.115.0.4.255.2 = IpAddress: 5.0.6.100
```

FortiGuard

This section includes information about FortiGuard related new features:

- Enhanced FortiGuard server selection and load balancing on page 495
- FortiGuard rating server prioritization improvements on page 503

Enhanced FortiGuard server selection and load balancing

This information is also available in the FortiOS 8.0 Administration Guide:

- FortiGuard server preference

FortiGuard connectivity has been improved. The default load balancing setting has changed from one to five servers, and GUI controls are available to configure server preference. Administrators can prioritize public or custom servers and choose multiple types of override server, including the previously CLI-only IoT device collection and virtual patching query options.

The default value changed from 1 to 5 servers:

```
config system fortiguard
    set load-balance-servers <integer>
end
```

Option	Description
`load-balance-servers <integer>`	Enter the number of servers to alternate between as first FortiGuard option (1 - 255, default = 5). After upgrade to FortiOS 8.0.0, the new default value takes effect when the field was not set prior to upgrade. For example, the previous default of `set load-balance-servers 1` changes to `set load-balance-servers 5` after upgrade. If the field was set prior to upgrade, the setting remains unchanged after upgrade. For example, `set load-balance-servers 3` remains `set load-balance-servers 3` after upgrade.

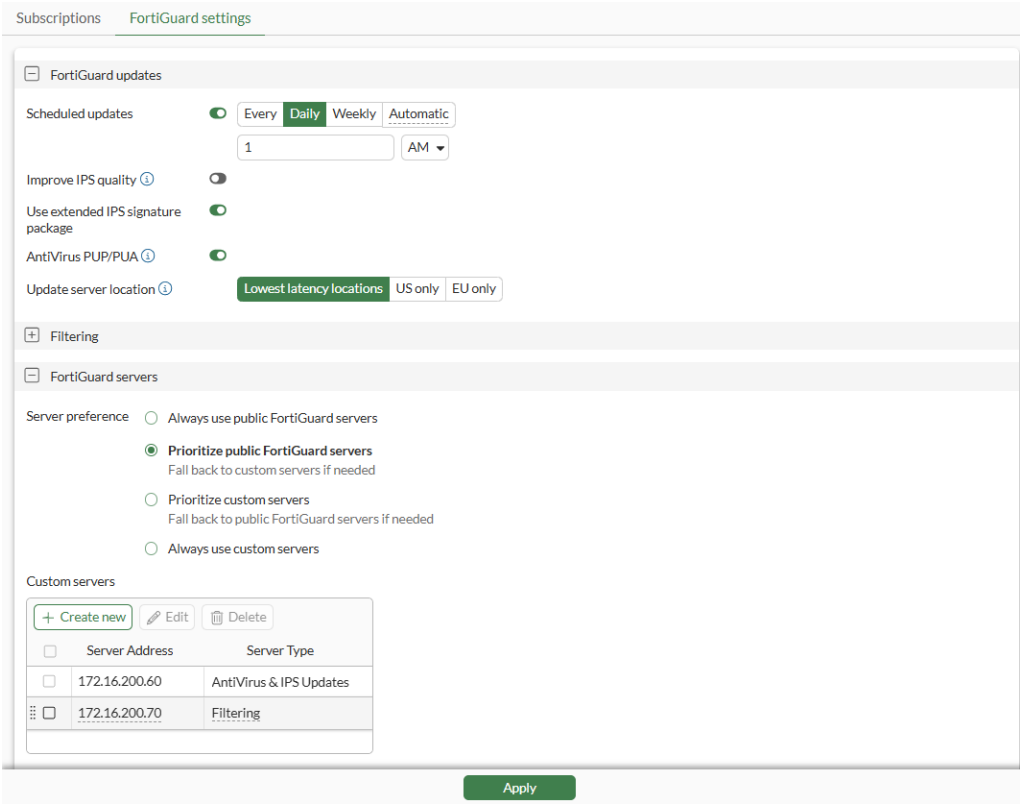
A new option is available:

```
config system central-management
    set use-default-servers-as main {enable | disable}
end
```

Option	Description
`use-default-servers-as main {enable | disable}`	Enable/disable use of the public FortiGuard servers as main servers: • `enable`: Enable use of the public FortiGuard servers as main servers. • `disable`: Disable use of the public FortiGuard servers as main servers.

On the *System > FortiGuard > FortiGuard settings* pane, updated *Server preference* options are available under *FortiGuard servers*:

- Always use public FortiGate servers
- Prioritize public FortiGuard servers
- Prioritize custom servers
- Always use custom servers



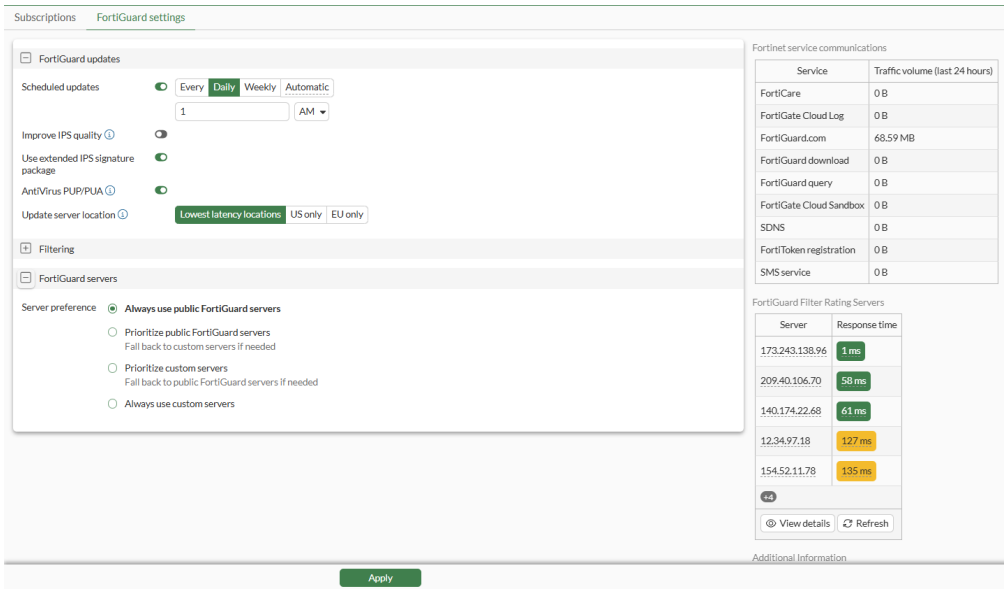
When creating a custom server, you can set *Type* to one or more types of override servers, including *IoT device collection* and *Virtual patching query*, which were previously only available in the CLI:

Example 1: always use public FortiGuard servers

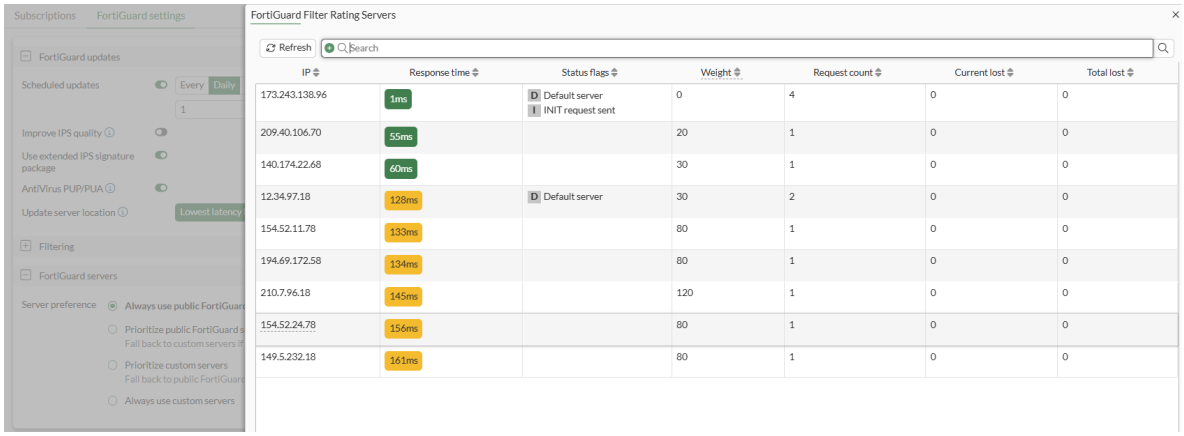
This example describes how to configure FortiGate to use only default FortiGuard servers. No custom servers are used.

To configure FortiGuard servers in the GUI:

1. Go to *System* > *FortiGuard* > *FortiGuard settings*.
2. Under *FortiGuard servers*, set *Server preference* to *Always use public FortiGuard servers*, and click *Apply*.



3. In the right-side gutter, under *FortiGuard Filter Rating Servers*, click *View Details*.
   Only default FortiGuard servers are used. No custom servers are used.



To configure FortiGuard servers in the CLI:

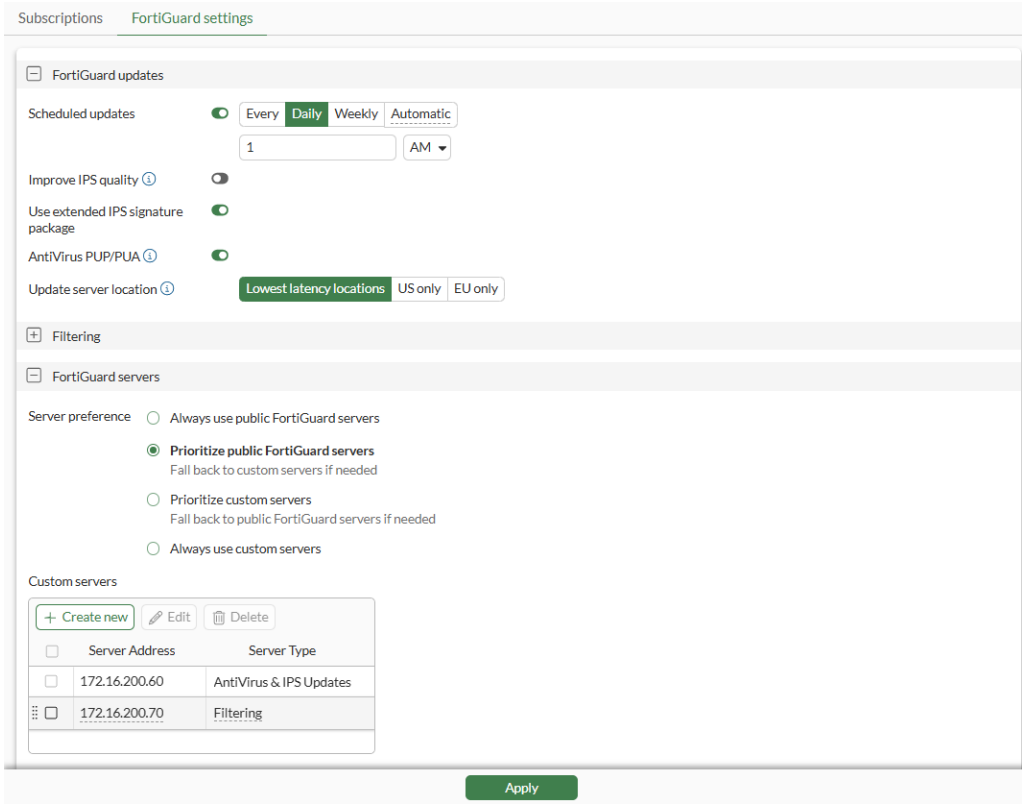
```
config system central-management
    set include-default-servers enable
    set use-default-servers-as-main enable
end
```

Example 2: prioritizing public FortiGuard servers

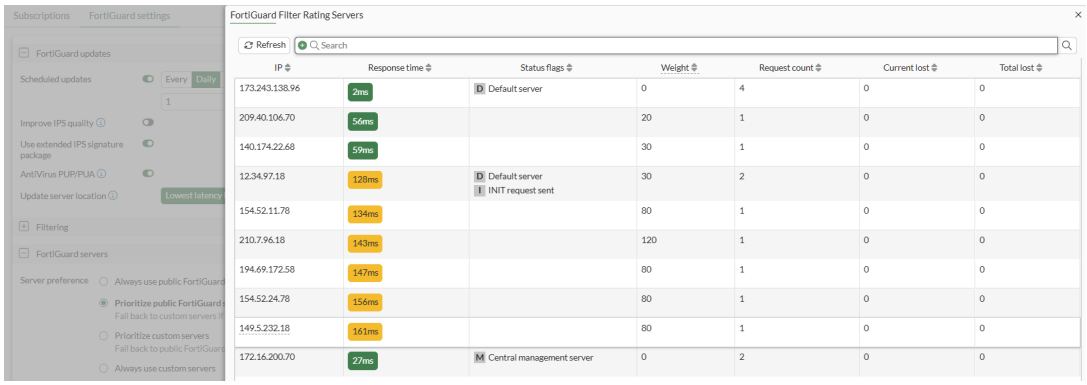
This example describes how to configure FortiGate to prioritize public FortiGuard servers and fall back to custom servers if needed.

To configure FortiGuard servers in the GUI:

1. Go to *System* > *FortiGuard* > *FortiGuard settings*.
2. Under *FortiGuard servers*, set *Server preference* to *Prioritize FortiGuard servers*.
3. Under *Custom servers*, click *Create new*, and add custom servers.
   a. Click *Create new*.
   b. Set *Address Type*.
   c. In *Address*, enter the server address.
   d. In the *Type* box, select one or more types of server.
   e. Click *OK*. The custom server is displayed.
   f. Repeat this procedure to add additional custom servers.



4. Click *Apply*.
5. In the right-side gutter, under *FortiGuard Filter Rating Servers*, click *View Details*.
   The custom server (172.16.200.70) is below the default FortiGuard servers.



To configure FortiGuard servers in the CLI:

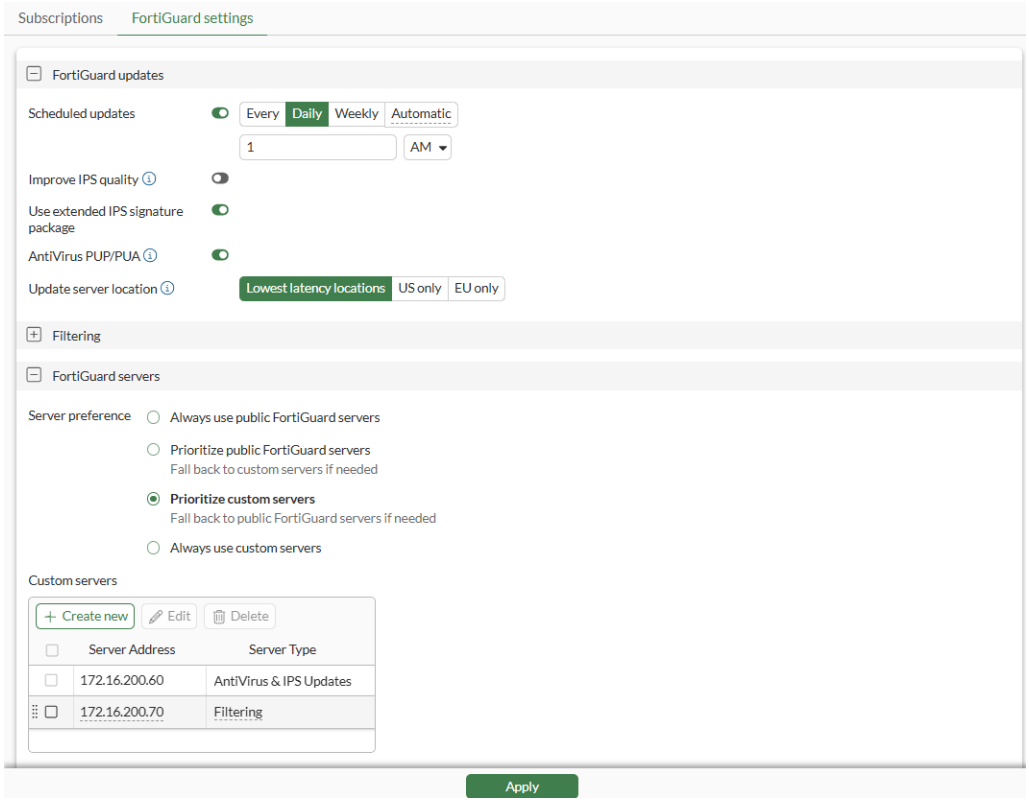
```
config system central-management
    config server-list
        edit 1
            set server-type update
            set addr-type ipv4
            set server-address 172.16.200.60
        next
        edit 2
            set server-type rating
            set addr-type ipv4
            set server-address 172.16.200.70
        next
    end
    set include-default-servers enable
    set use-default-servers-as-main enable
end
```

Example 3: prioritizing custom servers

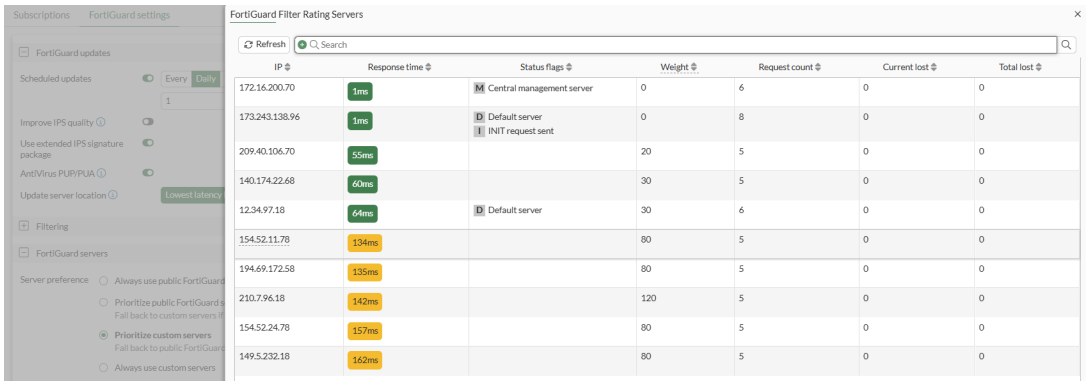
This example describes how to configure FortiGate to prioritize custom servers and fall back to public FortiGuard servers if needed.

To configure FortiGuard servers in the GUI:

1. Go to *System* > *FortiGuard* > *FortiGuard settings*.
2. Under *FortiGuard servers*, set *Server preference* to *Prioritize custom servers*.
3. Under *Custom servers*, add custom servers:
   a. Click *Create new*.
   b. Set *Address Type*.
   c. In *Address*, enter the server address.
   d. In the *Type* box, select one or more types of server.
   e. Click *OK*. The custom server is displayed.
   f. Repeat this procedure to add additional custom servers.



4. In the right-side gutter, under *FortiGuard Filter Rating Servers*, click *View Details*.
   The custom server (172.16.200.70) is above the default FortiGuard servers.



To configure FortiGuard servers in the CLI:

```
config system central-management
    config server-list
        edit 1
            set server-type update
            set addr-type ipv4
            set server-address 172.16.200.60
        next
        edit 2
            set server-type rating
```

```
            set addr-type ipv4
            set server-address 172.16.200.70
        next
    end
    set include-default-servers enable
    set use-default-servers-as-main disable
end
```

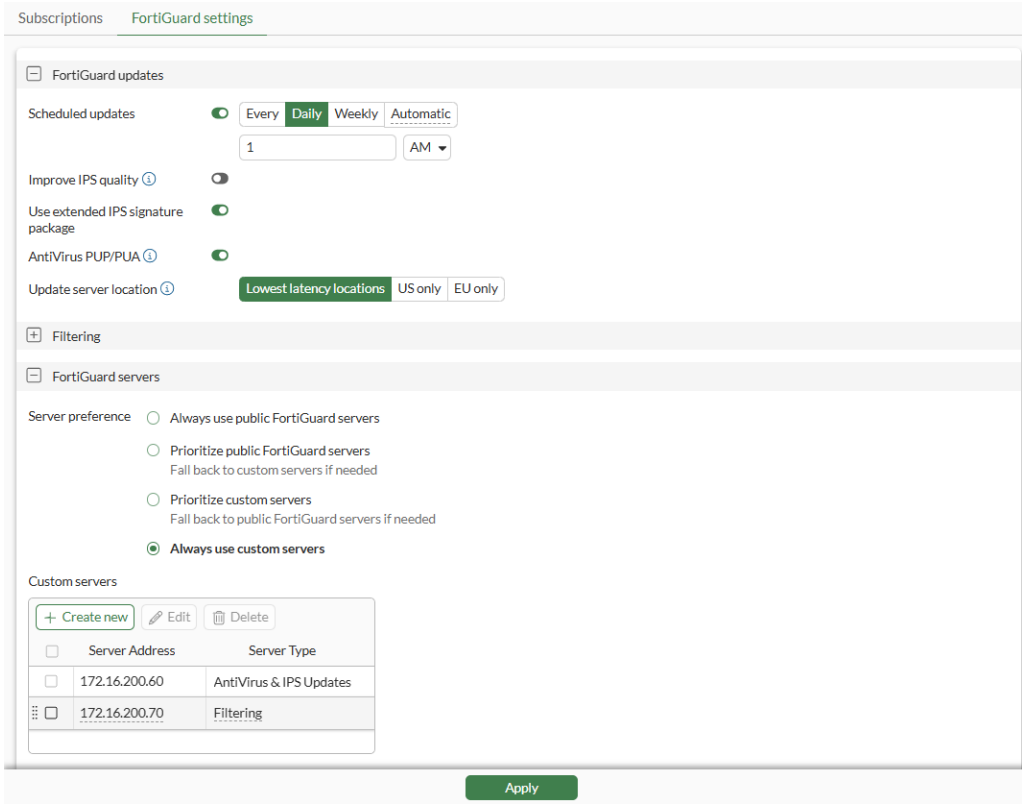
Example 4: always using custom servers

This example describes how to configure FortiGate to always use custom servers.

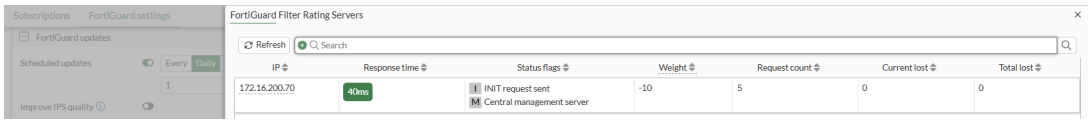
When central management is enabled and set to *FortiGate Cloud* in the GUI (`fortiguard` in the CLI), the *Always use custom servers* option is hidden.

To configure FortiGuard servers in the GUI:

1. Go to *System* > *FortiGuard* > *FortiGuard settings*.
2. Under *FortiGuard servers*, set *Server preference* to *Always use custom servers*.
3. Under *Custom servers*, add custom servers:
   a. Click *Create new*.
   b. Set *Address Type*.
   c. In *Address*, enter the server address.
   d. In the *Type* box, select one or more types of server.
   e. Click *OK*. The custom server is displayed.
   f. Repeat this procedure to add additional custom servers.



4. In the right-side gutter, under *FortiGuard Filter Rating Servers*, click *View Details*.
   Only custom servers are used. No default FortiGuard servers are used.



To configure FortiGuard servers in the CLI:

```
config system central-management
    config server-list
        edit 1
            set server-type update
            set addr-type ipv4
            set server-address 172.16.200.60
        next
        edit 2
            set server-type rating
            set addr-type ipv4
            set server-address 172.16.200.70
        next
    end
    set include-default-servers disable
end
```

FortiGuard rating server prioritization improvements

This information is also available in the FortiOS 8.0 Administration Guide:

- FortiGuard filter rating servers

FortiGuard rating servers are now selected by fastest round-trip time (RTT) instead of timezone-based weight, ensuring the closest, fastest FortiGuard servers are used. The *System > Settings > FortiGuard > FortiGuard settings* page also changes from a service-based status to a server-specific status with an updated *FortiGuard Filter Ratings Servers* table and a new *View details* button. The GUI updates provide granular visibility into server IPs, response times, and packet loss, making troubleshooting more efficient.

Before this enhancement, the output of `diagnose debug rating` indicated FortiGuard servers were sorted by `Weight` over `RTT` with the lower weight servers at the top.

```
# diagnose debug rating
Locale       : english

Service      : Web-filter
Status       : Enable
License      : Contract

Service      : Antispam
Status       : Disable

Service      : Virus Outbreak Prevention
Status       : Disable

Num. of servers : 9
Protocol        : https
Port            : 443
Anycast         : Disable
Default servers : Included

-=- Server List (Wed Feb 25 02:15:16 2026) -=-

IP                                            Weight    RTT Flags   TZ   FortiGuard-requests
 Curr Lost Total Lost            Updated Time
154.52.11.78                                       0    541          0                    1
    0          0 Wed Feb 25 02:13:42 2026
194.69.172.58                                      0    545          0                    1
    0          0 Wed Feb 25 02:13:42 2026
154.52.24.78                                       0    639          0                    1
    0          0 Wed Feb 25 02:13:43 2026
149.5.232.18                                       0    651          0                    1
    0          0 Wed Feb 25 02:13:43 2026
12.34.97.18                                       50     64 DI      -5                    4
    0          0 Wed Feb 25 02:13:42 2026
140.174.22.68                                     50    242         -5                    1
    0          0 Wed Feb 25 02:13:42 2026
209.40.106.70                                     60    230         -6                    1
    0          0 Wed Feb 25 02:13:42 2026
```

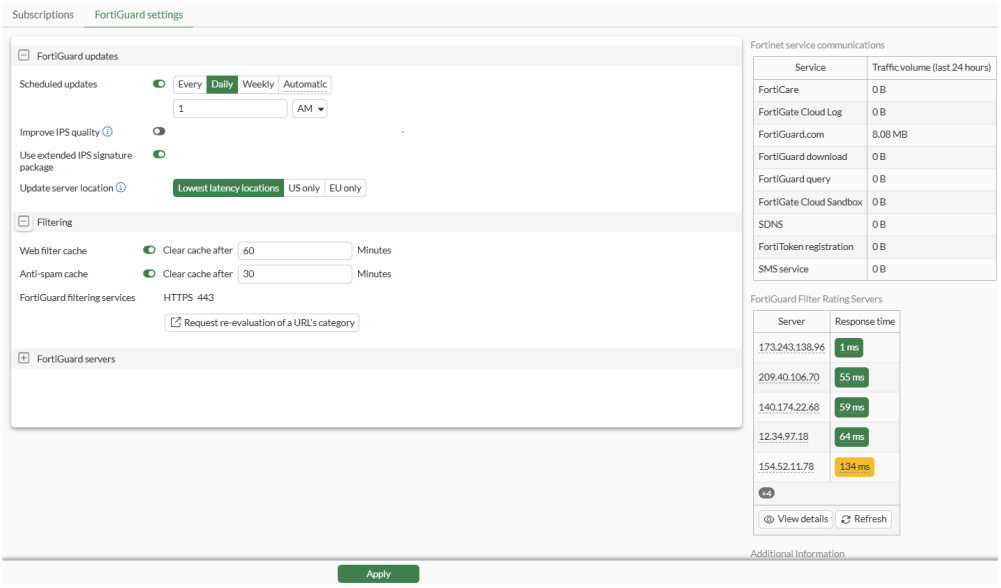
```
173.243.138.96                                    80      1 D       -8                    3
    0          0 Wed Feb 25 02:13:42 2026
210.7.96.18                                       90    581          9                    1
    0          0 Wed Feb 25 02:13:43 2026
```

With this enhancement, the output of `diagnose debug rating` indicates FortiGuard servers are sorted by `RTT`, and `Weight` is ignored. The lowest RTT servers are at the top.

```
# diagnose debug rating
Locale       : english

Service      : Web-filter
Status       : Enable
License      : Contract

Service      : Antispam
Status       : Disable

Service      : Virus Outbreak Prevention
Status       : Disable

Num. of servers : 9
Protocol        : https
Port            : 443
Anycast         : Disable
Default servers : Included

-=- Server List (Wed Feb 25 00:54:45 2026) -=-

IP                                            Weight    RTT Flags   TZ   FortiGuard-requests
 Curr Lost Total Lost             Updated Time
173.243.138.96                                    80      2 D       -8                   58
    0          0 Wed Feb 25 00:54:38 2026
209.40.106.70                                     60     56         -6                   56
    0          0 Wed Feb 25 00:54:39 2026
140.174.22.68                                     50     61         -5                   56
    0          0 Wed Feb 25 00:54:39 2026
12.34.97.18                                       50     63 DI      -5                   59
    0          0 Wed Feb 25 00:54:39 2026
154.52.11.78                                       0    134          0                   56
    0          0 Wed Feb 25 00:54:39 2026
194.69.172.58                                      0    137          0                   56
    0          0 Wed Feb 25 00:54:39 2026
210.7.96.18                                       90    142          9                   56
    0          0 Wed Feb 25 00:54:39 2026
154.52.24.78                                       0    156          0                   56
    0          0 Wed Feb 25 00:54:39 2026
149.5.232.18                                       0    161          0                   56
    0          0 Wed Feb 25 00:54:39 2026
```

To view connectivity status:

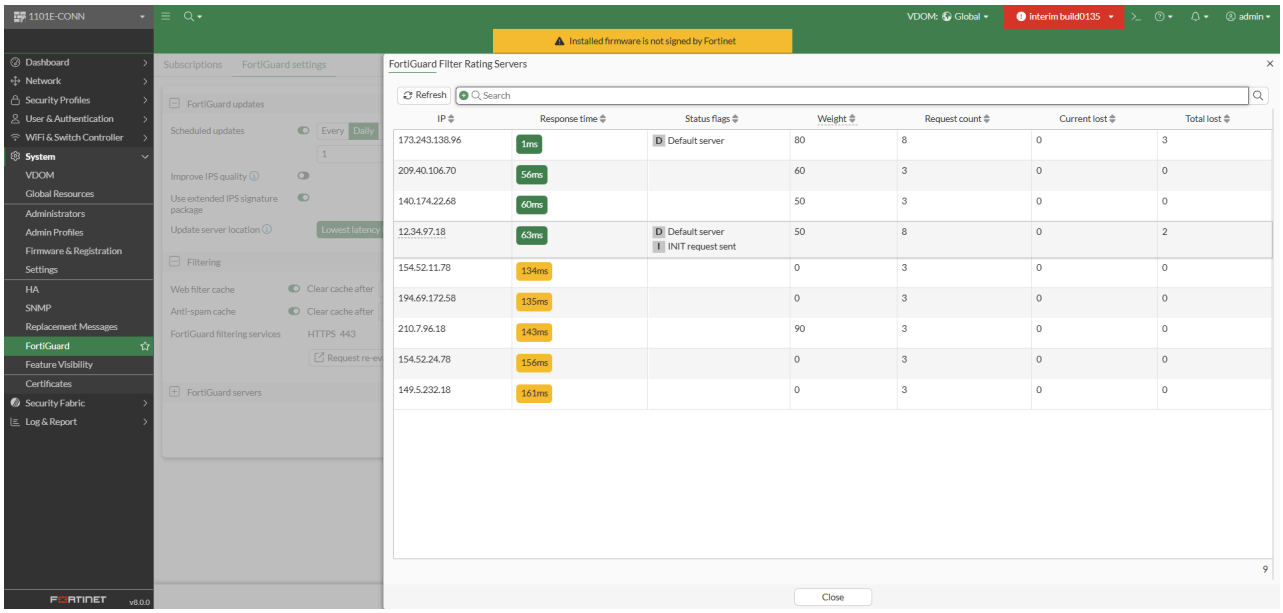
1. Go to *System* > *FortiGuard* > *FortiGuard* settings.



2. In the right-side gutter, the *FortiGuard Filter Rating Servers* displays connectivity information for each server in the following columns:
   - *Server*: Displays the IP address for each server.
   - *Response time*: Displays a color-coded round-trip time.

   The colors in the *Response time* column indicate latency:
   - Green: Indicates less than or equal to 100 ms.
   - Yellow: Indicates 101 - 200 ms.
   - Red: Indicates 201 ms and higher.
3. Click *Refresh* for the latest connectivity information.
4. Click *View Details*. The following columns of information are displayed:

   The same information can be viewed with the `diagnose debug rating` command.



5. Click *Close*.

API

This section includes information about API related new features:

- API support to manage firmware for FortiGate extended devices on page 506

API support to manage firmware for FortiGate extended devices

Firmware upgrades for extended FortiGate devices (FortiSwitch, FortiAP, FortiExtender) can now be controlled using several new REST APIs on the FortiGate from a RESTful API Client, such as FortiManager.

```
[POST] /api/v2/monitor/firmware/extension-device/upload-image
```

```
[POST] /api/v2/monitor/firmware/extension-device/download-image
```

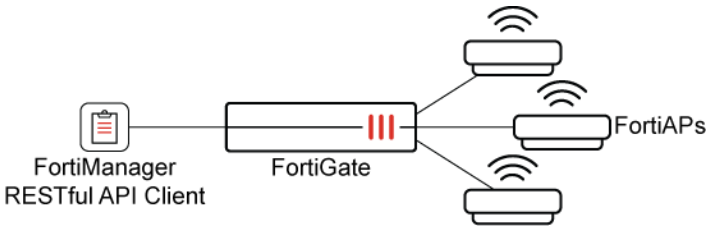
```
[GET] /api/v2/monitor/firmware/extension-device/image-list
```

```
[DELETE] /api/v2/monitor/firmware/extension-device/image
```

```
[POST] /api/v2/monitor/firmware/extension-device/trigger-update
```

```
[GET] /api/v2/monitor/firmware/extension-device/update-status
```

Example



Upload a firmware image for an extended device using FortiManager or a RESTful API client, and the FortiGate will store the image locally and return an image_id. The FortiManager or RESTful API client can then retrieve the status of the stored firmware image, download or delete the image, and trigger an update for the extended device.

To use the API to manage firmware for extended devices:

1. Upload an extended device image to get an image_id:

Required parameters	Description
type	Extension device type: `fortiap`, `fortiswitch`, or `fortiextender`.

upload_file	The image .out file to be uploaded
	In this example, the FortiAP image file will be sent to FortiGate by the Client in the following `Content-Type`:
	`Content-Type: multipart/form-data; boundary=----------------------139324056057982302494471`
	`Content-Length: 47943224`

Returns an object with the following fields:

Field	Description
major	Parsed major version of the firmware file.
minor	Parsed minor version of the firmware file.
patch	Parsed patch version of the firmware file.
build	Parsed build number of the firmware file.
type	Type of extension-device.
platform	Platform that the firmware file is for.
image_id	The firmware file identifier.

Example

```
POST /api/v2/monitor/firmware/extension-device/upload-image
```

Form-data

```
{
"type": "fortiap",
"upload_file": "FAP_831F-v7-build1032-FORTINET.out"
}
```

Returns 200 OK with the specific `image_id`.

```
{
    "http_method": "POST",
    "results": {
        "image_id": "EXTDEV_IMG_lhYx96",
        "major": 7,
        "minor": 6,
        "patch": 3,
        "build": 1032,
        "platform": "FP831F",
        "type": "fortiap"
    },
    "path": "firmware",
    "name": "extension-device",
    "action": "upload-image",
...
}
```

2. Download the image by specifying the image_id:

Required parameters	Description
type	Extension device type: `fortiap`, `fortiswitch`, or `fortiextender`.
image_id	Image ID to download.

Returns the same thing as `/api/v2/monitor/firmware/extension-device/upload-image`.

Example

```
POST /api/v2/monitor/firmware/extension-device/download-image
```

Payload

```
{
"type": "fortiap",
"image_id": "EXTDEV_IMG_lhYx96"
}
```

Returns 200 OK with the downloaded image.

3. Get a list of uploaded extension-device firmware files:
Returns an array of objects with fields that correspond to the returned object of the upload-image endpoint.

Example

```
GET /api/v2/monitor/firmware/extension-device/image-list
```

Returns 200 OK with `image_ids` of uploaded images

```
{
    "http_method": "GET",
    "results": [
        {
            "image_id": "EXTDEV_IMG_lhYx96",
            "type": "fortiap",
           ...
        }
    ],
    "path": "firmware",
    "name": "extension-device",
    "action": "image-list",
...
}
```

4. Delete an uploaded firmware file using its image ID:

Required parameters	Description
image_id	Image ID of the firmware file to be deleted. It can be multiple IDs, such as `?image_id=image_id1&image_id=image_id2&image_id=image_id3`.

Example

```
DELETE /api/v2/monitor/firmware/extension-device/image?image-id=EXTDEV_IMG_lhYx96
```

Returns 200 OK

```
{
    "http_method": "DELETE",
    "status": "success",
    "http_status": 200,
   ...
}
```

5. Trigger update for the uploaded extension-device with image_id, serial, and type:

Required parameters	Description
image_id	The image ID to use for upgrading.
serial	Serial number of the extension device to be upgraded. It can be multiple values, either by specifying query parameters, `?serial=serial1&serial=serial2&serial=serial3`, or by creating a POST body, `{ serial: ['serial1', 'serial2', 'serial'] }`.
type	Type of extension-device.

Returns an object where the key is the serial number, and the value is any of the following:

Value	Description
device not found	Extension device not found.
success	Successfully added an entry to the `system.device-upgrade` table. Upgrade should be on the way.
failed	Encountered an issue adding a new entry to the `system.device-upgrade` table. Will not be upgraded.
device already upgrading	An entry already exists in the `system.device-upgrade` table that is not in the End state (upgrade pending).
platform mismatch	If the firmware referenced by the specified image ID does not match the platform of the extension device.
wrong device type	If triggering the update for the wrong device type.

Example

```
POST /api/v2/monitor/firmware/extension-device/trigger-update
```

Payload

```
{
"image_id": "EXTDEV_IMG_lhYx96",
"serial": "FP831FTF23006968",
"type": "fortiap"
}
```

Returns 200 OK

```
{
    "http_method": "POST",
    "results": {
        "FP831FTF23006968": "success"
    },
    "vdom": "vdom1",
    "path": "firmware",
    "name": "extension-device",
    "action": "trigger-update",
    "status": "success",
    "serial": "FG4H1FT922902448",
...
}
```

6. Get update status, returns the status of the upgrades:

Optional parameter	Description
serial	Serial number of the extension device whose status will be returned. It can be multiple values, such as `?serial=serial1&serial=serial2` If not defined, it will return the status of all extension-device upgrades.

A list of devices and other information is returned.

Returned parameters	Description
serial	Serial number of the extension device.
image_id	The image ID that was used in the upgrade.
status	The status of the upgrade; similar to the status defined in the `system.device-upgrade`.

Example

```
GET /api/v2/monitor/firmware/extension-device/update-status
```

Returns 200 OK

```
{
    "http_method": "GET",
    "results": [
        {
            "image_id": "EXTDEV_IMG_lhYx96",
            "serial": "FP831FTF23006968",
            "status": "success"
        }
,
    "path": "firmware",
    "name": "extension-device",
    "action": "update-status",
    "status": "success",
    ...
}
```

Security Fabric

This section includes information about Security Fabric related new features:

- Fabric settings and connectors on page 512
- External SDN connectors on page 536

Fabric settings and connectors

This section includes information about Security Fabric settings and Fabric connector related new features:

- Cloud-based Fabric Feed synchronization on page 512
- Label-based sandbox exemptions for sensitive data control on page 517
- Simultaneous FortiSandbox Cloud and on-premise integration on page 520

Cloud-based Fabric Feed synchronization

Introducing Fabric Feed, an improvement to the existing external feeds feature, which is formerly known as threat feeds. This update lets FortiAnalyzer upload Fabric Feeds directly to the Cloud AMQP server. Once uploaded, these feeds are automatically synchronized with FortiGate devices linked to the same FortiCare account. This enhancement introduces a centralized, cloud-native approach to distributing feed intelligence, improving operational efficiency and consistency across the Security Fabric.

CLI syntax

The `config system external-resource` command includes new default external resources that are disabled by default and have `update-method` set to `fortimq`:

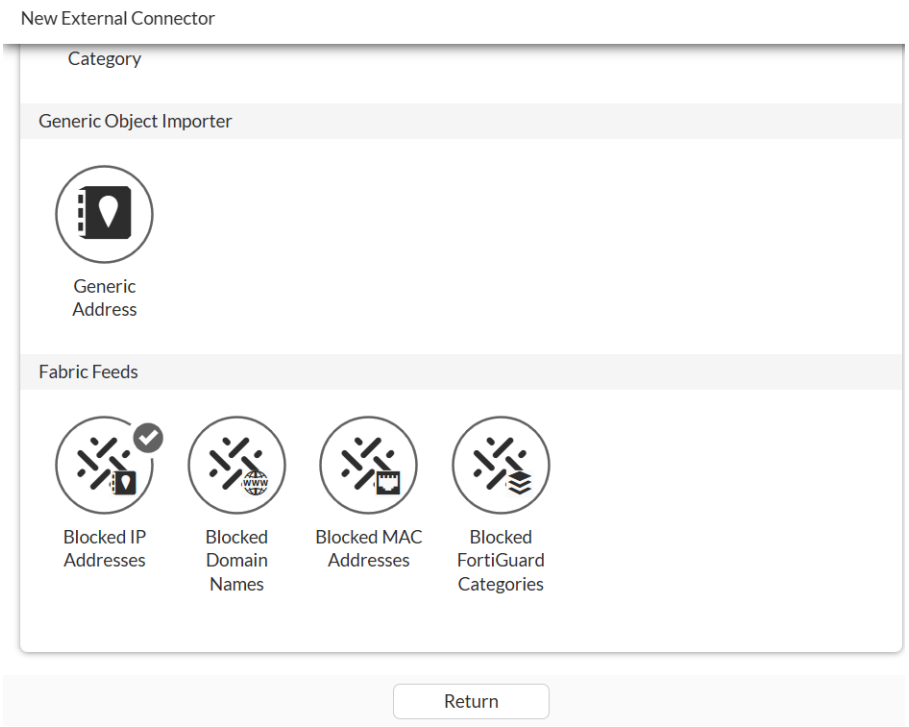
```
config system external-resource
    edit "IP_Blocked_by_Fabric_SecOps"
        set uuid 495e0cb0-3d94-51f0-7ec7-f0ff45079b48
        set status disable
        set type address
        set update-method fortimq
    next
    edit "Domain_Blocked_by_Fabric_SecOps"
        set uuid 495e126e-3d94-51f0-755a-1240f7ba7742
        set status disable
        set type domain
        set update-method fortimq
    next
```

```
    edit "URL_Blocked_by_Fabric_SecOps"
        set uuid 495e1430-3d94-51f0-b321-be7c2ba63e2c
        set status disable
        set update-method fortimq
    next
    edit "MAC_Blocked_by_Fabric_SecOps"
        set uuid 495e15ca-3d94-51f0-ff6a-5d03040d6215
        set status disable
        set type mac-address
        set update-method fortimq
    next
```

Option	Description
`edit <external resource name>`	Edit resource. The following default external resources are available and disabled by default: • IP_Blocked_by_Fabric_SecOps is type address. • Domain_Blocked_by_Fabric_SecOps is type domain. • URL_Blocked_by_Fabric_SecOps • MAC_Blocked_by_Fabric_SecOps is type mac-address. Each type of external resource can only be enabled once for each VDOM.
`status {enable | disable}`	Enable/disable user resource.
`update-method {feed | push | fortimq}`	External resource update method. • `feed`: FortiGate unit will pull update from the external resource. • `push`: External resource update is pushed to the FortiGate unit through the FortiGate unit's RESTAPI/CLI. • `fortimq`: External resource update is pushed to the FortiGate unit through the Fortinet Message Queue fabric.

GUI configuration

On the *Security Fabric > External Connectors* pane, click *Create New* to access and enable one or more Fabric Feeds.



Scope and limitations

FortiAnalyzer 8.0.0 and later is required.

Example

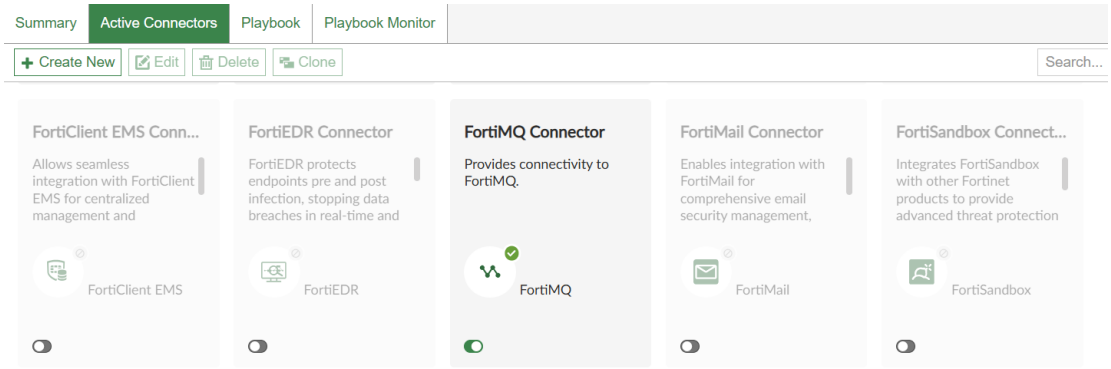
This example describes how to enable and use FortiMQ with FortiAnalyzer and FortiGate. Following is the high-level procedure:

1. On FortiAnalyzer, enable the FortiMQ connector and define indicators for the block list. See Enabling FortiMQ on FortiAnalyzer on page 515.
2. On FortiGate, enable Fabric Feeds using the GUI or CLI. See Enabling Fabric Feeds on FortiGate on page 515.
   After FortiGate synchronizes Fabric Feeds with FortiMQ server, the Fabric Feeds are ready to use. Apply Fabric Feeds to a firewall policy for use in blocking traffic and view the results in traffic logs.

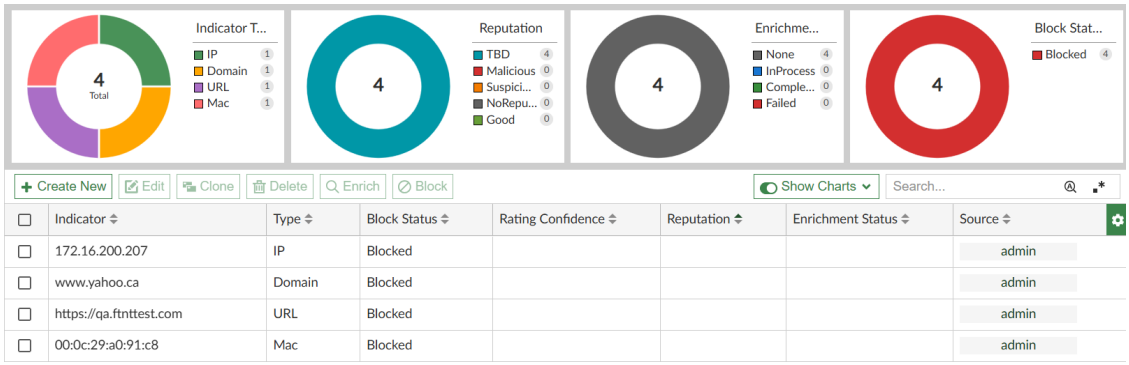
Enabling FortiMQ on FortiAnalyzer

To enable the FortiMQ connector on FortiAnalyzer:

1. On FortiAnalyzer, go to *Incidents & Events* > *Automation*, and enable the *FortiMQ Connector*.



2. Go to *Indicators* to define indicators and their block status. The following types of indicators are supported: Domain, URL, IP, and MAC address.



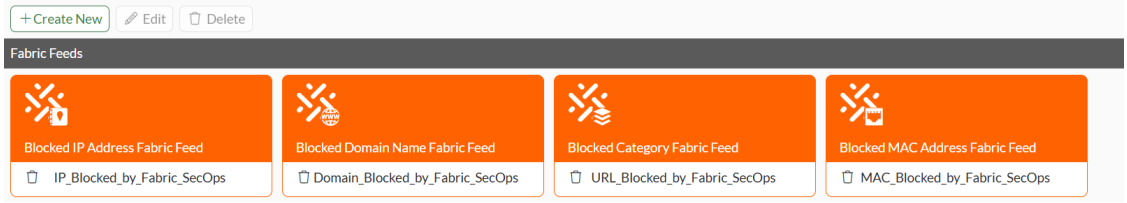
3. Go to FortiGate and enable Fabric Feeds using the GUI or CLI. See Enabling Fabric Feeds on FortiGate on page 515.

Enabling Fabric Feeds on FortiGate

After the FortiMQ connector is enabled on FortiAnalyzer and indicators defined, you must enable Fabric Feeds on FortiGate to use the FortiAnalyzer indicators. Fabric Feeds can be enabled using the the GUI or CLI.

To enable Fabric Feeds in the FortiGate GUI:

1. On FortiGate, enable Fabric Feeds:
   a. Go to *Security Fabric* > *External Connectors* pane, and click *Create New*.
   b. Under *Fabric Feeds*, select a Fabric Feed and click *OK* to enable it.
   c. Repeat this step to enable additional Fabric Feeds.
2. After synchronizing with the FortiMQ Cloud server, the Fabric Feeds are available on the *Security Fabric* > *External Connectors* pane and ready to use:



3. IP Address Fabric Feeds and MAC Address Fabric Feeds can be applied directly to a policy.

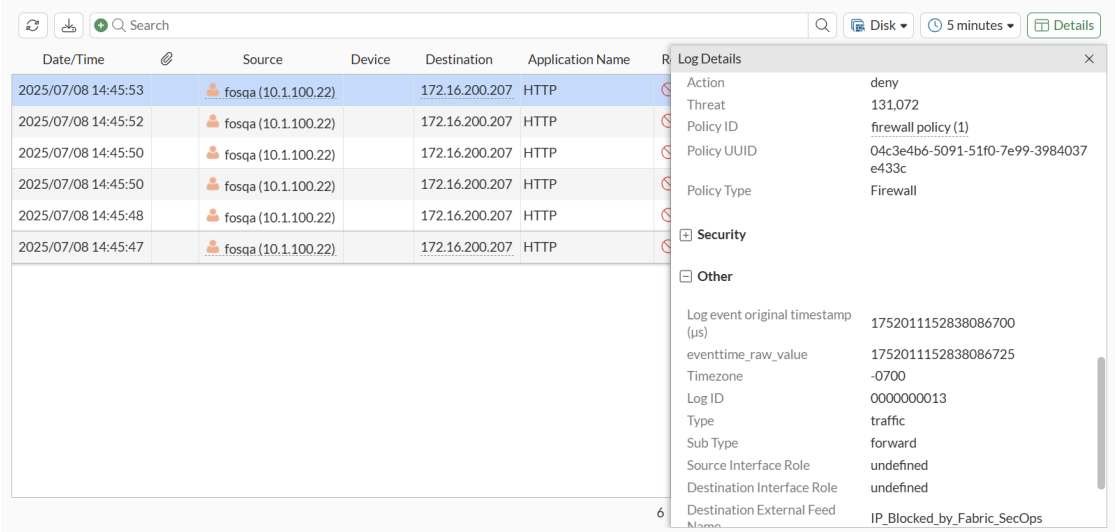
   For Domain Name Fabric Feeds, a DNS Filter profile is required.

   For Category Fabric Feeds, a Web Filter profile is required.

   In the Remote Categories section of the corresponding DNS or Web Filter profile, set the Fabric Feed action, such as Block, Monitor, or Warning, as required. After that, apply the appropriate profile to a policy to enforce the feed.

   > Remote Categories fall under the FortiGuard Category Based Filter, and this feature must be enabled for them to function. It is recommended to avoid using the Allow action for remote categories, as it will not override the original action defined in the FortiGuard Category Based Filter. In contrast, actions such as Block or Monitor can override the original action specified in the FortiGuard Category Based Filter.

4. Go to *Log & Report* > *Forward Traffic* and view the logs.



To enable Fabric Feeds in the CLI:

1. On FortiGate, enable the default Fabric Feeds (also known as external resources) that correspond to the FortiAnalyzer indicators.

FortiAnalyzer indicator	FortiGate default Fabric Feed
Domain	Domain_Blocked_by_Fabric_SecOps
URL	URL_Blocked_by_Fabric_SecOps
IP	IP_Blocked_by_Fabric_SecOps

FortiAnalyzer indicator	FortiGate default Fabric Feed
MAC address	MAC_Blocked_by_Fabric_SecOps

In this example, all default Fabric Feeds are enabled.

```
config system external-resource
    edit "IP_Blocked_by_Fabric_SecOps"
        set uuid 495e0cb0-3d94-51f0-7ec7-f0ff45079b48
        set status enable
        set type address
        set update-method fortimq
    next
    edit "Domain_Blocked_by_Fabric_SecOps"
        set uuid 495e126e-3d94-51f0-755a-1240f7ba7742
        set status enable
        set type domain
        set update-method fortimq
    next
    edit "URL_Blocked_by_Fabric_SecOps"
        set uuid 495e1430-3d94-51f0-b321-be7c2ba63e2c
        set status enable
        set update-method fortimq
    next
    edit "MAC_Blocked_by_Fabric_SecOps"
        set uuid 495e15ca-3d94-51f0-ff6a-5d03040d6215
        set status enable
        set type mac-address
        set update-method fortimq
    next
```

2. After synchronizing with the FortiMQ Cloud server, the Fabric Feeds are ready to use.
3. Apply the Fabric Feeds to a policy.
4. View the logs.

   Following is an example of a log of traffic blocked by the Fabric Feed.

```
4: date=2025-07-08 time=15:37:54 eventtime=1752014274607868096 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.22 srcport=33532
srcintf="port2" srcintfrole="undefined" dstip=172.16.200.207 dstport=80 dstintf="port1"
dstintfrole="undefined" dstthreatfeed="IP_Blocked_by_Fabric_SecOps" srccountry="Reserved"
dstcountry="Reserved" sessionid=105067 proto=6 action="deny" policyid=1 policytype="policy"
poluuid="04c3e4b6-5091-51f0-7e99-3984037e433c" policyname="firewall policy" service="HTTP"
trandisp="noop" appcat="unscanned" duration=0 sentbyte=0 rcvdbyte=0 sentpkt=0 rcvdpkt=0
fctuid="F56F1D334C0B4CF599F47C9B4F013EDE" unauthuser="fosqa" unauthusersource="forticlient"
srcremote=172.18.62.231 crscore=30 craction=131072 crlevel="high"
```

Label-based sandbox exemptions for sensitive data control

Support added for sandbox submission exemptions based on AIP/MPIP labels, enabling FortiGate to respect data governance policies and prevent sensitive files from leaving the network. This provides more granular control than

exemptions based solely on file types.

```
config antivirus profile
    edit <name>
        set analytics-ignore-mpip <string>
    next
end
```

Sandbox submission exemptions can be configured based on Microsoft Purview Information Protection (MPIP/AIP) labels. These labels can come from remote connectors (Microsoft 365/Azure) or be defined as local GUID-based labels. The antivirus profile's analytics-ignore-mpip option is used to enforce these exemptions during post-transfer scanning.

Use case 1

When an Azure SDN connector syncs Microsoft 365 sensitivity labels, selecting the corresponding remote MPIP label for `analytics-ignore-mpip` lets FortiGate skip sandbox submissions for files tagged with that label while still analyzing all others.

1. Configure an MPIP external connector and a DLP label that uses the external connector's labels. See Sensitivity labels for a sample configuration.
2. Configure an antivirus profile that uses the DLP label:

```
config antivirus profile
    edit "FSA_IGNORE_MPIP_REMOTE"
        set feature-set proxy
        set analytics-ignore-mpip "remote_label_azure"
        config http
            set av-scan monitor
            set fortisandbox monitor
        end
    next
end
```

3. Download a file from an HTTP server that matches the MPIP label.
   No logs are created, as the file is not submitted to FortiSandbox.

```
# execute log display

0 logs found.
0 logs returned.
```

4. Download a file from an HTTP server that does not matches the MPIP label.
   Antivirus FortiSandbox submission logs are created:

```
# execute log display
2 logs found.
2 logs returned.

1: date=2025-10-06 time=15:39:55 eventtime=1759790394976263532 tz="-0700" logid="0201009233"
type="utm" subtype="virus" eventtype="analytics" level="information" vd="vdom1" policyid=2
```

```
poluuid="2802b718-8c5a-51f0-0eaa-af53d454e7fc" policytype="policy" msg="File submitted to
Sandbox." action="analytics" service="HTTP" sessionid=144855 srcip=10.1.100.252
dstip=172.16.200.204 srcport=63711 dstport=80 srccountry="Reserved" dstcountry="Reserved"
srcintf="port2" srcintfrole="lan" dstintf="port3" dstintfrole="lan" srcuuid="eb7a7402-4489-
51ef-4afd-ca286e50ca04" dstuuid="eb7a7402-4489-51ef-4afd-ca286e50ca04" proto=6
direction="incoming" filename="Sample_spreadsheet_personal.xlsx" filetype="msofficex"
url="http://172.16.200.204/sandbox/aip_labels/Sample_spreadsheet_personal.xlsx" profile="FSA_
IGNORE_MPIP_REMOTE" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:142.0) Gecko/20100101
Firefox/142.0" httpmethod="GET" referralurl="http://172.16.200.204/sandbox/aip_labels/"
analyticscksum="7cdcd7bb2a50c16418c15258327ee33f46c9919ee82082ed0886b19236049eed"
analyticssubmit="true"

2: date=2025-10-06 time=15:39:55 eventtime=1759790394976252778 tz="-0700" logid="0201009238"
type="utm" subtype="virus" eventtype="analytics" level="information" vd="vdom1"
srcip=10.1.100.252 dstip=172.16.200.204 srcport=63711 dstport=80 action="monitored"
service="HTTP" filename="Sample_spreadsheet_personal.xlsx" fsaverdict="clean"
analyticscksum="7cdcd7bb2a50c16418c15258327ee33f46c9919ee82082ed0886b19236049eed"
dtype="fortisandbox"
```

Use case 2

When a local MPIP label with a Globally Unique Identifier (GUID) is assigned for `analytics-ignore-mpip`, FortiGate bypasses sandbox submission for files carrying that label, keeping sensitive internal data within the network.

1. Configure a local MPIP label. See Sensitivity labels for a sample configuration.
2. Configure an antivirus profile that uses the local label:

```
config antivirus profile
    edit "FSA_IGNORE_MPIP_REMOTE"
        set feature-set proxy
        set analytics-ignore-mpip "local_label"
        config http
            set av-scan monitor
            set fortisandbox monitor
        end
    next
end
```

3. Download a file from an HTTP server that matches a GUID defined in the local label.
   No logs are created, as the file is not submitted to FortiSandbox.

```
# execute log display

0 logs found.
0 logs returned.
```

4. Download a file from an HTTP server that does not matches any of GUID defined in the local label.
   Antivirus FortiSandbox submission logs are created:

```
# execute log display
2 logs found.
```

```
2 logs returned.

1: date=2025-10-06 time=16:19:46 eventtime=1759792785952026788 tz="-0700" logid="0201009238"
type="utm" subtype="virus" eventtype="analytics" level="information" vd="vdom1"
srcip=10.1.100.252 dstip=172.16.200.204 srcport=63811 dstport=80 action="monitored"
service="HTTP" filename="Fortigate_fsa_integration_general_anyone_unrestricted.docx"
fsaverdict="clean"
analyticscksum="e0fb1e820302e59cc1b501fc7b2841706705c0c6dcee943f746679044eadb37e"
dtype="fortisandbox"

2: date=2025-10-06 time=16:19:46 eventtime=1759792785952002181 tz="-0700" logid="0201009233"
type="utm" subtype="virus" eventtype="analytics" level="information" vd="vdom1" policyid=2
poluuid="2802b718-8c5a-51f0-0eaa-af53d454e7fc" policytype="policy" msg="File submitted to
Sandbox." action="analytics" service="HTTP" sessionid=146076 srcip=10.1.100.252
dstip=172.16.200.204 srcport=63811 dstport=80 srccountry="Reserved" dstcountry="Reserved"
srcintf="port2" srcintfrole="lan" dstintf="port3" dstintfrole="lan" srcuuid="eb7a7402-4489-
51ef-4afd-ca286e50ca04" dstuuid="eb7a7402-4489-51ef-4afd-ca286e50ca04" proto=6
direction="incoming" filename="Fortigate_fsa_integration_general_anyone_unrestricted.docx"
filetype="msofficex" url="http://172.16.200.204/sandbox/aip_labels/Fortigate_fsa_integration_
general_anyone_unrestricted.docx" profile="FSA_IGNORE_MPIP_REMOTE" agent="Mozilla/5.0 (Windows
NT 10.0; Win64; x64; rv:142.0) Gecko/20100101 Firefox/142.0" httpmethod="GET"
referralurl="http://172.16.200.204/sandbox/aip_labels/"
analyticscksum="e0fb1e820302e59cc1b501fc7b2841706705c0c6dcee943f746679044eadb37e"
analyticssubmit="true"
```

Simultaneous FortiSandbox Cloud and on-premise integration

FortiGate now support simultaneous use of FortiSandbox Cloud and on-premise, enabling antivirus profiles to route files based on policy or auto-selection. This hybrid approach offers flexible sandboxing, data privacy, and compliance. Signature and URL databases are combined.

To enable FortiSandbox Cloud and on-premise integration in the GUI:

1. Go to *Security Fabric > Fabric Connectors* and double-click the *Sandbox* card.
2. In the *FortiSandbox* section, set *Status* to *Enable* and configure the settings as required. See FortiSandbox appliance.
3. In the *FortiSandbox Cloud* section, set*Status* to *Enable* and configure the settings as required. See FortiSandbox Cloud (FortiSandbox PaaS).
4. Set *Preferred FortiSandbox device* to *FortiSandbox* or *FortiSandbox Cloud*.
5. Click *OK*.

To enable FortiSandbox Cloud and on-premise integration in the CLI:

Only one can be set as the default.

```
config system fortisandbox
    edit "sandbox"
        set status enable
        set default enable
    next
    edit "sandbox-cloud"
        set status enable
        set default enable
    next
end
```

To configure an antivirus profile:

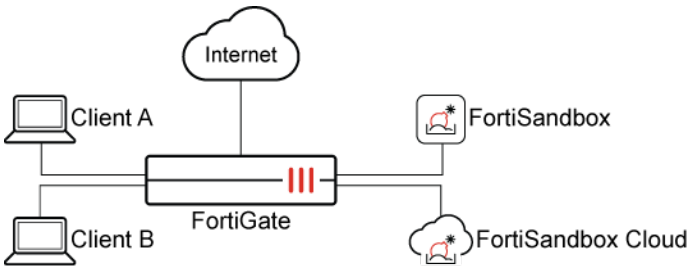
1. Go to *Security Profiles* > *AntiVirus* and click *Create New*.
2. Set *Feature set* to *Flow-based* or *Proxy-based*.
3. Enable the protocols to inspect.
4. Enable *Send files to FortiSandbox for inspection*.
5. Set *FortiSandbox destination* to *Auto*, *FortiSandbox*, or *FortiSandbox Cloud*.
6. Configure the remaining settings as required, then click *OK*.
   See Antivirus for more information.

To add the antivirus profile to a policy:

1. Go to *Policy & Objects* > *Firewall Policy*, and click *Create New* or edit an existing policy.
2. Set *Inspection mode* to *Flow-based* or *Proxy-based*.
3. In the *Security Profiles* section, enable *AntiVirus* and select the antivirus profile from the drop-down list.
4. Configure the remaining settings as needed, then click *OK*.

Use cases

- Use case 1: Default on-premise, post transfer, proxy mode on page 522
- Use case 2: Default cloud, post transfer, flow mode on page 523
- Use case 3: On-premise override, inline scan on page 524
- Use case 4: Cloud override, inline scan on page 526
- Use case 5: Split by policy, post transfer on page 528
- Use case 6: Split by policy, inline scan on page 531
- Use case 7: Combined analytics, broader coverage on page 533



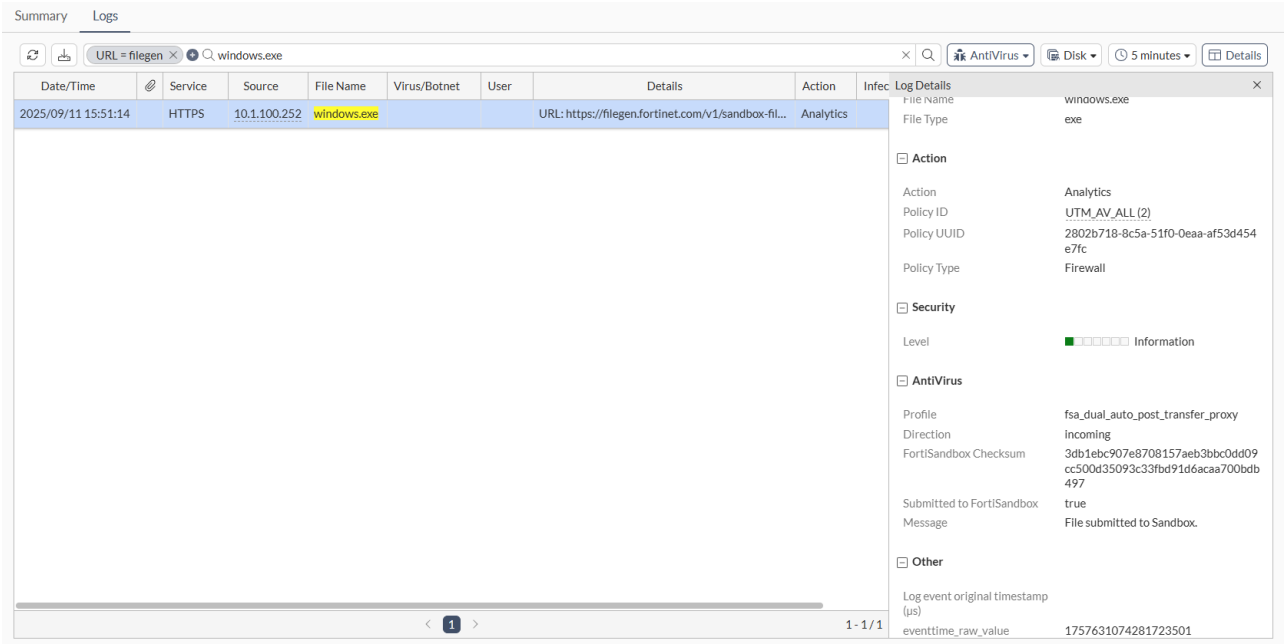
Use case 1: Default on-premise, post transfer, proxy mode

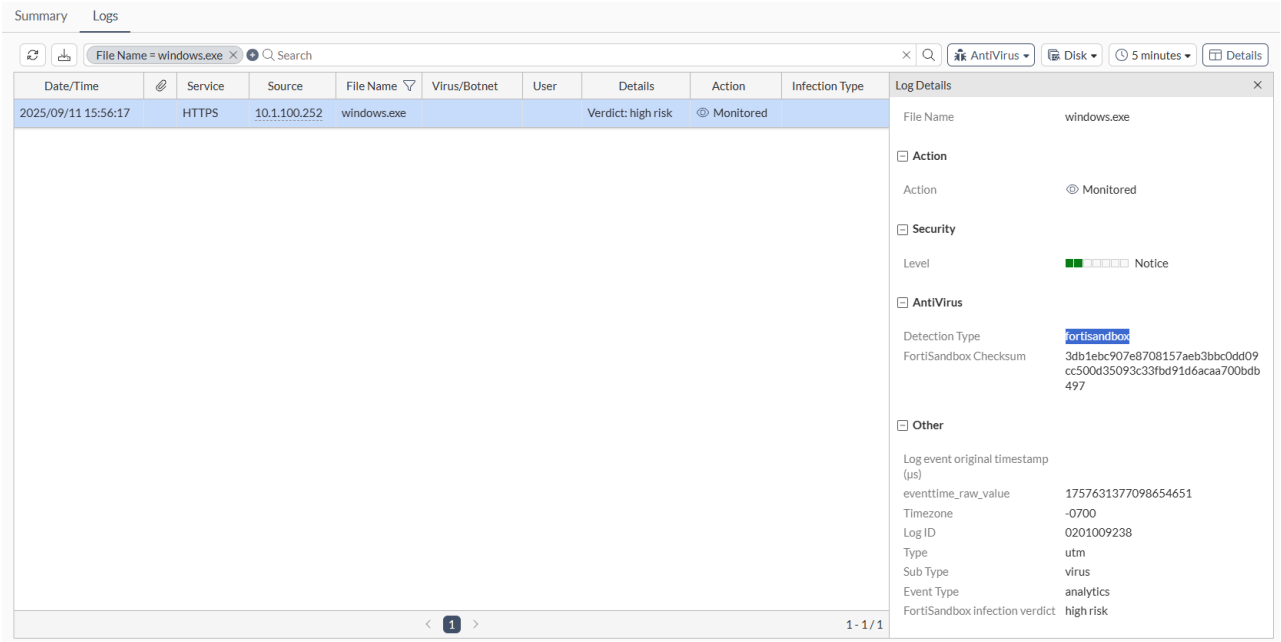
In this use case, the system is set to default to an on-premise FortiSandbox appliance. All file submissions are routed to the on-premise FortiSandbox for post transfer analysis.

Configure the following:

Option	Settings
Sandbox Settings	
FortiSandbox	• Inline scan: enabled
FortiSandbox Cloud	• Inline scan: enabled
Default FortiSandbox Device	• Preferred FortiSandbox device: FortiSandbox
Antivirus profile: fsa_dual_auto_post_transfer_proxy	• Feature set: Proxy-based • Scan strategy: Post Transfer • FortiSandbox destination: Auto
Policy	• Inspection mode: Proxy-based • Security Profiles: fsa_dual_auto_post_transfer_proxy

Download a file from https://www.fortiguard.com/sample-files and check the logs on FortiGate to see the analytics submission log showing that the antivirus profile was applied, and the verdict returned from the FortiSandbox showing that the detection type was FortiSandbox.





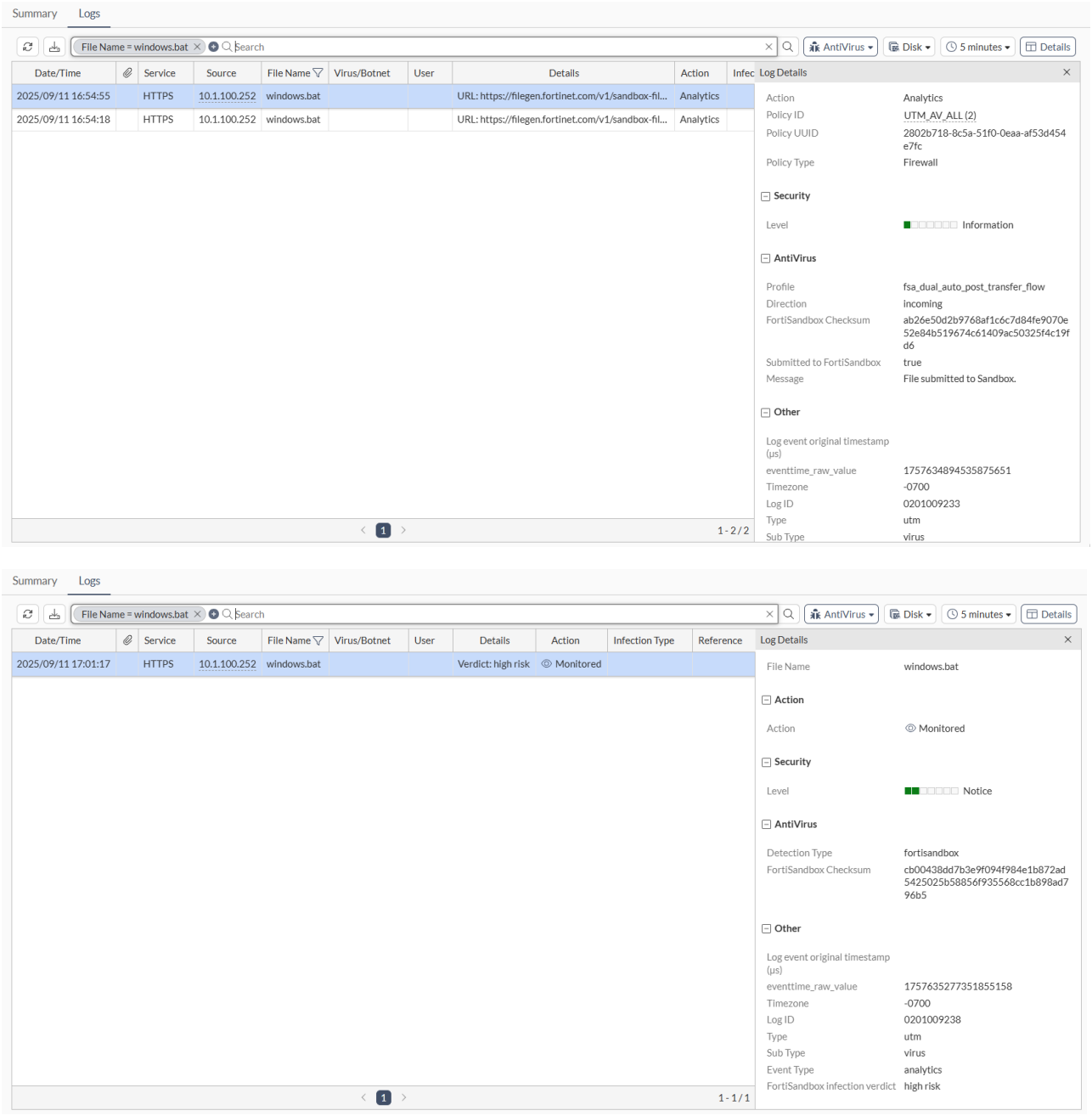
Use case 2: Default cloud, post transfer, flow mode

In this use case, the system is set to default to FortiSandbox Cloud. All file submissions are routed to FortiSandbox Cloud for post transfer analysis.

Configure the following:

Option	Settings
Sandbox Settings	
FortiSandbox	• Inline scan: enabled
FortiSandbox Cloud	• Inline scan: enabled
Default FortiSandbox Device	• Preferred FortiSandbox device: FortiSandbox Cloud
Antivirus profile: fsa_dual_auto_post_transfer_flow	• Feature set: Flow-based • Scan strategy: Post Transfer • FortiSandbox destination: Auto
Policy	• Inspection mode: Flow-based • Security Profiles: fsa_dual_auto_post_transfer_flow

Download a file from https://www.fortiguard.com/sample-files and check the logs on FortiGate to see the analytics submission log showing that the antivirus profile was applied, and the verdict returned from FortiSandbox Cloud showing that the detection type was FortiSandbox.



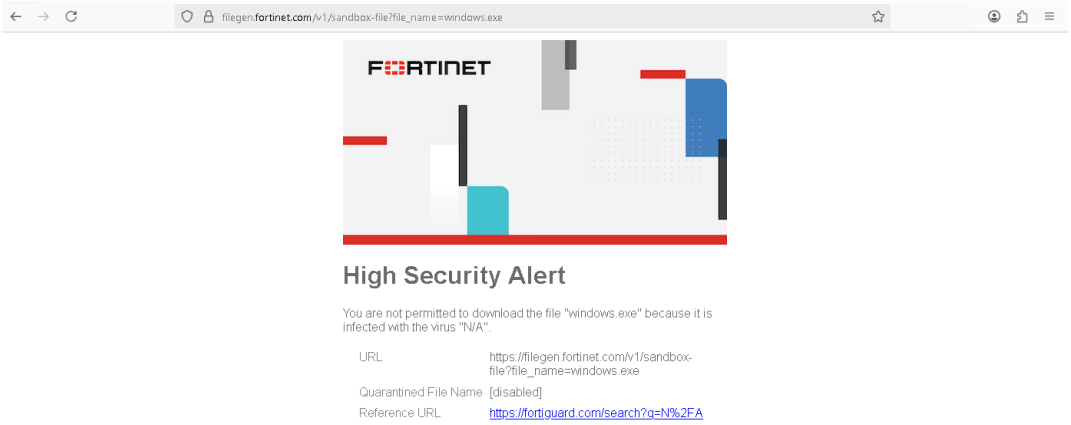
Use case 3: On-premise override, inline scan

In this use case, FortiSandbox cloud is set as the default, but the user explicitly assigns an antivirus profile that forces traffic to the on-premise FortiSandbox. All file submissions are routed to the on-premise FortiSandbox for real-time analysis.

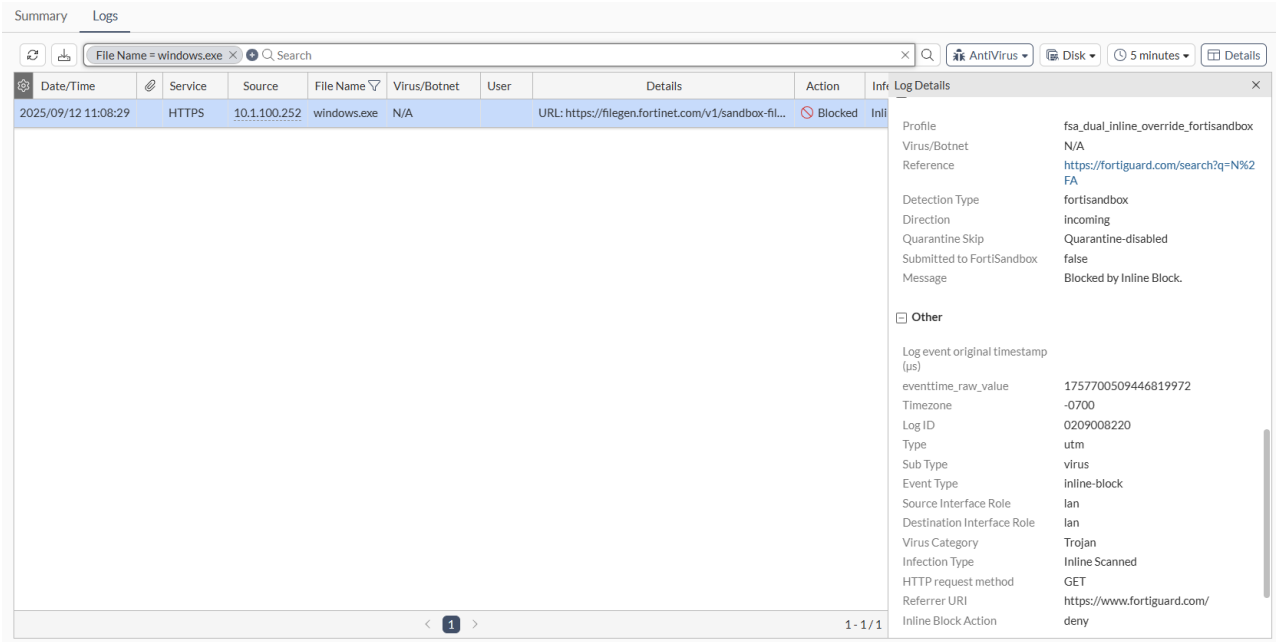
Configure the following:

Option	Settings
Sandbox Settings	
FortiSandbox	• Inline scan: enabled
FortiSandbox Cloud	• Inline scan: enabled
Default FortiSandbox Device	• Preferred FortiSandbox device: FortiSandbox Cloud
Antivirus profile: fsa_dual_inline_override_fortisandbox	• Feature set: Proxy-based • Scan strategy: Inline • FortiSandbox destination: FortiSandbox
Policy	• Inspection mode: Proxy-based • Security Profiles: fsa_dual_inline_override_fortisandbox

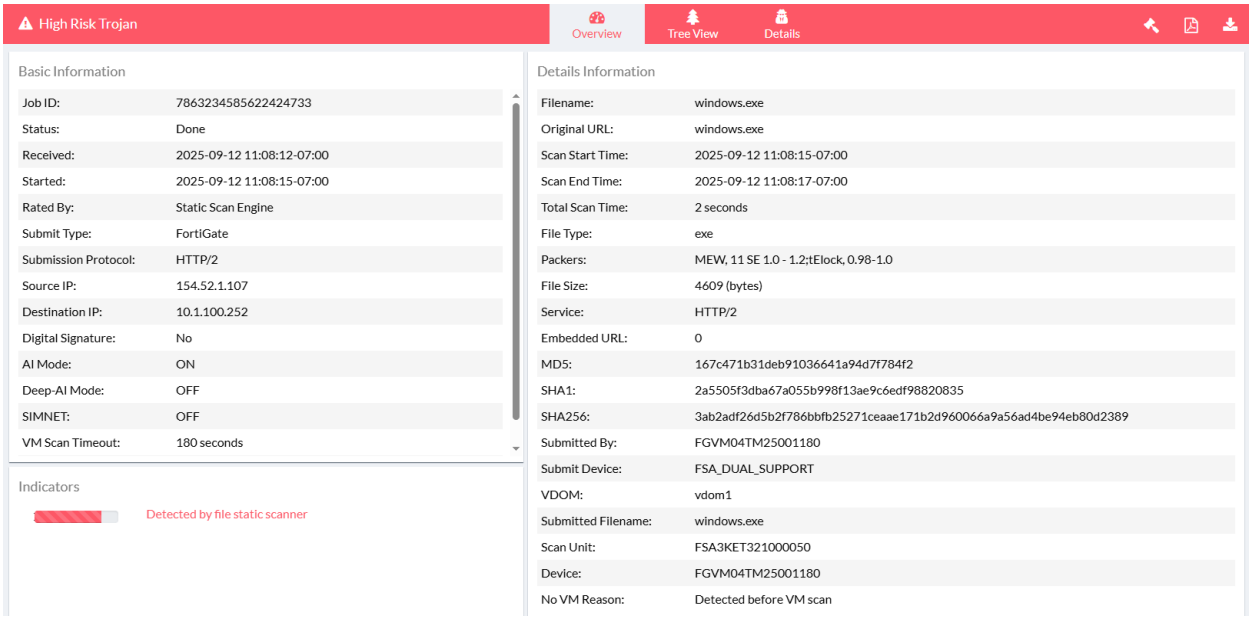
Attempt to download a file from https://www.fortiguard.com/sample-files. The file should be blocked.



Check the log on the FortiGate to see that the file was blocked.



Check the on-premise FortiSandbox to confirm the override selection.



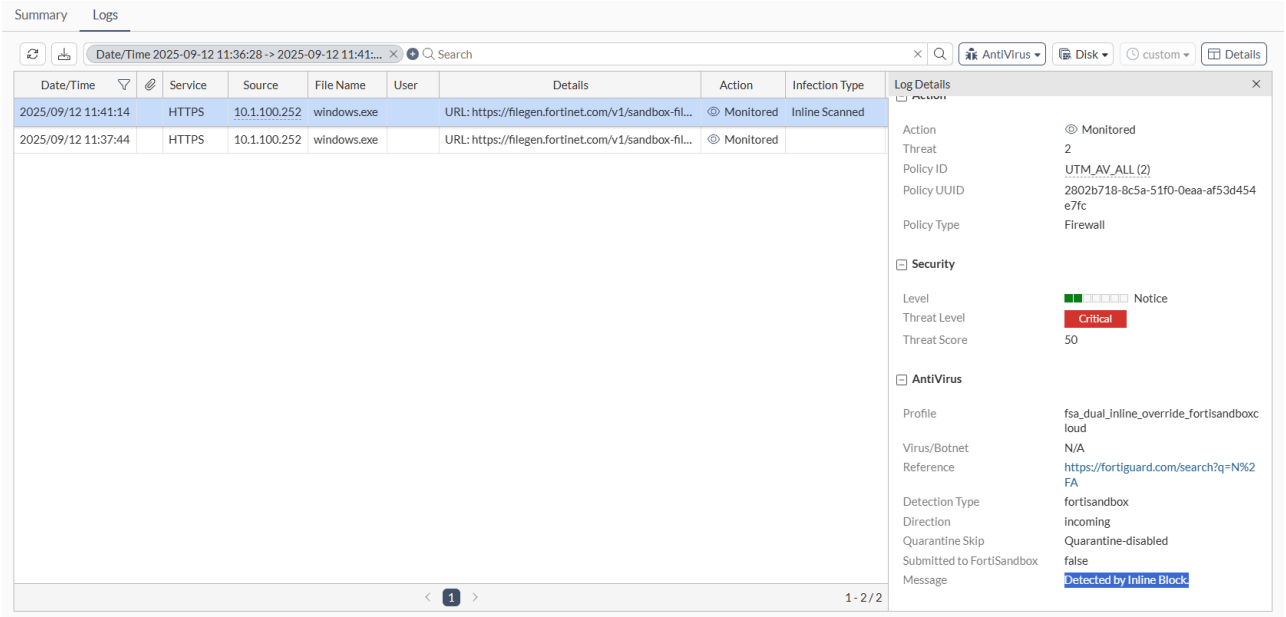
Use case 4: Cloud override, inline scan

In this use case, on-premise FortiSandbox is set as the default, but the user explicitly assigns an antivirus profile that forces traffic to the FortiSandbox cloud. All file submissions are routed to FortiSandbox Cloud for real-time analysis.

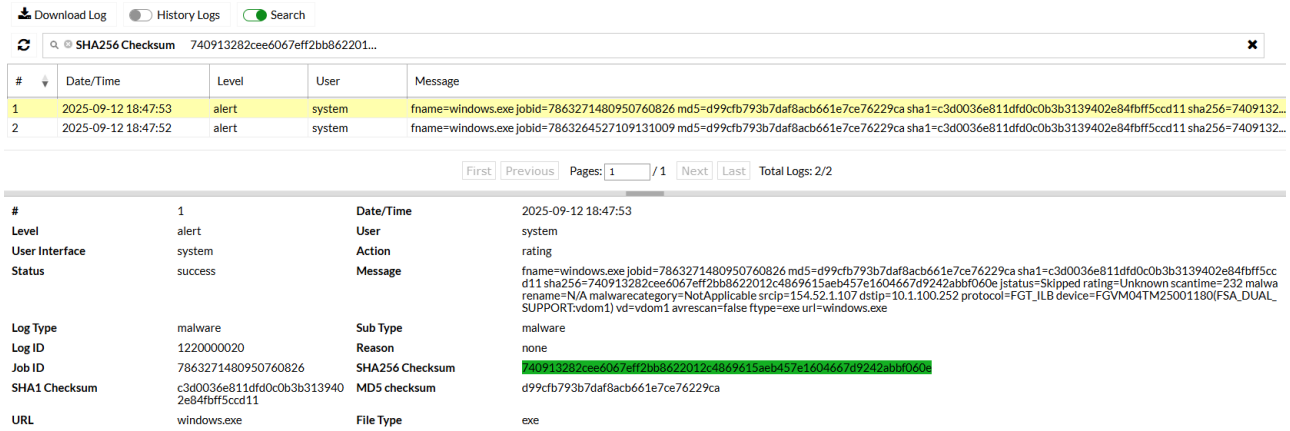
Configure the following:

Option	Settings
Sandbox Settings	
FortiSandbox	• Inline scan: enabled
FortiSandbox Cloud	• Inline scan: enabled
Default FortiSandbox Device	• Preferred FortiSandbox device: FortiSandbox
Antivirus profile: fsa_dual_inline_override_fortisandboxcloud	• Feature set: Proxy-based • Scan strategy: Inline • FortiSandbox destination: FortiSandbox Cloud
Policy	• Inspection mode: Proxy-based • Security Profiles: fsa_dual_inline_override_fortisandboxcloud

Download a file from https://www.fortiguard.com/sample-files and check the logs on FortiGate to see that the file was detected.



Check the FortiSandbox Cloud *Job Events* to confirm.



Use case 5: Split by policy, post transfer

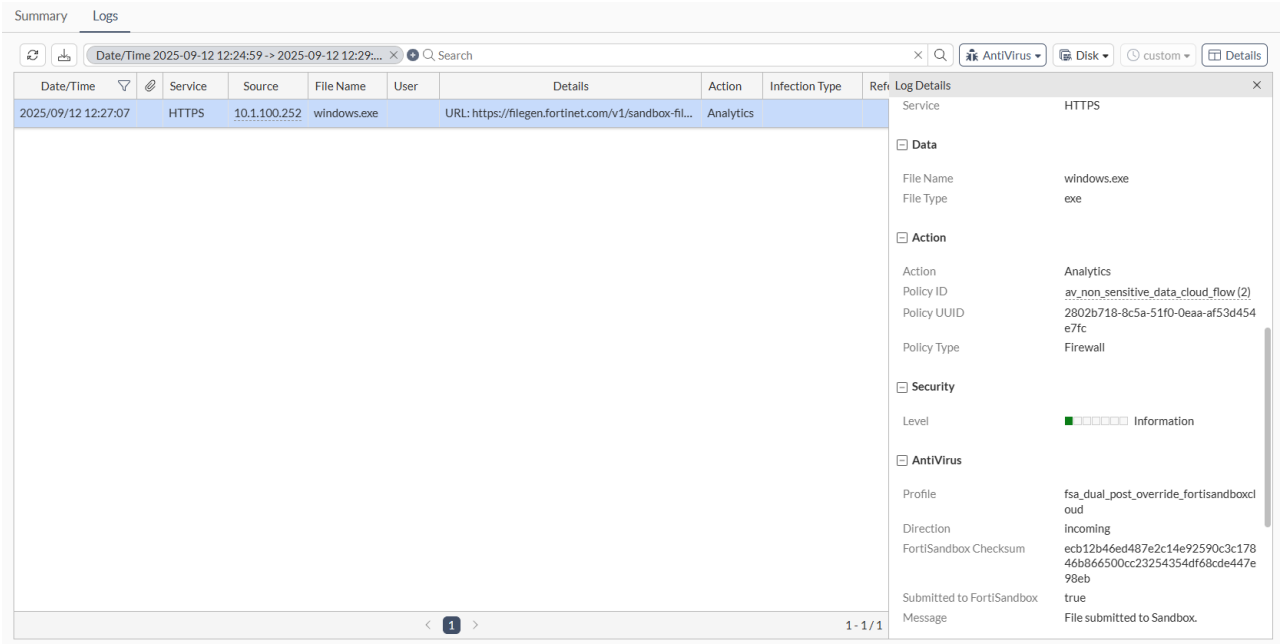
In this use case, the system is set to default to an on-premise FortiSandbox appliance. Two firewall policies are configured with different antivirus profiles, one targeting the on-premise FortiSandbox (proxy-based) and the other targeting the FortiSandbox cloud (flow-based).

Routine traffic is routed to FortiSandbox Cloud and sensitive traffic is directed to the on-premise FortiSandbox for post transfer analysis.

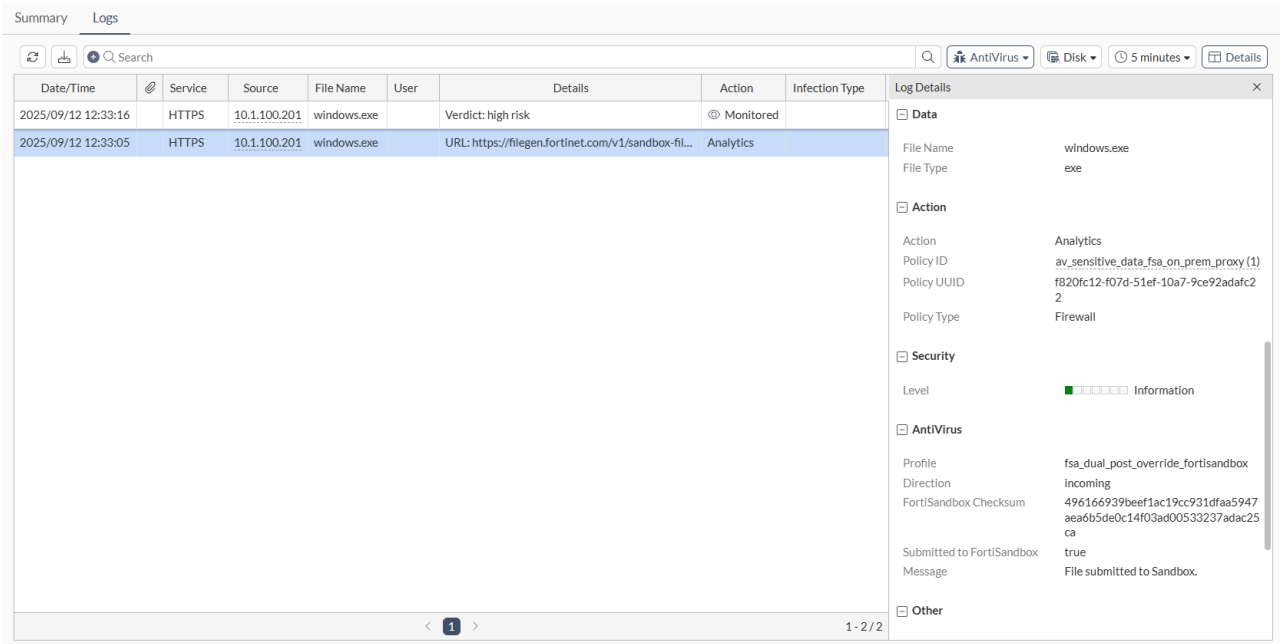
Configure the following:

Option	Settings
Sandbox Settings	
FortiSandbox	• Inline scan: disabled
FortiSandbox Cloud	• Inline scan: disabled
Default FortiSandbox Device	• Preferred FortiSandbox device: FortiSandbox
Antivirus profile 1: fsa_dual_post_override_fortisandbox	• Feature set: Proxy-based • Scan strategy: Post Transfer • FortiSandbox destination: FortiSandbox
Antivirus profile 2: fsa_dual_post_override_fortisandboxcloud	• Feature set: Flow-based • Scan strategy: Post Transfer • FortiSandbox destination: FortiSandbox Cloud
Policy 1	• Inspection mode: Proxy-based • Security Profiles: fsa_dual_post_override_fortisandbox
Policy 1	• Inspection mode: Flow-based • Security Profiles: fsa_dual_post_override_fortisandboxcloud

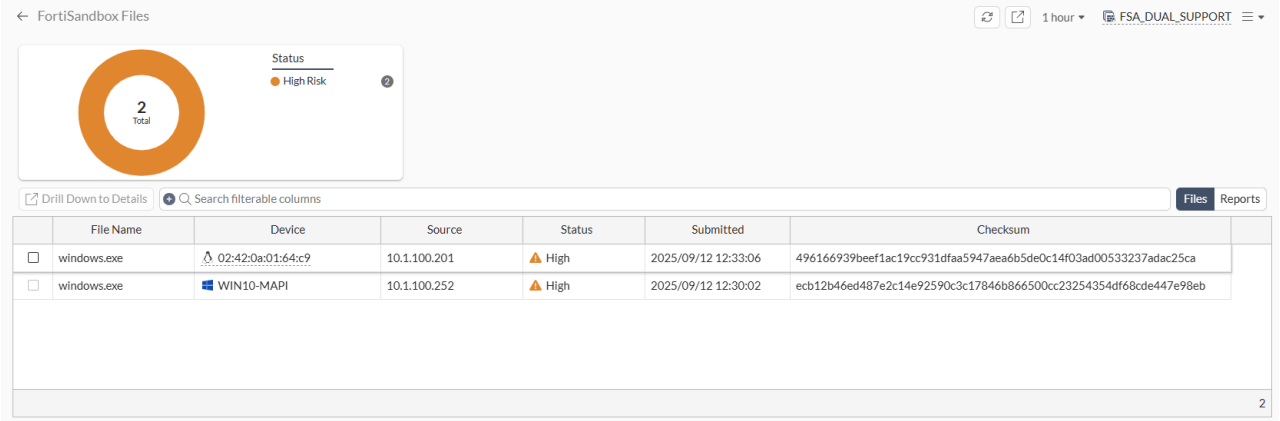
Download a file from a client that is routed to the non-sensitive policy and check the logs on FortiGate.



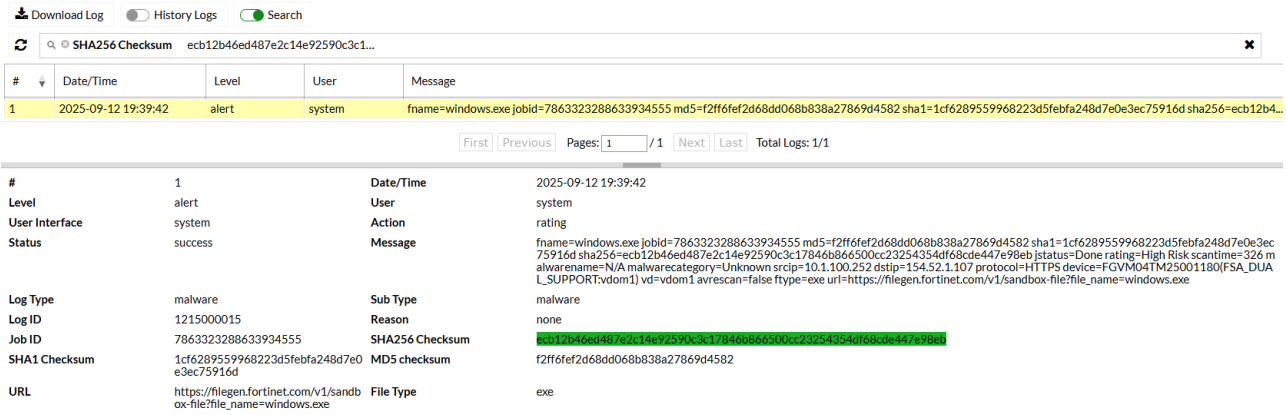
Download a file from a client that is routed to the sensitive policy and check the logs on FortiGate.



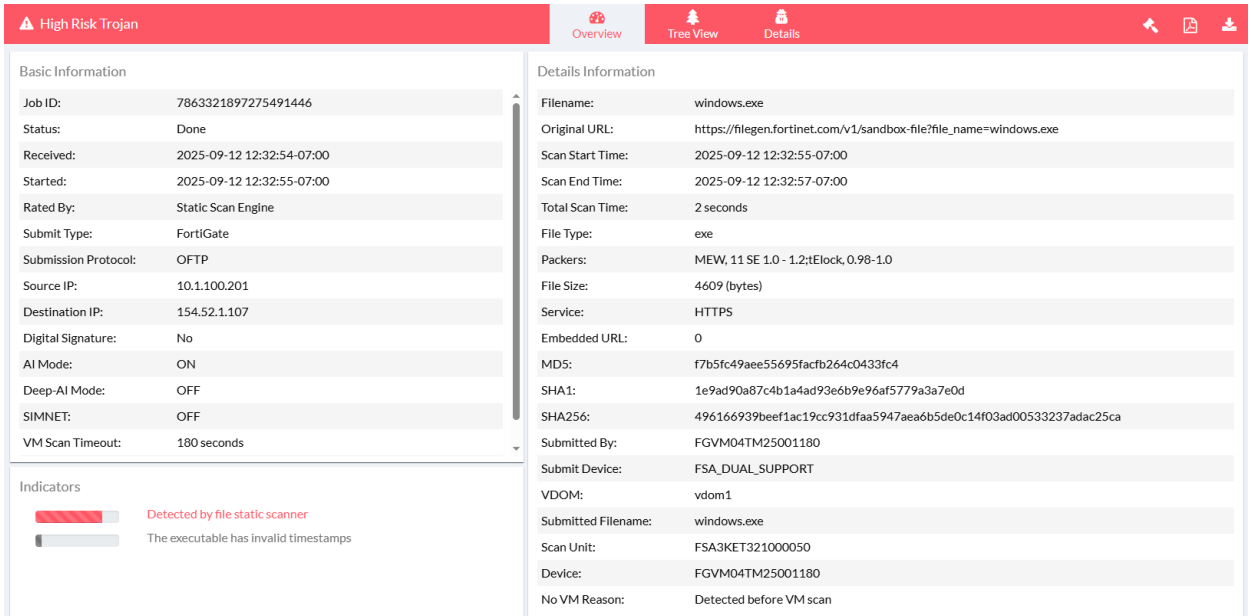
Add the *FortiSandbox Files* widget to a dashboard and then expand it to see that both submissions have verdicts returned from separate FortiSandboxes.



Check the FortiSandbox Cloud *Job Events* for the non-sensitive file log.



Check the on-premise FortiSandbox for the sensitive file log.



Use case 6: Split by policy, inline scan

In this use case, the system is set to default to an on-premise FortiSandbox appliance. Two firewall policies are configured with different antivirus profiles, one using the on-premise FortiSandbox and the other using FortiSandbox cloud.

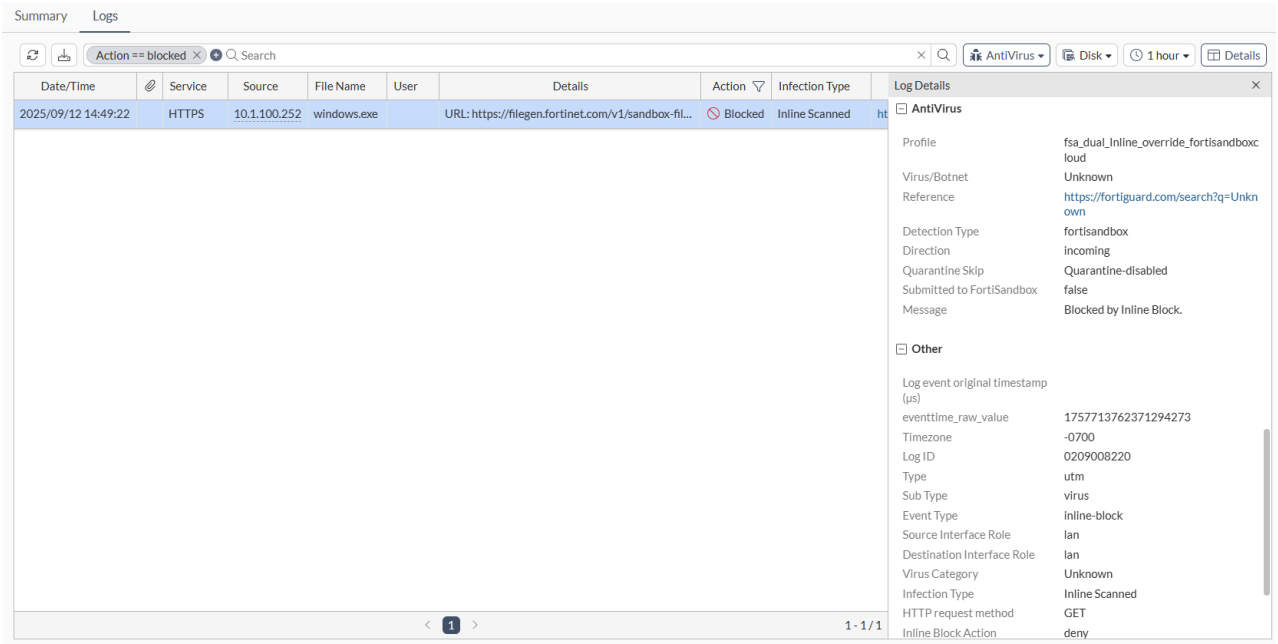
File submissions are inspected in real-time based on the antivirus profile assigned to each policy, allowing simultaneous inline use of both the on-premise FortiSandbox and FortiSandbox Cloud.

Configure the following:

Option	Settings
Sandbox Settings	
FortiSandbox	• Inline scan: enabled
FortiSandbox Cloud	• Inline scan: enabled
Default FortiSandbox Device	• Preferred FortiSandbox device: FortiSandbox
Antivirus profile 1: fsa_dual_inline_auto_selection	• Feature set: Proxy-based • Scan strategy: Inline • FortiSandbox destination: Auto
Antivirus profile 2: fsa_dual_inline_override_ fortisandboxcloud	• Feature set: Proxy-based • Scan strategy: Inline • FortiSandbox destination: FortiSandbox Cloud
Policy 1	• Inspection mode: Proxy-based • Security Profiles: fsa_dual_inline_auto_selection
Policy 2	• Inspection mode: Proxy-based • Security Profiles: fsa_dual_inline_override_fortisandboxcloud

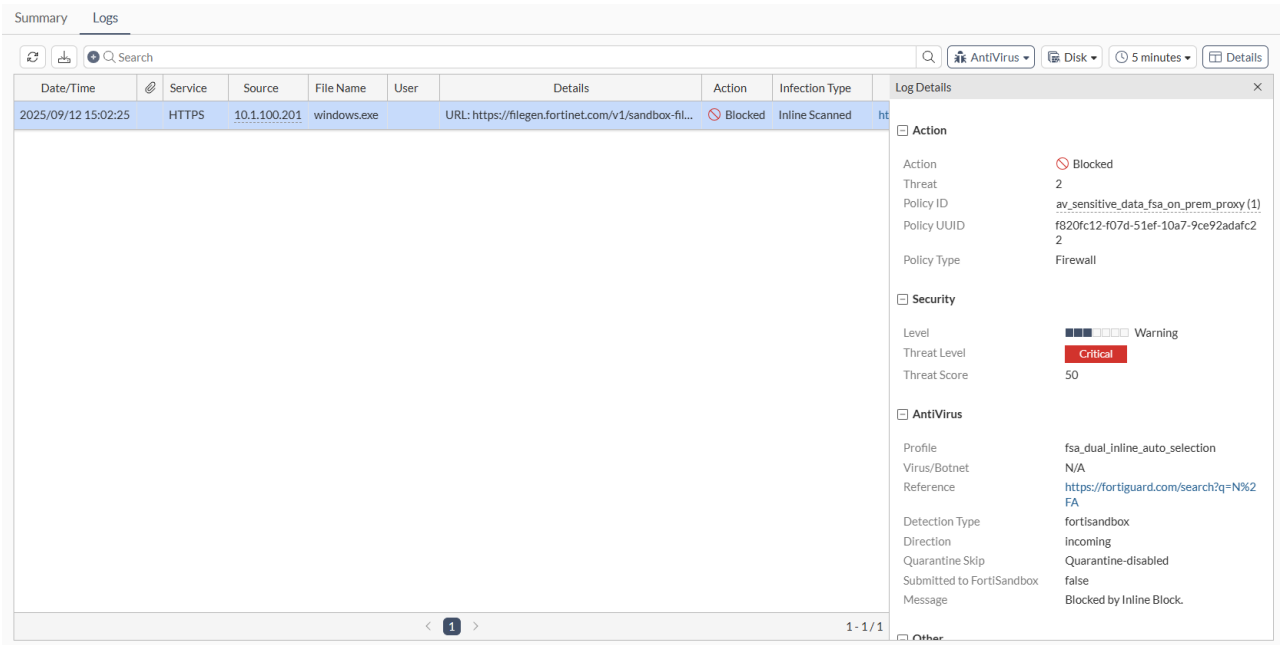
Attempt to download a file from a client that is routed to the non-sensitive policy and submits files to FortiSandbox Cloud. The file should be blocked.

Check the log on the FortiGate to see that the file was blocked.

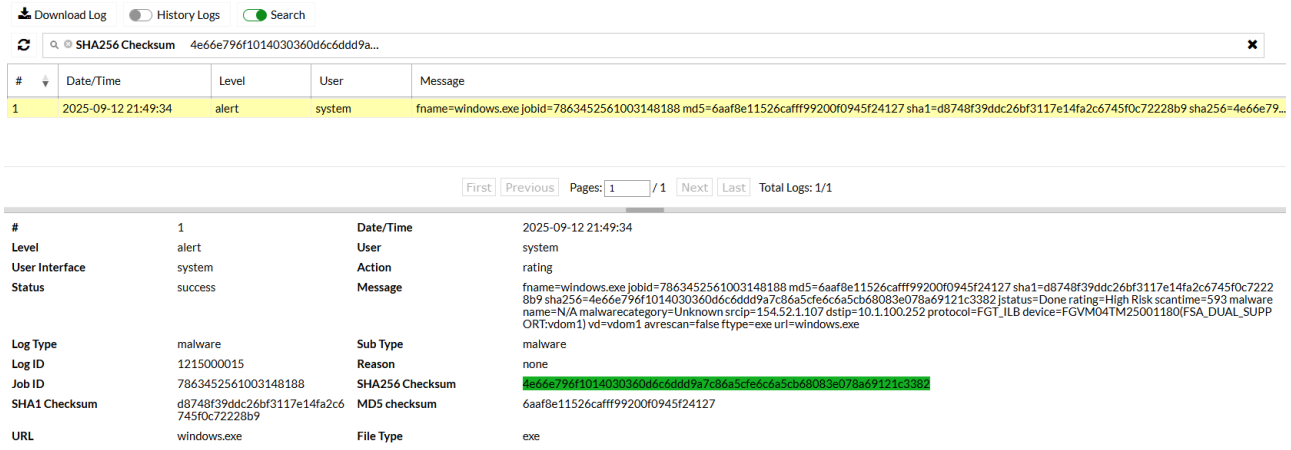


Attempt to download a file from a client that is routed to the sensitive policy and submits files to the on-premise FortiSandbox. The file should be blocked.

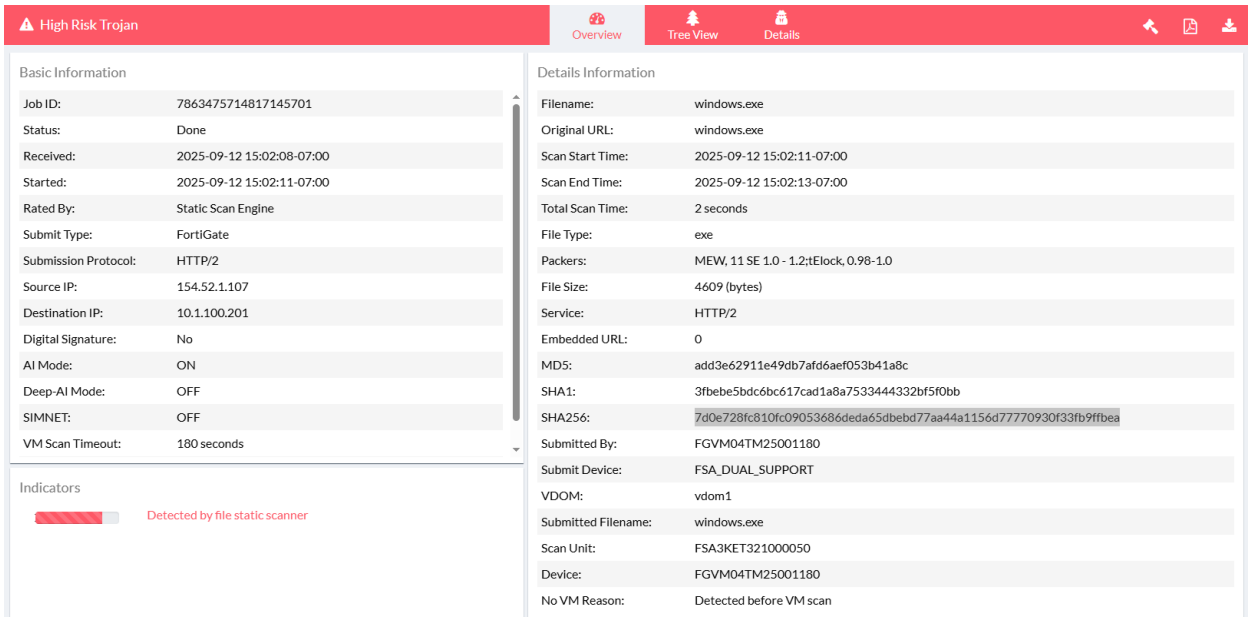
Check the log on the FortiGate to see that the file was blocked.



Check the FortiSandbox Cloud *Job Events* for the non-sensitive file log.



Check the on-premise FortiSandbox for the sensitive file log.



Use case 7: Combined analytics, broader coverage

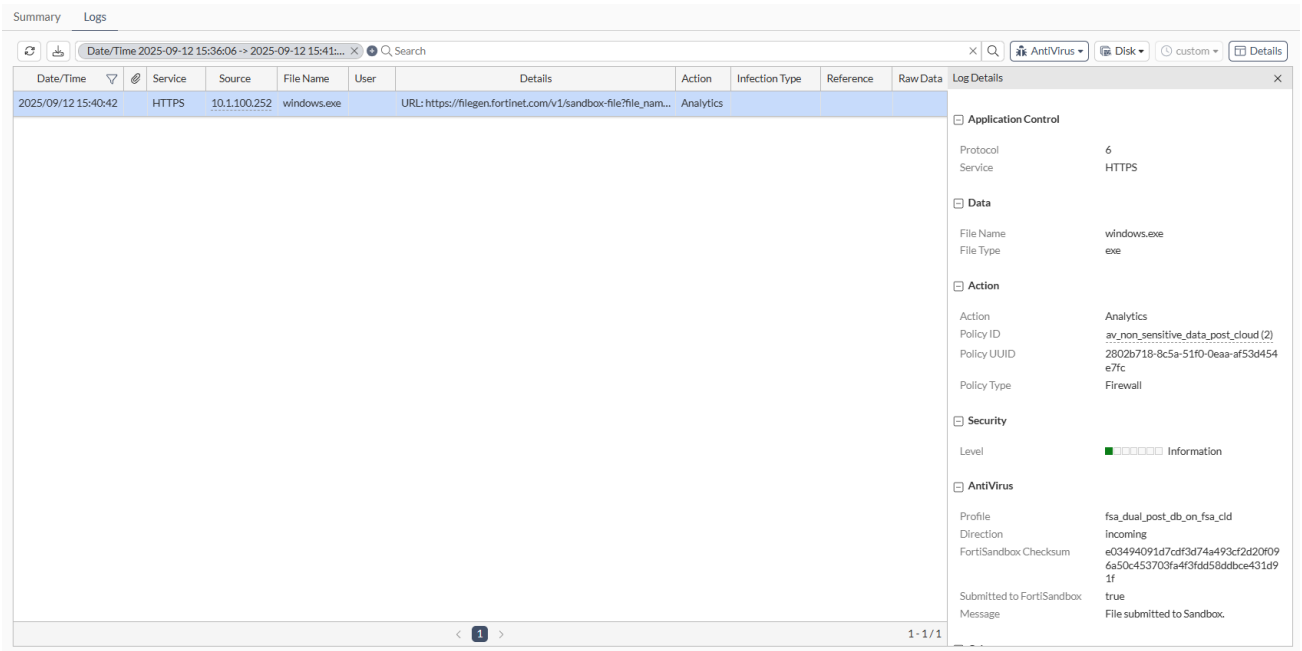
In this use case, dual FortiSandbox appliances provides access to merged signature and URL databases across both the on-premise FortiSandbox and FortiSandbox Cloud. Any antivirus profile applied to a firewall policy utilizes data from both sources, resulting in broader detection coverage.

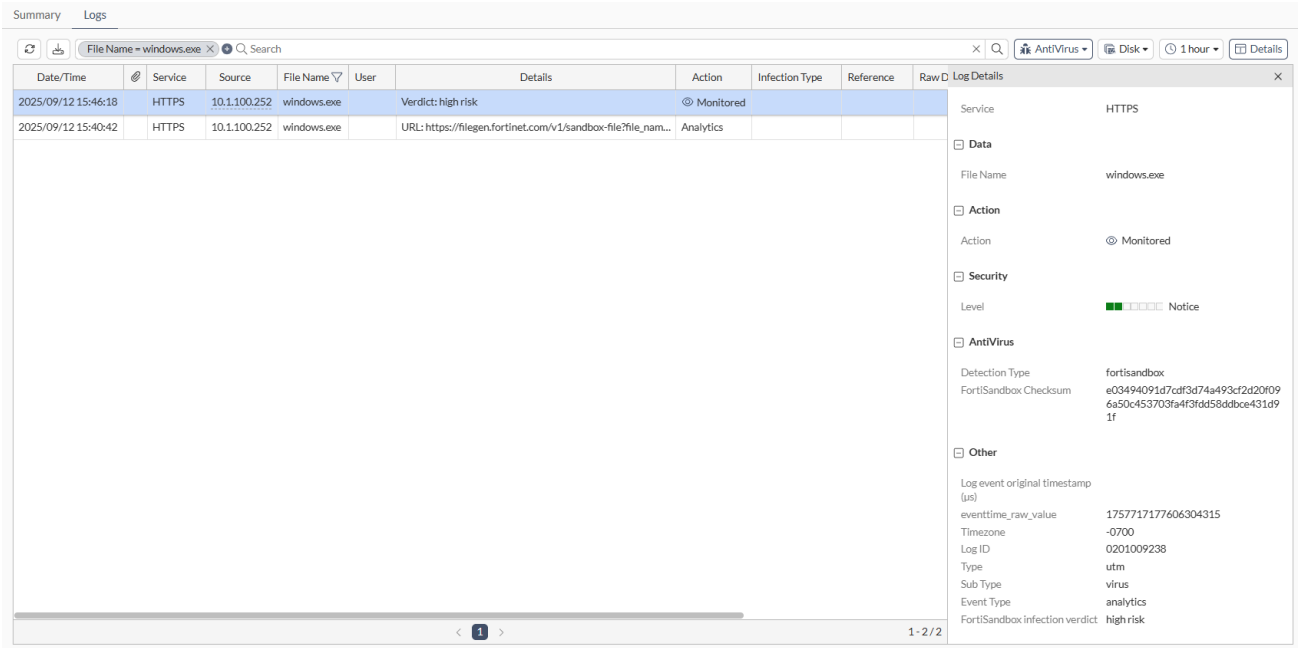
Configure the following:

Option	Settings
Sandbox Settings	
FortiSandbox	• Inline scan: enabled
FortiSandbox Cloud	• Inline scan: enabled

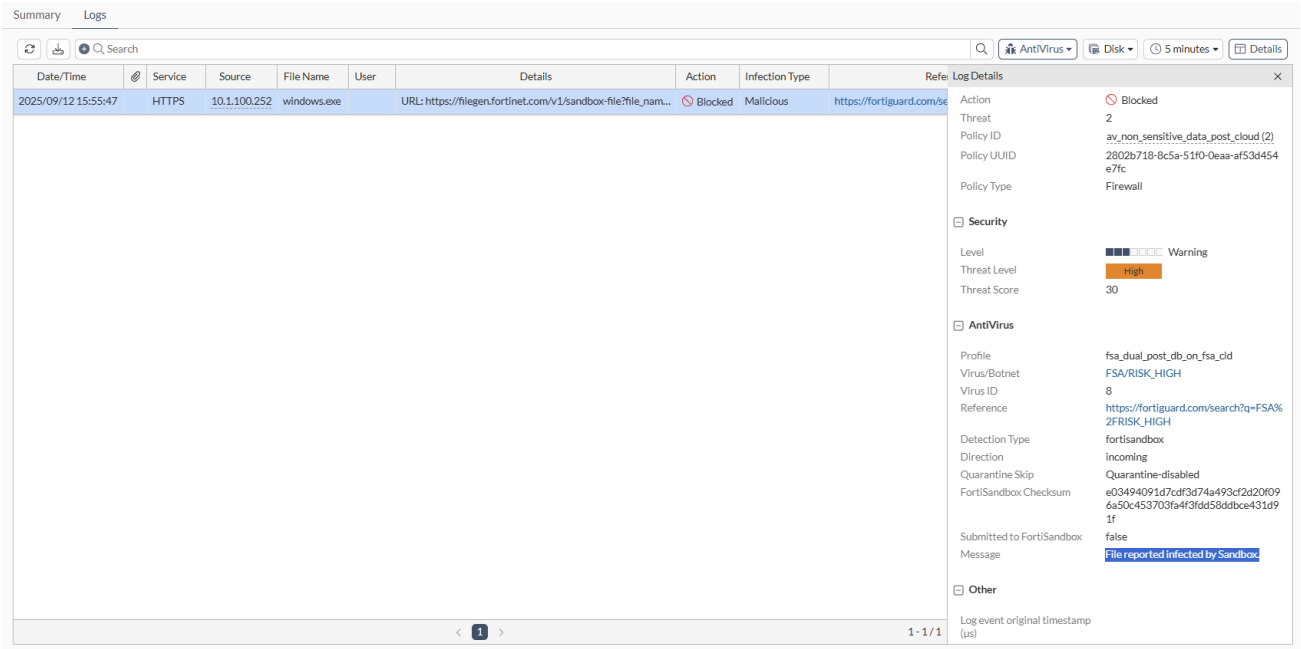
Option	Settings
Default FortiSandbox Device	• Preferred FortiSandbox device: FortiSandbox
Antivirus profile 1: fsa_dual_post_db_on_fsa_cld	• Feature set: Proxy-based • Scan strategy: Post Transfer • FortiSandbox destination: FortiSandbox Cloud
Antivirus profile 2: fsa_dual_post_db_on_auto	• Feature set: Proxy-based • Scan strategy: Post Transfer • FortiSandbox destination: Auto
Policy 1	• Inspection mode: Proxy-based • Security Profiles: fsa_dual_post_db_on_fsa_cld
Policy 2	• Inspection mode: Proxy-based • Security Profiles: fsa_dual_post_db_on_auto

Download a file from https://www.fortiguard.com/sample-files and check the logs on FortiGate to see the analytics submission and verdict logs.

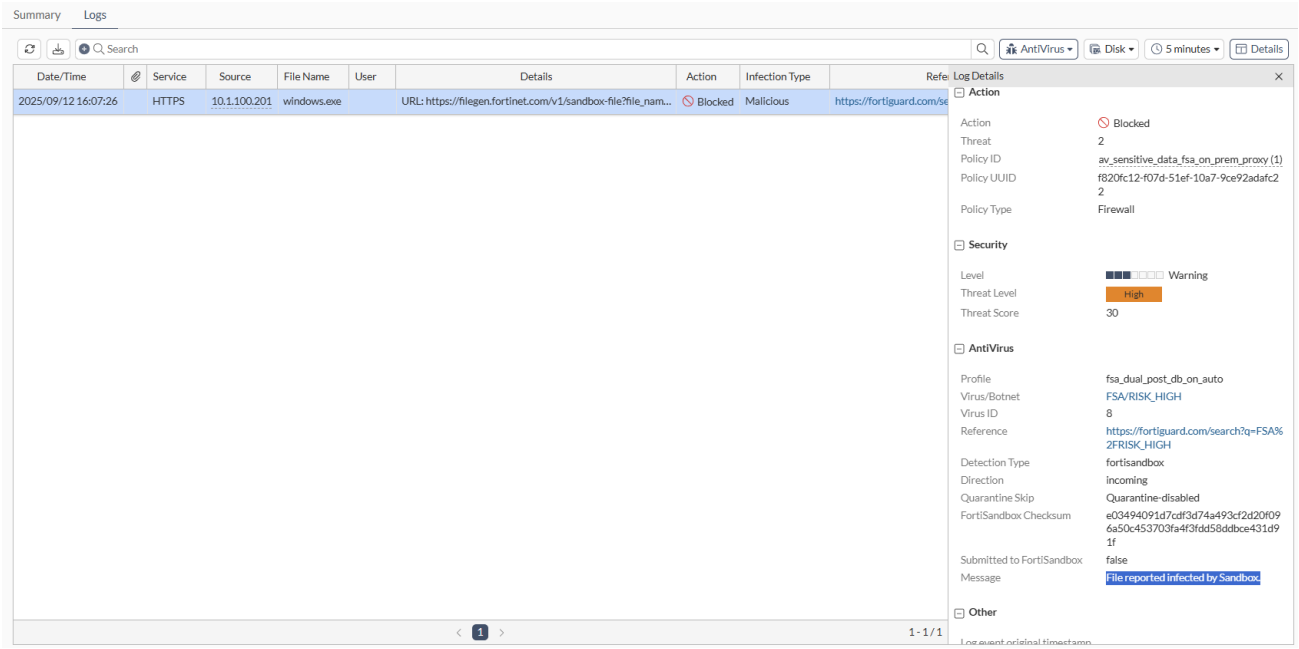




Download the same file again using the same client. The file should be blocked. Check the log on the FortiGate to see that the file was blocked.



Download the same file again using the other policy that is pointing to the other FortiSandbox. The file should be blocked. Check the log on the FortiGate to see that the file was blocked.



External SDN connectors

This section includes information about external SDN connector related new features:

- Support for Cisco ACI External EPG Subnets in Direct Connector on page 536
- Negation support in ACI direct connector address filters on page 542

Support for Cisco ACI External EPG Subnets in Direct Connector

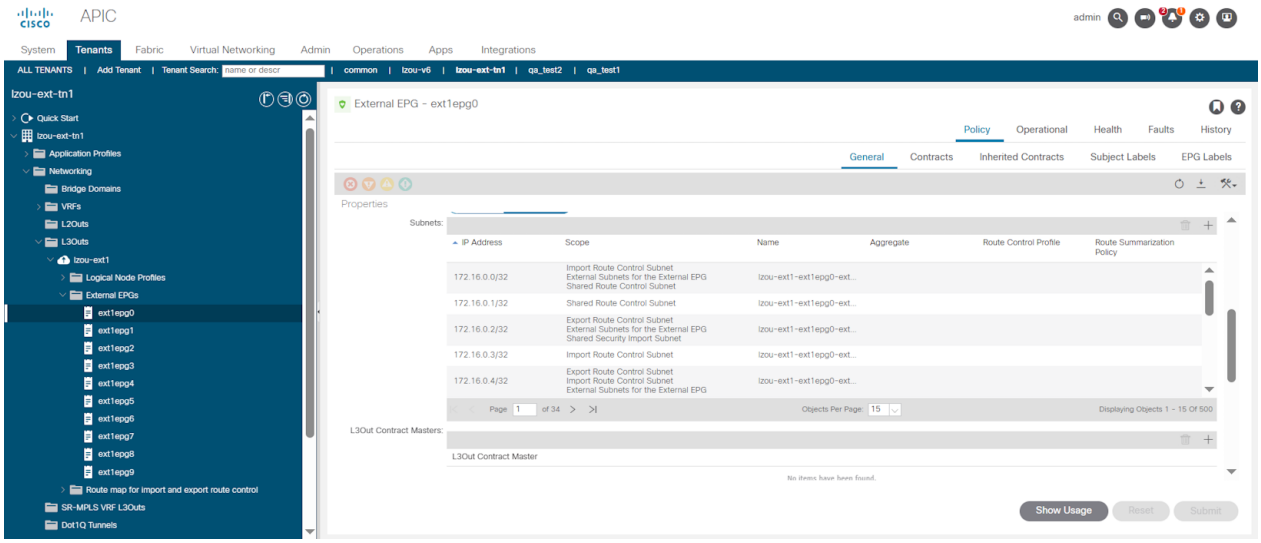
Cisco ACI external EPG (l3extSubnet) subnets are now supported by the FortiOS Direct Connector. This allow external EPG subnets to be automatically imported for use security policies, improving integration and policy accuracy.

The available firewall address filters now include <L3OutName>, <L3OutEpg>, <L3OutSubnet>, <L3OutScope>, and <L3OutEpgTag>:

```
config firewall address
    edit <address>
        set filter <key1=value1>    [& <key2=value2>] [| <key3=value3>]
    next
end
```

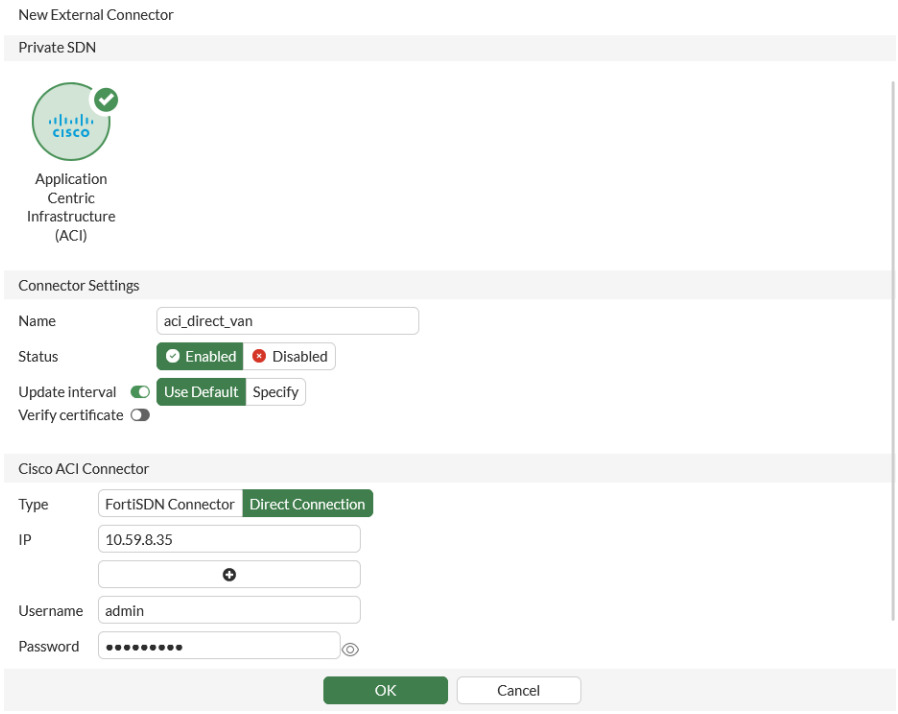
Example

Configure a CISCO ACI tenant with ACI external EPG subnets.

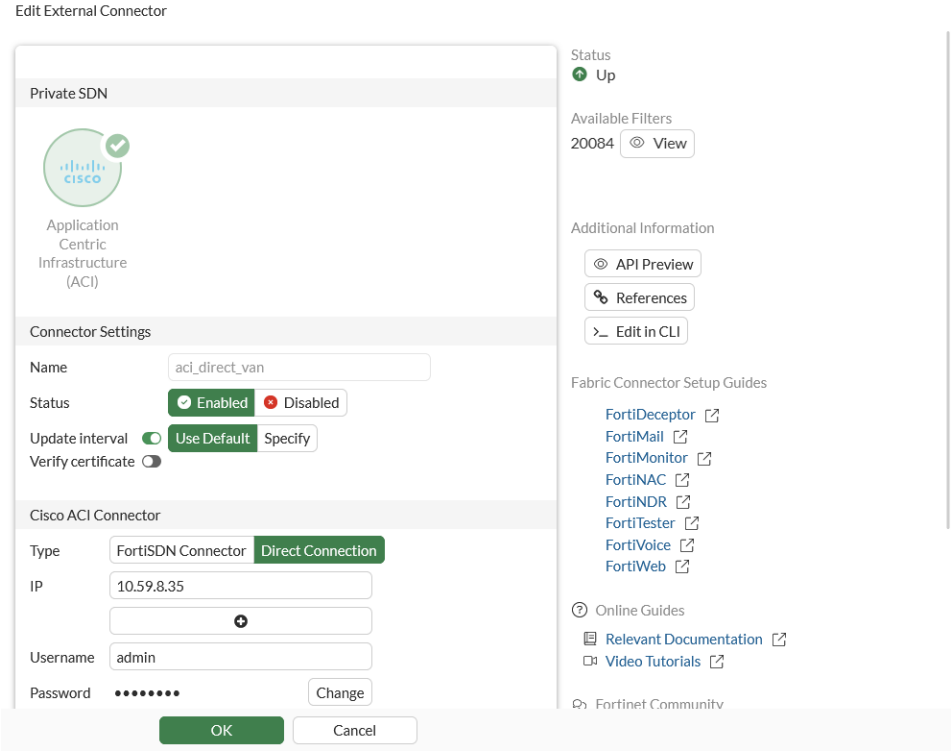


To configure an ACI SDN connector in the GUI:

1. On the FortiGate, go to *Security Fabric* > *External Connectors* and click *Create New*.
2. In the *Private SDN* section, click *Application Centric Infrastructure (ACI)*.
3. Enter a unique name for the connector.
4. Set *Type* to *Direct Connection*.



5. Configure the remaining settings as needed, then click *OK*.
6. Edit the connector to make sure that the ACI SDN connector status is up:



To configure an ACI SDN connector in the CLI:

```
config system sdn-connector
    edit "aci_direct_van"
        set type aci-direct
        set verify-certificate disable
        set server-list "10.59.8.35"
        set username "admin"
        set password xxxxxx
    next
end
```

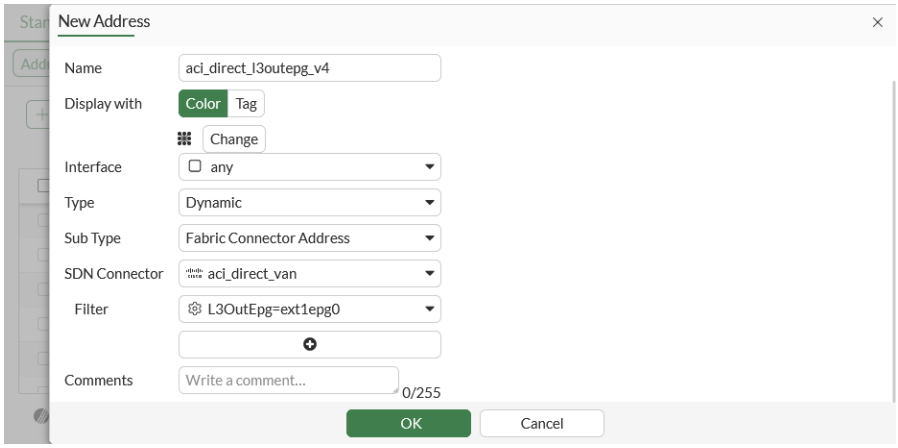
Make sure that the ACI SDN connector status is up:

```
# diagnose sys sdn status aci_direct_van
SDN Connector                       Type        Status
------------------------------------------------------------
aci_direct_van                      aci-direct  Up
```

To create a IPv4 dynamic firewall address associated with the ACI SDN connector in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*.
3. Enter a unique name for the address.

4. Set *Type* to *Dynamic*.
5. Set *Sub Type* to *Fabric Connector Address*.
6. Set *SDN Connector* to the ACI connector.
7. Set the *Filter* selection to filter out needed addresses.



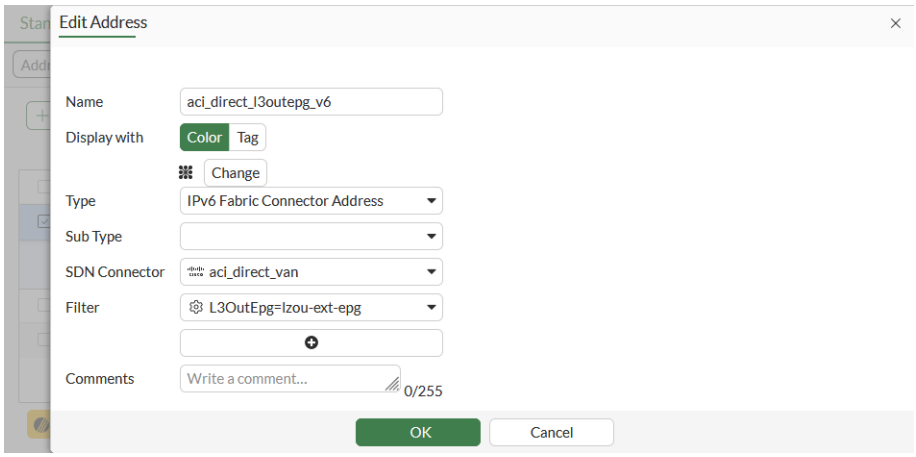
8. Configure the remaining settings as needed, then click *OK*.

To create an IPv4 dynamic firewall address associated with the ACI SDN connector in the CLI:

```
config firewall address
    edit "aci_direct_l3outepg_v4"
        set type dynamic
        set sdn "aci_direct_van"
        set filter "L3OutEpg=ext1epg0"
    next
end
```

To create a IPv6 dynamic firewall address associated with the ACI SDN connector in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *IPv6 Address*.
2. Click *Create new*.
3. Enter a unique name for the address.
4. Set *Type* to *IPv6 Fabric Connector Address*.
5. Set *SDN Connector* to the ACI connector.
6. Set the *Filter* selection to filter out needed addresses.



7. Configure the remaining settings as needed, then click *OK*.

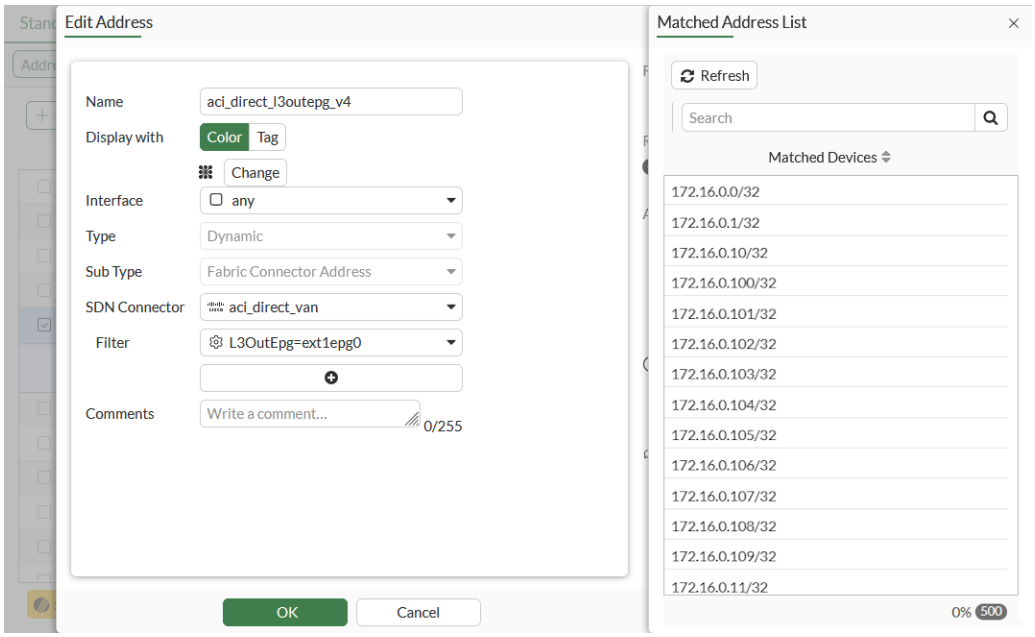
To create an IPv6 dynamic firewall address associated with the ACI SDN connector in the CLI:

```
config firewall address6
    edit "aci_direct_l3outepg_v6"
        set type dynamic
        set sdn "aci_direct_van"
        set filter "L3OutEpg=lzou-ext-epg"
    next
end
```

To check the resolved addresses in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address* or *IPv6 Address*.
2. Hover over an address name and click *View Matches Addresses* in the pop-up, or edit the address and click *View Matches Addresses* in the right pane.

   The resolved dynamic addresses are shown in the *Matched Address List*.



To check the resolved addresses in the CLI:

```
# show firewall address aci_direct_l3outepg_v4
config firewall address
    edit "aci_direct_l3outepg_v4"
        ...
        config list
            ...
            edit "172.16.0.1/32"
            next
            edit "172.16.0.10/32"
            ...
        end
    next
end
```

```
# show firewall address6 aci_direct_l3outepg_v6
config firewall address6
    edit "aci_direct_l3outepg_v6"
        ...
        config list
            edit "2033::/64"
            next
            edit "8000::/1"
            next
            edit "::/0"
            next
            edit "::/1"
            next
            edit "::/64"
            next
```

```
        end
    next
end
```

Negation support in ACI direct connector address filters

Support is added for negation in Cisco ACI direct connector address object filters, including new L3OutSubnetDescription and L3OutSubnetNameAlias filters with full IPv4/IPv6 compatibility. This simplifies filtering by allowing administrators to exclude specific subnets instead of enumerating all included ones, improving usability in environments with many networks and few exceptions.

To test the filters:

1. Configure a CISCO ACI tenant with ACI external EPG subnets.
2. On the FortiGate, configure ACI SDN connector.

```
config system sdn-connector
    edit "aci_direct_van"
        set type aci-direct
        set verify-certificate disable
        set server-list "10.59.8.35"
        set username "admin"
        set password *********
    next
end
```

3. Make sure that the ACI SDN connector status is up.

```
# diagnose sys sdn status aci_direct_van
SDN Connector                       Type        Status
------------------------------------------------------------
aci_direct_van                      aci-direct  Up
```

4. Create dynamic firewall addresses (both IPv4 and IPv6) associated with the ACI SDN connector, and apply specific filters without negation to filter out the required addresses (0.0.0.0/0 and ::/1).

   The resolved dynamic addresses should appear in the configuration for the dynamic firewall address.

```
config firewall address
    edit "aci_direct_v4_normal"
        set uuid 91e469ac-fe0f-51f0-4de7-6429a66130b0
        set type dynamic
        set sdn "aci_direct_van"
        set color 18
        set filter "L3OutName=lzou-ext & L3OutSubnet=negate-subnet-v4-1"
        config list
            edit "0.0.0.0/0"
            next
        end
```

```
   next
end
```

```
config firewall address6
    edit "aci_direct_v6_normal"
        set uuid cf662c1a-fe10-51f0-affc-f317cf137c9d
        set type dynamic
        set sdn "aci_direct_van"
        set color 18
        set filter "L3OutName=lzou-ext&L3OutSubnet=negate-subnet-v6-2"
        config list
            edit "::/1"
            next
        end
    next
end
```

5. Create two more dynamic firewall addresses (both IPv4 and IPv6) associated with the ACI SDN connector, and apply specific filters with negation to negate the resolved addresses in the previous step (0.0.0.0/0 and ::/1). The resolved dynamic addresses should appear in the configuration for the dynamic firewall address.

```
config firewall address
    edit "aci_direct_v4_negation1"
        set uuid 6221b4a8-f8bb-51f0-d612-2c5060e8d37e
        set type dynamic
        set sdn "aci_direct_van"
        set color 18
        set filter "L3OutName=lzou-ext & L3OutSubnet!=negate-subnet-v4-1"
        config list
            edit "0.0.0.0/1"
            next
            edit "128.0.0.0/1"
            next
            edit "172.16.165.0/24"
            next
            edit "192.168.1.0/24"
            next
        end
    next
end
```

```
config firewall address6
    edit "aci_direct_v6_negation1"
        set uuid 8f336364-f8bc-51f0-9ce5-dd75bc837471
        set type dynamic
        set sdn "aci_direct_van"
        set color 18
        set filter "L3OutName=lzou-ext&L3OutSubnet!=negate-subnet-v6-2"
        config list
            edit "2033::/64"
            next
```

```
            edit "8000::/1"
            next
            edit "::/0"
            next
            edit "::/64"
            next
        end
    next
end
```

6. Create two more dynamic firewall addresses (both IPv4 and IPv6) associated with the ACI SDN connector, and apply specific filters with negation and a wildcard (*) to negate the specified L3OutSubnet filter.
   The resolved dynamic addresses should appear in the configuration for the dynamic firewall address.

```
config firewall address
    edit "aci_direct_v4_negation_wild"
        set uuid 5791bd76-fe10-51f0-0df0-fc8b52314ff6
        set type dynamic
        set sdn "aci_direct_van"
        set color 18
        set filter "L3OutName=lzou-ext & L3OutSubnet!=negate-subnet-v4-*"
        config list
            edit "172.16.165.0/24"
            next
            edit "192.168.1.0/24"
            next
        end
    next
end
```

```
config firewall address6
    edit "aci_direct_v6_negation_wild"
        set uuid cf68fa76-fe10-51f0-0111-1c1ba3ae0912
        set type dynamic
        set sdn "aci_direct_van"
        set color 18
        set filter "L3OutName=lzou-ext&L3OutSubnet!=negate-subnet-v6-*"
        config list
            edit "2033::/64"
            next
            edit "::/64"
            next
        end
    next
end
```

7. Create four more dynamic firewall addresses (both IPv4 and IPv6) associated with the ACI SDN connector, and apply specific filters for the supported L3OutSubnetDescription and L3OutSubnetNameAlias.
   The resolved dynamic addresses should appear in the configuration for the dynamic firewall address.

```
config firewall address
    edit "aci_addr4_L3OutSubnetDescription"
        set uuid 7db47bcc-fe0d-51f0-ed6f-d9606aa2199a
        set type dynamic
        set sdn "aci_direct_van"
        set color 18
        set filter "L3OutName=lzou-ext & L3OutSubnetDescription=descr 192.168.1.0/24"
        config list
            edit "192.168.1.0/24"
            next
        end
    next
end
```

```
config firewall address
    edit "aci_addr4_L3OutSubnetNameAlias"
        set uuid 7db619be-fe0d-51f0-7760-4334cf575bd4
        set type dynamic
        set sdn "aci_direct_van"
        set color 18
        set filter "L3OutName=lzou-ext & L3OutSubnetNameAlias=alias-172.16.165.0"
        config list
            edit "172.16.165.0/24"
            next
        end
    next
end
```

```
config firewall address6
    edit "aci_addr6_L3OutSubnetDescription"
        set uuid 6dfe4df6-fe0e-51f0-4ddf-4d9c2f15ac45
        set type dynamic
        set sdn "aci_direct_van"
        set color 18
        set filter "L3OutName=lzou-ext&L3OutSubnetDescription=descr 2033::/64"
        config list
            edit "2033::/64"
            next
        end
    next
end
```

```
config firewall address6
    edit "aci_addr6_L3OutSubnetNameAlias"
        set uuid b5cb14a2-fe0e-51f0-87f4-ecb5ee03661c
        set type dynamic
        set sdn "aci_direct_van"
        set color 18
        set filter "L3OutName=lzou-ext | L3OutSubnetNameAlias=alias-192.168.1.0"
        config list
            edit "2033::/64"
            next
```

```
            edit "8000::/1"
            next
            edit "::/0"
            next
            edit "::/1"
            next
            edit "::/64"
            next
        end
    next
end
```

Log and report

This section includes information about logging and reporting related new features:

- Logging on page 547

Logging

This section includes information about logging related new features:

- Enhanced TACACS+ accounting log detail on page 547
- Custom log format support for syslog server on page 549
- Secure log upload enhancements with SFTP and LZ4 support on page 558

Enhanced TACACS+ accounting log detail

TACACS+ accounting logs are improved with full CLI change details to the `reason` field, providing more complete event log entries for external audit systems.

To test the new logs:

1. Configure the FortiGate and TACACS+ accounting server.
2. Verify that audit logging is enabled:

```
config system global
    set cli-audit-log enable
end
config log tacacs+accounting filter
    set login-audit enable
    set config-change-audit enable
    set cli-cmd-audit enable
end
```

3. Log on to the FortiGate using remote TACACS+ authentication.
4. Execute some tasks, such as adding a local user and changing port settings, then compare the information from the FortiGate log files with information collected in a PCAP file and from the external TACACS+accounting server.

   Adding a local user:

- FortiOS log:

```
# execute log display
14 logs found.
10 logs returned.

1: date=2026-01-16 time=07:04:38 eventtime=1768575878906101594 tz="-0800"
logid="0100032102" type="event" subtype="system" level="alert" vd="vdom1"
logdesc="Configuration changed" user="tac1" ui="https(172.16.200.254)" msg="Configuration
is changed in the admin session"
...
9: date=2026-01-16 time=07:03:01 eventtime=1768575781267058847 tz="-0800"
logid="0100032132" type="event" subtype="system" level="notice" vd="vdom1" logdesc="Local
user added" user="tac1" ui="https(172.16.200.254)" name="test101" status="enable"
msg="User tac1 added local user test101 from https(172.16.200.254)"
```

- Message extracted from PCAP file:

```
Frame 195: 210 bytes on wire (1680 bits), 210 bytes captured (1680 bits)
Ethernet II, Src: Fortinet_a3:50:f3 (80:80:2c:a3:50:f3), Dst: VMware_e2:b1:02
(00:0c:29:e2:b1:02)
Internet Protocol Version 4, Src: 10.1.100.8, Dst: 10.1.100.142
Transmission Control Protocol, Src Port: 6188, Dst Port: 49, Seq: 1, Ack: 1, Len: 144
TACACS+
    Major version: TACACS+
    Minor version: 0
    Type: Accounting (3)
    Sequence number: 1
    Flags: 0x00 (Encrypted payload, Multiple Connections)
    Session ID: 3656374799
    Packet length: 132
    Encrypted Request
    Decrypted Request
        Flags: 0x04
        Auth Method: NOT_SET (0x00)
        Privilege Level: 0
        Authentication type: Unknown (0)
        Service: Login (1)
        User len: 4
        User: tac1
        Port len: 0
        Remaddr len: 0
        Arg count: 4
        Arg[0] length: 17
        Arg[0] value: service=fortigate
        Arg[1] length: 14
        Arg[1] value: event=sys_acct
        Arg[2] length: 29
        Arg[2] value: stop_time=1768575783514241967
        Arg[3] length: 55
        Arg[3] value: reason="Add user.local test101 type[password]passwd[*]"
```

Changing port settings:

- Old message sent to TACACS+ accounting server:

```
Frame 28: 192 bytes on wire (1536 bits), 192 bytes captured (1536 bits)
...
        Arg count: 4
        Arg[0] length: 17
        Arg[0] value: service=fortigate
        Arg[1] length: 14
        Arg[1] value: event=sys_acct
        Arg[2] length: 29
        Arg[2] value: stop_time=1768578897378248400
        Arg[3] length: 36
        Arg[3] value: reason="Edit system.interface port5"
```

- New message sent to TACACS+ accounting server:

```
Frame 4: 300 bytes on wire (2400 bits), 300 bytes captured (2400 bits)
...
        Arg count: 4
        Arg[0] length: 17
        Arg[0] value: service=fortigate
        Arg[1] length: 14
        Arg[1] value: event=sys_acct
        Arg[2] length: 29
        Arg[2] value: stop_time=1768579536565487819
        Arg[3] length: 144
        Arg[3] value: reason="Edit system.interface port10 allowaccess[ping https ssh snmp
http telnet radius-acct->ping https ssh snmp http telnet radius-acct scim]"
```

Custom log format support for syslog server

This information is also available in the FortiOS 8.0 Administration Guide:

- Customizing log formats for syslog servers

You can now configure and send custom log formats to syslog servers using custom-format and log-template configurations. Previously, FortiGate only sent logs in fixed formats (CSV, CEF, RFC 5424, JSON). With this enhancement, you can tailor syslog output to match the exact requirements of your external logging system.

Use the feature by:

1. Configuring the custom log format.
2. Configuring the syslog logging devices to use the custom log format.

A new command is available:

```
config log custom-format
    edit <name>
        set field-exclusion-list {date | devid | devname | eventtime | level | logid | slot |
subtype | time | type | tz | vd | vsn}
        empty-value-indicator <string>
```

```
        config log-templates
            edit <name>
                set category <string>
                set subtypes <string>
                set template <string>
            next
        end
    next
end
```

Option	Description
`config log custom-format`	Configure custom log format for syslog servers.
`field-exclusion-list`	Specify log fields to exclude from the default format. Available fields for exclusion: • `date` • `devid` • `devname` • `eventtime` • `level` • `logid` • `slot` • `subtype`: When empty, all subtypes are included. • `time` • `type` • `tz` • `vd` • `vsn` The list applies when the log-template does not have a custom template.
`empty-value-indicator <string>`	Enter a character to indicate log field is empty. Supports only a single symbol, such as %, or empty (default = empty).
`config log-templates`	Custom log templates available for each category.
`*category <string>`	Set log category: • `traffic` • `event` • `virus` • `webfilter` • `attack` • `spam` • `anomaly` • `voip`: • `dlp` • `tyapp-ctrle`

Option	Description
	• waf • gtp • dns • ssh • ssl • file-filter • icap • virtual-patch • debug
subtypes <string>	Set log subtypes to apply template to: • forward • local • multicast • sniffer • ztna • http-transaction Can set one or more subtypes under the same category. The custom format template is applied to all specified subtypes. When no subtypes are set, the custom format template applies to all subtypes in the category.
*template <string>	Enter log template string. Template help text displays for the selected category. For example, the following help text is displayed when category is set to traffic: Please enter a log template string. Use syntax:$(fieldname) to insert the value of FortiOS log field. For example: "datetime=$(date):$(time) id=$(logid) my_field=abc_$(action)_xyz ..." "$(date) $(time) $(logid) abc_$(virus)_xyz ..." For utm related fields, use syntax: $(utmcategory.fieldname). For example: "abc_$(webfilter.url)" Please note that this is only available for traffic logs. To append to the default logs, use $(*) in the template. For example: "$(*) prefix=$(webfilter.url)postfix ..." Available log fields for category:traffic accessctrl, accessproxy, action, agent, ap,

Option	Description
	app, appact, appcat, appid, applist, apprisk, appstatus, apsn, authserver, centralnatid, channel, clientcert, clientdeviceems, clientdeviceid, clientdevicemanageable, clientdeviceowner, clientdevicetags, collectedemail, comment, countapp, countav, countcasb, countcifs, countdlp, countdns, countemail, countff, counticap, countips, countllm, countsctpf, countssh, countssl, countvpatch, countwaf, countweb, craction, crlevel, crscore, date, devcategory, devid, devname, devtype, devtype, dstauthserver, dstcity, dstcollectedemail, dstcountry, dstdevcategory, dstdevtype, dstdevtype, dstfamily, dsthwvendor, dsthwversion, dstinetsvc, dstintf, dstintfrole, dstip, dstmac, dstname, dstosname, dstosversion, dstport, dstregion, dstreputation, dstserver, dstssid, dstswversion, dstthreatfeed, dstunauthuser, dstunauthusersource, dstuser, dstuuid, dstzone, duration, durationdelta, emsconnection, emstag, emstag2, eventtime, fctuid, fwdsrv, gatewayid, group, hostname, httpmethod, identifier, lanin, lanout, level, logid, masterdstmac, mastersrcmac, msg, osname, osversion, pdstport, policyid, policymode, policyname, policytype, poluuid, proto, proxyapptype, psrcport, radioband, rcvdbyte, rcvddelta, rcvdpkt, rcvdpktdelta, realserverid, referralurl, replydstintf, replysrcintf, reqlength, reqtime, respfinishtime, resplength, resptime, resptype, saasname, scheme, sentbyte, sentdelta, sentpkt, sentpktdelta, service, sessionid, shaperdroprcvdbyte, shaperdropsentbyte, shaperperipdropbyte, shaperperipname, shaperrcvdname, shapersentname, shapingpolicyid, shapingpolicyname, signal, slot, snr, srccity, srccountry, srcdomain, srcfamily, srchwvendor, srchwversion, srcinetsvc, srcintf, srcintfrole, srcip, srcmac, srcmacvendor, srcname, srcport, srcregion, srcremote, srcreputation, srcserver, srcssid, srcswversion, srcthreatfeed, srcuuid, srczone, sslaction, statuscode, subtype, tcpnrt, tcporgrtrs, tcprplrtrs, tcprst, tcpsrt, tcpsynackrtrs, tcpsynrtrs, time, trandisp, tranip, tranport, transid, transip, transport, tunneldstip, tunnelid, tunnelproto, tunnelsessionid,

Option	Description
	, vip, vipincomingip, vlan, vpntype, vrf, vsn, vwlid, vwlname, vwlquality, vwlservice, vwpvlanid

A new option and parameter are available:

```
config log syslogd setting
    set format {default | csv | cef | rfc5424 | json | custom}
    set custom-log-format <string>
end
```

Option	Description
format {default | csv | cef | rfc5424 | json | custom}	Set the log format: • default: Syslog format. • csv: CSV (Comma Separated Values) form. • cef: CEF (Common Event Format) format. • rfc5424: Syslog RFC 5424 format. • json: JSON (JavaScript Object Notation) format. • custom: Custom format.
custom-log-format <string>	Enter the name of the customized format for logs. Available when format is set to custom. Before you can select the custom format, you must configure it using config log custom-format.

Custom log formats can also be configured for syslogd2, syslogd3, and syslogd-override under multi-VDOM:

- For syslogd2 and syslogd3 and the config log syslogd2 setting/config log syslogd3 setting commands, set format to custom, and select the log format.
- For syslogd override under vdom, enable syslog-override under config log setting. Then set format as custom, and select log format under config log syslogd override-setting.

This topic includes the following examples:

- Example 1: exclude fields on page 553
- Example 2: configure specific categories, subtypes, and fields on page 555
- Example 3: include UTM-related fields in traffic logs on page 556
- Example 4: use wildcard syntax on page 557

Example 1: exclude fields

In this example, logs sent from FortiGate to a syslog server are customized as follows:

- The percentage (%) character indicates empty values in the logs.
- The “date”, “eventtime”, “time” and “tz” fields are excluded from the logs.

To customize logs for syslog servers:

1. Set the custom log format:
   The `empty-value-indicator` can be a single character symbol, such as `%`.

```
config log custom-format
    edit “test”
        set empty-value-indicator  “%”
        set field-exclusion-list  "date" "eventtime" "time" "tz"
    end
end
```

2. Set the log template:

```
config log syslogd setting
    set status enable
    set server "172.16.200.55"
    set  facility local6
    set format custom
    set custom-log-format “test”
end
```

3. Send logs to a syslog server.
   Logs received by the syslog server exclude “date”, “eventtime”, “time”, and “tz” fields.
   - Traffic log example:

```
Feb  3 10:03:31 172.16.200.2 devname=FGT-B-LOG devid=FG201FT920901892  logid="0006000026"
type="traffic" subtype="http-transaction" level="notice" vd="vdom1" srcip=10.1.100.22
srcport=33012 dstip=172.16.200.55 dstport=80 sessionid=100663312 transid=100663315
action="accept" policyid=1 policytype="proxy-policy" poluuid="e66b90f6-008d-51f1-e5f4-
799c5495850b" url="http://172.16.200.55/" agent="curl/7.68.0" appcat="unscanned"
duration=0 reqlength=127 resplength=1145 rcvdbyte=1145 sentbyte=127 scheme="http"
hostname="172.16.200.55" resptype="normal" httpmethod="GET" statuscode="200"
reqtime=1770134611 resptime=1770134611 respfinishtime=1770134611
```

   - UTM log example:

```
Feb  2 15:31:47 172.16.200.2 devname=FGT-B-LOG devid=FG201FT920901892  logid="0211008192"
type="utm" subtype="virus" eventtype="infected" level="warning" vd="vdom1" policyid=1
poluuid="2a0da664-90c8-51ef-c42e-31dd5cc56aea" policytype="policy" msg="File is infected."
action="blocked" service="HTTP" sessionid=1009 srcip=10.1.100.22 dstip=172.16.200.55
srcport=56586 dstport=80 srccountry="Reserved" dstcountry="Reserved" srcintf="port1"
srcintfrole="undefined" dstintf="port2" dstintfrole="undefined" srcuuid="85bf1200-90ae-
51ef-7a00-759244bb35e4" dstuuid="85bf1200-90ae-51ef-7a00-759244bb35e4" proto=6
direction="incoming" filename="eicar.com" checksum="6851cf3c" quarskip="No-skip"
virus="EICAR_TEST_FILE" viruscat="Virus" dtype="cached" itype="infected"
ref="https://fortiguard.com/encyclopedia/virus/2172" virusid=2172
url="http://172.16.200.55/virus/eicar.com" profile="av-test" agent="curl/7.68.0"
httpmethod="GET"
analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

   - Event log example:

```
Feb  2 15:31:41 172.16.200.2 devname=FGT-B-LOG devid=FG201FT920901892  logid="0100020025"
type="event" subtype="system" level="notice" vd="vdom1" logdesc="Report generated
successfully" msg="Report generation succeeded for layout:default."
file="/var/log/log/vdom1/report/generated/On-Demand-default-2026-02-02-123140/default.pdf"
filesize=272974 datarange="01-Feb-2026 12:00 To 02-Feb-2026 11:59" reporttype="On-demand"
processtime=0"
```

Example 2: configure specific categories, subtypes, and fields

In this example, logs sent from FortiGate to a syslog server are customized as follows:

- The percentage (%) character indicates empty values in the logs.
- A log type category (`event`), subtype (`system`), and specific fields are configured for the logs.

FortiGate only sends logs for this category and subtype with the configured fields to the syslog server. This feature is supported for all categories.

To customize logs for syslog servers:

1. Configure a custom log format:

```
config log custom-format
    edit “test”
        set empty-value-indicator "%"
            config log-templates
                edit "1"
                   set category event
                   set subtypes "system"
                   set template “eve_type=$(subtype) eve_msg=$(msg) eve_vdom=$(vd) eve_
level=$(level) eve_status=$(status) eve_profile=$(profile)”
                next
            end
        next
    end
end
```

2. Set the log template:

```
config log syslogd setting
    set status enable
    set server "172.16.200.55"
    set  facility local6
    set format custom
    set custom-log-format “test”
end
```

3. Send logs to a syslog server.

   Logs received by the syslog server only include logs configured in template, and fields with empty value display %.

```
Feb  2 16:00:55 172.16.200.2 eve_type="system" eve_msg="Administrator admin login failed from
https(10.6.30.254) because of invalid password" eve_vdom="vdom1" eve_level="alert" eve_
status="failed" eve_profile=%
```

```
Feb  2 16:01:12 172.16.200.2 eve_type="system" eve_msg="Edit system.global " eve_vdom="vdom1"
eve_level="information" eve_status=% eve_profile=%
```

Example 3: include UTM-related fields in traffic logs

This example demonstrates how to include UTM-related fields in traffic logs. This feature is only available for traffic logs.

In this example, `category` is set to `traffic`, `subtype` is set to `forward`, and specific fields are configured for the logs FortiGate sends to a syslog server.

When a UTM field does not exist for the category, the `invalid log field name` error is displayed.

To customize logs for syslog servers:

1. Configure a custom log format:

```
config log custom-format
    edit "test"
        set empty-value-indicator ' '
        config log-templates
            edit "1"
                set category traffic
                set subtypes "forward"
                set template "type=$(type) subtype=$(subtype) http_status_code=$(statuscode)
url=$(virus.url) virus=$(virus.virus) agent=$(virus.agent) profile=$(virus.profile)"
            next
        end
    next
end
```

2. Set the log template:

```
config log syslogd setting
    set status enable
    set server "172.16.200.55"
    set  facility local6
    set format custom
    set custom-log-format “test”
end
```

3. Send logs to a syslog server.
   Logs received by the syslog server include selected traffic and UTM log fields.

```
Feb  3 11:44:07 172.16.200.2 type="traffic" subtype="forward" http_status_code=
url="http://172.16.200.55/virus/eicar.com" virus="EICAR_TEST_FILE" agent="curl/7.68.0"
profile="av-test"
```

Example 4: use wildcard syntax

This example demonstrates how to include `WILDCARD_SYNTAX: $(*)` in the template. Note:

- `$(*)` will be replaced by default-format log content.
- You can append additional UTM fields with the `WILDCARD_SYNTAX`.
- `$(*)` behaves like regular field in custom-format.log-templates and can include both prefixes and postfixes.
- Each template can only have one `WILDCARD_SYNTAX $(*)`. An error is displayed when more than one `$(*)` are used in the template.

To customize logs for syslog servers:

1. Configure a custom log format:
   When the custom log format has `exclusion-list` set, and the template is set for a specific category, then the template overrides the `exclusion-list` for the category.

```
config log custom-format
    edit "test"
        set field-exclusion-list "date" "eventtime" "time" "tz"
        set empty-value-indicator "%"
        config log-templates
            edit "1"
                set category traffic
                set subtypes "forward"
                set template "url=$(virus.url) virus=$(virus.virus) http_status_
code=$(statuscode) $(*) agent=$(virus.agent) profile=$(virus.profile)"
            next
        end
    next
end
```

2. Set the log template:

```
config log syslogd setting
    set status enable
    set server "172.16.200.55"
    set  facility local6
    set format custom
    set custom-log-format “test”
end
```

3. Send logs to a syslog server.
   Logs received by the syslog server includes all traffic log fields with selected UTM log fields.

```
Feb  3 09:27:45 172.16.200.2 date=2026-02-03 time=09:27:45 devname=FGT-B-LOG
devid=FG201FT920901892 url="http://172.16.200.55/virus/eicar.com" virus="EICAR_TEST_FILE"
http_status_code=% eventtime=1770139665020444596 tz="-0800" logid="0000000013" type="traffic"
subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.22 srcname="10.1.100.22"
srcport=40028 srcintf="port1" srcintfrole="undefined" dstip=172.16.200.55
dstname="172.16.200.55" dstport=80 dstintf="port2" dstintfrole="undefined"
srccountry="Reserved" dstcountry="Reserved" sessionid=5120 proto=6 action="close" policyid=1
```

```
policytype="policy" poluuid="2a0da664-90c8-51ef-c42e-31dd5cc56aea" policyname="B-out"
service="HTTP" trandisp="snat" transip=172.16.200.2 transport=40028 appcat="unscanned"
duration=30 sentbyte=1784 rcvdbyte=1242 sentpkt=31 rcvdpkt=6 utmaction="block" countav=1
crscore=50 craction=2 agent="curl/7.68.0" profile="av-test"
```

Secure log upload enhancements with SFTP and LZ4 support

FortiGate logs can now be uploaded in standard LZ4 log compression format to SFTP servers. Previously logs could only be uploaded in Fortinet FLZ4 formation to FTP servers. The ehancement improves log transfer security and flexibility while supporting both default FLZ4 and industry-standard LZ4 formats.

New options are available:

```
config log disk setting
    set upload-file-format {default | lz4}
    set upload-destination {ftp-server | sftp-server}
end
```

Option	Description
`upload-file-format {default | lz4}`	Configure the file format to be used for log files prior to being uploaded (default = default): • `default`: Upload rolled log files as they appear on disk. • `lz4`: Upload rolled log files with standard LZ4 format.
`upload-destination {ftp-server | sftp-server}`	Set upload destination (default = ftp-server): • `ftp-server`: Upload rolled log files to a FTP server. • `sftp-server`: Upload rolled log files to a SFTP server.

Example

This example describes how to:

- Upload log files in default file format to an SFTP server
- Enable a schedule to upload log files in standard LZ4 file format to an SFTP server

The default file format is Fortinet FLZ4. A third-party tool named *lz4_reader* can be used to decompress and convert FLZ4 log files to a readable .txt format. See Technical Tip: Transferring historical logs from a FortiGate hard disk to a FortiAnalyzer for more information.

The LZ4 file format is standard LZ4 compression. The standard Linux CLI tool named *lz4* can be used to decompress LZ4 files using `lz4 -d [log_filename].lz4`. The lz4 tool is a standard Linux CLI tool for LZ4 compression/decompression.

To upload log files in default format to an SFTP server:

1. On FortiGate, set SFTP server as the upload destination:

```
config log disk setting
    set status enable
    set upload enable
    set upload-destination sftp-server
    set uploadport 22
    set uploadpass *
    set uploaddir "fgtlog/sftp"
    set uploadip 172.16.200.55
    set uploaduser "pc5user2"
end
```

2. Run `execute log roll`.
3. FortiGate uploads the log files to the SFTP server.
   The `diagnose debug application uploadd -1` command can be used to view upload activity.
4. On the SFTP server, view the log files in the correct directory:

```
pc5user2@logging_pc05:/fgtlog/sftp# ls
elog.FG201FT920901892.vdom1.65508.20260303110938
mlog.FG201FT920901892.vdom1.65516.20260303110659
plog.FG201FT920901892.vdom1.65516.20260303110659
tlog.FG201FT920901892.vdom1.65505.20260303110938
```

To upload log files in standard LZ4 format to an SFTP server:

1. On FortiGate, set the file format to LZ4, set destination to SFTP server, and enable an upload schedule:

```
config log disk setting
    set status enable
    set upload enable
    set upload-destination sftp-server
    set upload-file-format lz4
    set uploadport 22
    set uploadpass *
    set uploaddir "fgtlog/sftp”
    set uploadsched enable
    set uploadtime 11:50
    set uploadip 172.16.200.55
    set uploaduser "pc5user2"
end
```

2. FortiGate uploads the log files to the SFTP server on schedule.
   FortiGate logs include upload details:

```
1: date=2026-03-09 time=11:51:06 eventtime=1773078605563616801 tz="-0700" logid="0100020108"
type="event" subtype="system" level="notice" vd="vdom1" logdesc="Log upload completed"
action="upload" status="completed" user="pc5user2" server="172.16.200.55" port=22 msg="Log
upload to SFTP completed on vdom vdom1"
```

```
2: date=2026-03-09 time=11:50:06 eventtime=1773078605080086909 tz="-0700" logid="0100032051"
type="event" subtype="system" level="notice" vd="vdom1" logdesc="Disk logs upload started"
ui="uploadd" action="upload" status="start" msg="Start uploading disk logs to SFTP from vdom
vdom1."
```

3. On the SFTP server, view the log files in the correct directory in standard LZ4 format:

```
pc5user2@logging_pc05:/fgtlog/sftp# ls
elog.FG201FT920901892.vdom1.65506.20260309115102.lz4
mlog.FG201FT920901892.vdom1.65515.20260309115102.lz4
plog.FG201FT920901892.vdom1.65515.20260309115102.lz4
tlog.FG201FT920901892.vdom1.65503.20260309115041
```

The `tlog` file is a tar package that includes traffic logs and most UTM logs:

```
pc5user2@logging_pc05:/fgtlog/sftp# tar –
xvf   tlog.FG201FT920901892.vdom1.65503.20260309115041
tlog.65503.lz4
vlog.65503.lz4
wlog.65503.lz4
alog.65503.lz4
slog.65503.lz4
dlog.65503.lz4
rlog.65503.lz4
flog.65503.lz4
olog.65503.lz4
hlog.65503.lz4
ssllog.65503.lz4
fflog.65503.lz4
gclog.65503.lz4
sctplog.65503.lz4
vplog.65503.lz4
casblog.65503.lz4
```

4. View FTP server and SFTP server information using the following command:

```
# diagnose  test application uploadd 1
vdom_admin:1
have_disk:1
in_ha_mode:0
ha_vfid:3
mgmt_vfid:1
ha_direct:0
hamgmt_vfid:-1
global:

vdom:0-root

vdom:1-vdom1
        FTP info:0x7fe370f2a908 ip:172.16.200.55 port:21 override:1 sch:0x7fe370f2aa00
option:1 intf_sel:auto() vrf:0
```

```
                next_upload_time:2522399024
        ftp state:0 file:

        SFTP info:0x7fe370f2ac80 ip:172.16.200.55 port:22 override:1 sch:0x7fe370f2ad78
option:1 intf_sel:auto() vrf:0
                next_upload_time:85828
sftp: local: remote:/fgtlog/sftp
        server info:
                dst:172.16.200.55:22 src:0.0.0.0
                user:pc5user2 pass:*
                vdom:vdom1(1)
```

Cloud

This section includes information about cloud related new features:

- Public and private cloud on page 562

Public and private cloud

This section includes information about public and private cloud related new features:

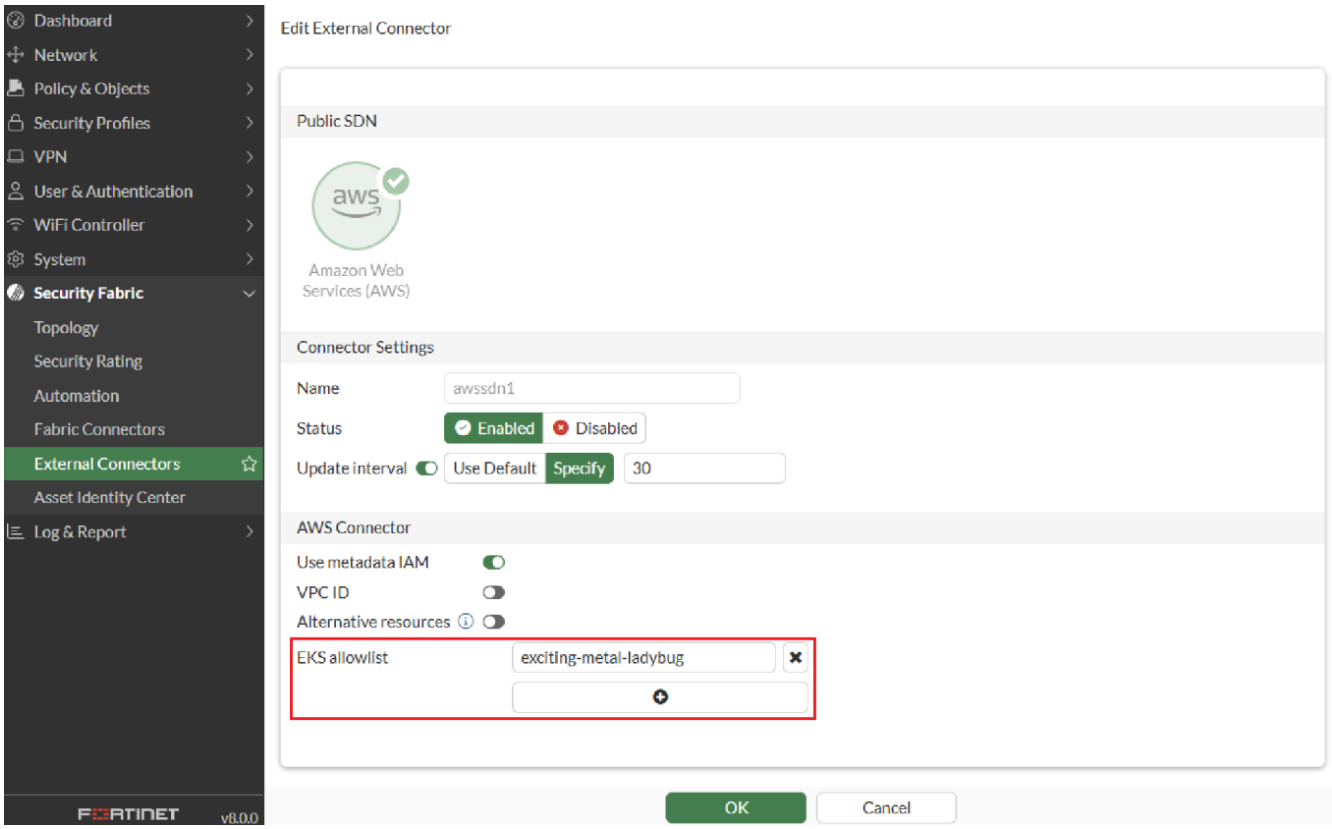
- AWS SDN connector EKS filtering support on page 562

AWS SDN connector EKS filtering support

The AWS SDN connector now supports filtering of EKS clusters. If the allow-list contains at least one entry, only clusters explicitly included in the allow-list will be resolved by the AWS SDN connector. Clusters not in the allow-list will be excluded from dynamic address object resolution. As a result, AWS resource resolution performance is improved.

EKS filter configuration

The *EKS allowlist* field is available when creating or editing an AWS SDN connector. The field supports listing multiple EKS/Kubernetes cluster names.



After an allow-list is configured, the AWS SDN connector will resolve only the specified EKS/Kubernetes clusters.

CLI configuration example

```
config system sdn-connector
    edit "awssdn1"
        set status enable
        set type aws
        set use-metadata-iam enable
        set update-interval 30
        set k8s-allow-list "exciting-metal-ladybug"
    next
end
```

Verifying the EKS filtering

Run debug commands to verify EKS allowlist filters on your desired clusters:

```
# diagnose debug reset
# diagnose debug application awsd -1
# diagnose debug enable

…
2026-04-13 16:07:21 EKS cluster adorable-country-outfit is skipped by k8s allowlist.
2026-04-13 16:07:21 awsd getting IPs from EKS cluster: exciting-metal-ladybug (us-east-1),
```

```
endpoint: https://62C90##############4FC4CC9FA8.gr7.us-east1.eks.amazonaws.com
…

2026-04-13 16:07:51 EKS cluster adorable-country-outfit is skipped by k8s allowlist.
2026-04-13 16:07:51 awsd getting IPs from EKS cluster: exciting-metal-ladybug (us-east-1),
endpoint: https://62C90##############4FC4CC9FA8.gr7.us-east1.eks.amazonaws.com
2026-04-13 16:07:51 kube host: 62C90##############4FC4CC9FA8.gr7.us-east-
1.eks.amazonaws.com:443:10.0.1.115
2026-04-13 16:07:51 k8s service ip: 172.20.0.1, svcname: kubernetes, namespace: default
2026-04-13 16:07:51 k8s service ip: 172.20.81.114, svcname: eks-extension-metrics-api, namespace:
kube-system
2026-04-13 16:07:51 k8s service ip: 172.20.116.198, svcname: metrics-server, namespace: kube-
system
2026-04-13 16:07:51 kube host: 62C90##############4FC4CC9FA8.gr7.us-east-
1.eks.amazonaws.com:443:10.0.1.115
2026-04-13 16:07:51 k8s node ip: 10.0.1.215, nodename: i-0b27e094dc6ee09e1, zone: us-east-1a
2026-04-13 16:07:51 k8s node ip: 98.XXX.XXX.XX, nodename: i-0b27e094dc6ee09e1, zone: us-east-1a
2026-04-13 16:07:51 k8s node ip: 10.0.11.224, nodename: i-0b6403fce57f51e77, zone: us-east-1b
2026-04-13 16:07:51 k8s node ip: 98.89.27.30, nodename: i-0b6403fce57f51e77, zone: us-east-1b
2026-04-13 16:07:51 kube host: 62C90##############4FC4CC9FA8.gr7.us-east-
1.eks.amazonaws.com:443:10.0.1.115
2026-04-13 16:07:51 k8s pod ip: 10.0.11.176, podname: metrics-server-#########-sbdzd, namespace:
kube-system, zone: us-east-1b
2026-04-13 16:07:51 k8s pod ip: 10.0.11.176, podname: metrics-server-#########-sbdzd, namespace:
kube-system, zone: us-east-1b
2026-04-13 16:07:51 k8s pod ip: 10.0.1.6, podname: metrics-server-#########-z8q7j, namespace:
kube-system, zone: us-east-1a
2026-04-13 16:07:51 k8s pod ip: 10.0.1.6, podname: metrics-server-#########-z8q7j, namespace:
kube-system, zone: us-east-1a
2026-04-13 16:07:51 awsd sdn connector awssdn1 start updating IP addresses
2026-04-13 16:07:51 awsd checking firewall address object awsaddr1, vd 0
2026-04-13 16:07:51 awsd checking firewall address object awsaddr2, vd 0
2026-04-13 16:07:51 awsd sdn connector awssdn1 finish updating IP addresses 2026-04-13 16:07:51
awsd reap child pid: 3974
```

FortiOS Carrier

This section includes information about FortiOS Carrier new features:

- Selective GTP FGSP sync: only S10 tunnels on page 565
- Selective GTP FGSP sync: only synchronize GTP tunnels on page 569
- New RAT types on page 573
- 3G to 4G/5G handover support on page 575
- Increase GTP tunnel limit per profile to 50M tunnels on page 577

See also FortiOS Carrier.

Selective GTP FGSP sync: only S10 tunnels

You can use the following option to configure a FortiOS Carrier FGSP cluster to synchronize only S10 GTP tunnels:

```
config system settings
   set gtp-fgsp-s10-only {disable | enable}
end
```

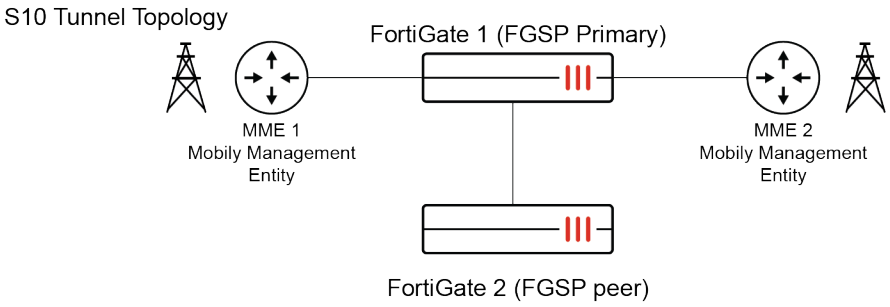
By default `gtp-fgsp-s10-only` is disabled and FortiOS Carrier FGSP synchronizes all GTP tunnels. You can enable `gtp-fgsp-s10-only` to synchronize only G10 tunnels.

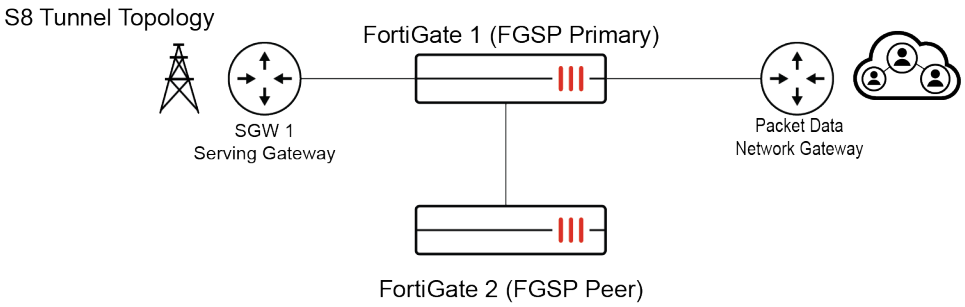
If you are using FortiOS Carrier FGSP to synchronize GTP tunnels among multiple locations, and if you only need to synchronize S10 GTP tunnels among those locations, you can enable `gtp-fgsp-s10-only`. Enabling this option supports the GTP tunnel synchronization you need and potentially reduces the amount of bandwidth required for FGSP session synchronization.

Enabling `gtp-fgsp-s10-only` is compatible with disabling `session-sync` to stop synchronizing IP sessions, see Selective GTP FGSP sync: only synchronize GTP tunnels on page 569. If you disable `session-sync` and enable `gtp-fgsp-s10-only`, FortiOS Carrier FGSP only synchronizes S10 GTP tunnels and also does not synchronize IP sessions.

When `gtp-fgsp-s10-only` is enabled, FGSP support for GTP asymmetric routing is no longer supported and the `gtp-asym-fgsp` system settings CLI option becomes hidden.

Example topology





Default configuration: gtp-fgsp-s10-only disabled

CLI configuration:

```
config system settings
   set gtp-fgsp-s10-only disable
end
```

All GTP tunnels are received by FortiGate 1 (FGSP primary). S10 and S8 GTP tunnels are synchronized to FortiGate 2 (FGSP peer).

The output of the `diagnose firewall gtp tunnel list` command for each FortiGate shows S10 and S8 tunnels:

FortiGate 1 (FGSP primary)

```
diagnose firewall gtp tunnel list
list gtp tunnels

----------prof=gtpp ref=6 imsi=987654112233445 msisdn=896745214365 mei=43658709.212030.1 ms_
addr=80.80.80.1 s11_s4 0---------- (S8 Tunnel)
----------index=00000013 life=23(sec) idle=22(sec) vd=0  ver=2----------
c_pkt=2 c_bytes=540 u_pkt=0 u_bytes=0
rat type: utran
downlink cfteid:
    addr=10.10.10.10 teid=0x1ce7eab0 role=control vd=0 intf_type=s5/s8 sgw gtp-c
uplink cfteid:

    addr=20.20.20.10 teid=0x1ce7eab2 role=control vd=0 intf_type=s5/s8 pgw gtp-c
```

```
1/1 bearers:
    id=6 linked_id=6 type=regular dead=0 apn=internet2 selection=ms-or-net-provided-apn apn_
restriction=public-2 user_addr=80.80.80.1 u_pkt=0 u_bytes=0
    2 fteids:
        addr=10.10.10.10 teid=0x1ce7eab1 role=data vd=0 intf_type=s5/s8 sgw gtp-u
        addr=20.20.20.10 teid=0x1ce7eab3 role=data vd=0 intf_type=s5/s8 pgw gtp-u

-----------prof=gtpp ref=5 imsi=280202019012163 msisdn=unknown mei=unknown ms_addr=Unknown s11_s4
0----------- (S10 Tunnel)
-----------index=00000014 life=5(sec) idle=5(sec) vd=0  ver=2-----------
c_pkt=1 c_bytes=52 u_pkt=0 u_bytes=0
rat type: eutran
downlink cfteid:
    addr=Unknown teid=0x00000000 role=control vd=0 intf_type=s1-u enodeb gtp-u
uplink cfteid:

    addr=194.154.140.241 teid=0x024e88f4 role=control vd=0 intf_type=s5/s8 pgw gtp-c
1/1 bearers:
    id=5 linked_id=5 type=regular dead=0 apn=ip.primetel.MNC020.MCC280.GPRS selection=ms-or-net-
provided-apn user_addr=10.131.138.209 u_pkt=0 u_bytes=0
    2 fteids:
        addr=10.144.1.164 teid=0x024c88f4 role=data vd=0 intf_type=s1-u sgw gtp-u
        addr=194.154.140.241 teid=0x024c88f4 role=data vd=0 intf_type=s5/s8 pgw gtp-u
```

FortiGate 2 (FGSP peer)

```
diagnose firewall gtp tunnel list
list gtp tunnels

-----------prof=gtpp ref=5 imsi=987654112233445 msisdn=896745214365 mei=43658709.212030.1 ms_
addr=80.80.80.1 s11_s4 0----------- (S8 Tunnel)
-----------index=00000013 life=30(sec) idle=30(sec) vd=0  ver=2-----------
c_pkt=0 c_bytes=0 u_pkt=0 u_bytes=0
rat type: utran
downlink cfteid:
    addr=10.10.10.10 teid=0x1ce7eab0 role=control vd=0 intf_type=s5/s8 sgw gtp-c
uplink cfteid:

    addr=20.20.20.10 teid=0x1ce7eab2 role=control vd=0 intf_type=s5/s8 pgw gtp-c
1/1 bearers:
    id=6 linked_id=6 type=regular dead=0 apn=internet2 selection=ms-or-net-provided-apn apn_
restriction=public-2 user_addr=80.80.80.1 u_pkt=0 u_bytes=0
    2 fteids:
        addr=10.10.10.10 teid=0x1ce7eab1 role=data vd=0 intf_type=s5/s8 sgw gtp-u
        addr=20.20.20.10 teid=0x1ce7eab3 role=data vd=0 intf_type=s5/s8 pgw gtp-u

-----------prof=gtpp ref=4 imsi=280202019012163 msisdn=unknown mei=unknown ms_addr=Unknown s11_s4
0----------- (S10 Tunnel)
-----------index=00000014 life=11(sec) idle=11(sec) vd=0  ver=2-----------
c_pkt=0 c_bytes=0 u_pkt=0 u_bytes=0
rat type: eutran
downlink cfteid:
```

```
    addr=Unknown teid=0x00000000 role=control vd=0 intf_type=s1-u enodeb gtp-u
uplink cfteid:

    addr=194.154.140.241 teid=0x024e88f4 role=control vd=0 intf_type=s5/s8 pgw gtp-c
1/1 bearers:
    id=5 linked_id=5 type=regular dead=0 apn=ip.primetel.MNC020.MCC280.GPRS selection=ms-or-net-
provided-apn user_addr=10.131.138.209 u_pkt=0 u_bytes=0
    2 fteids:
        addr=10.144.1.164 teid=0x024c88f4 role=data vd=0 intf_type=s1-u sgw gtp-u
        addr=194.154.140.241 teid=0x024c88f4 role=data vd=0 intf_type=s5/s8 pgw gtp-u
```

Sync S10 GTP tunnels only: gtp-fgsp-s10-only enabled

CLI configuration:

```
config system settings
   set gtp-fgsp-s10-only enabled
end
```

All GTP tunnels are received by FortiGate 1 (FGSP primary). Only S10 GTP tunnels are synchronized to FortiGate 2 (FGSP peer).

The output of the `diagnose firewall gtp tunnel list` command for each FortiGate shows FortiGate 1 (FGSP primary) has S10 and S8 tunnels and FortiGate 2 (FGSP peer) has only S8 tunnels:

FortiGate 1 (FGSP primary)

```
diagnose firewall gtp tunnel list
list gtp tunnels

-----------prof=gtpp ref=6 imsi=987654112233445 msisdn=896745214365 mei=43658709.212030.1 ms_
addr=80.80.80.1 s11_s4 0----------- (S8 Tunnel)
-----------index=00000017 life=30(sec) idle=29(sec) vd=0  ver=2-----------
c_pkt=2 c_bytes=540 u_pkt=0 u_bytes=0
rat type: utran
downlink cfteid:
    addr=10.10.10.10 teid=0x1ce7eab0 role=control vd=0 intf_type=s5/s8 sgw gtp-c
uplink cfteid:

    addr=20.20.20.10 teid=0x1ce7eab2 role=control vd=0 intf_type=s5/s8 pgw gtp-c
1/1 bearers:
    id=6 linked_id=6 type=regular dead=0 apn=internet2 selection=ms-or-net-provided-apn apn_
restriction=public-2 user_addr=80.80.80.1 u_pkt=0 u_bytes=0
    2 fteids:
        addr=10.10.10.10 teid=0x1ce7eab1 role=data vd=0 intf_type=s5/s8 sgw gtp-u
        addr=20.20.20.10 teid=0x1ce7eab3 role=data vd=0 intf_type=s5/s8 pgw gtp-u

-----------prof=gtpp ref=5 imsi=280202019012163 msisdn=unknown mei=unknown ms_addr=Unknown s11_s4
0----------- (S10 Tunnel)
-----------index=00000018 life=19(sec) idle=19(sec) vd=0  ver=2-----------
c_pkt=1 c_bytes=52 u_pkt=0 u_bytes=0
rat type: eutran
downlink cfteid:
```

```
    addr=Unknown teid=0x00000000 role=control vd=0 intf_type=s1-u enodeb gtp-u
uplink cfteid:
    addr=194.154.140.241 teid=0x024e88f4 role=control vd=0 intf_type=s5/s8 pgw gtp-c
1/1 bearers:
    id=5 linked_id=5 type=regular dead=0 apn=ip.primetel.MNC020.MCC280.GPRS selection=ms-or-net-
provided-apn user_addr=10.131.138.209 u_pkt=0 u_bytes=0
    2 fteids:
        addr=10.144.1.164 teid=0x024c88f4 role=data vd=0 intf_type=s1-u sgw gtp-u
        addr=194.154.140.241 teid=0x024c88f4 role=data vd=0 intf_type=s5/s8 pgw gtp-u
```

FortiGate 2 (FGSP peer)

```
diagnose firewall gtp tunnel list
list gtp tunnels

-----------prof=gtpp ref=4 imsi=280202019012163 msisdn=unknown mei=unknown ms_addr=Unknown s11_s4
0-----------  (S10 Tunnel)
-----------index=00000018 life=25(sec) idle=25(sec) vd=0  ver=2-----------
c_pkt=0 c_bytes=0 u_pkt=0 u_bytes=0
rat type: eutran
downlink cfteid:
    addr=Unknown teid=0x00000000 role=control vd=0 intf_type=s1-u enodeb gtp-u
uplink cfteid:
    addr=194.154.140.241 teid=0x024e88f4 role=control vd=0 intf_type=s5/s8 pgw gtp-c
1/1 bearers:
    id=5 linked_id=5 type=regular dead=0 apn=ip.primetel.MNC020.MCC280.GPRS selection=ms-or-net-
provided-apn user_addr=10.131.138.209 u_pkt=0 u_bytes=0
    2 fteids:
        addr=10.144.1.164 teid=0x024c88f4 role=data vd=0 intf_type=s1-u sgw gtp-u
        addr=194.154.140.241 teid=0x024c88f4 role=data vd=0 intf_type=s5/s8 pgw gtp-u
```

Selective GTP FGSP sync: only synchronize GTP tunnels

You can use the following option to disable IP session synchronization for an FGSP cluster:

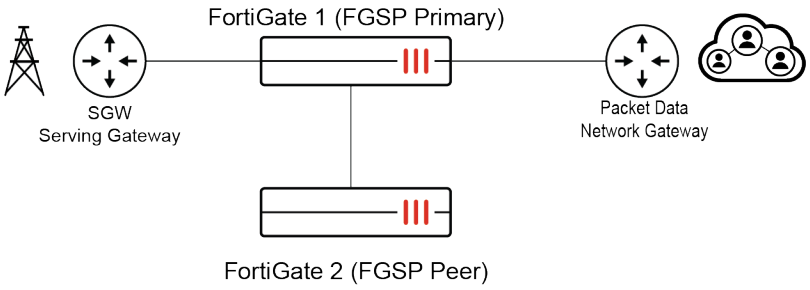
```
config system standalone-cluster
   set session-sync {disable | enable}
end
```

By default `session-sync` is enabled and FortiOS Carrier FGSP operates normally. You can disable `session-sync` to stop synchronizing IP sessions. GTP tunnel synchronization is not changed. Disabling IP session synchronization potentially reduces the amount of bandwidth required for FGSP session synchronization.

Disabling `session-sync` is compatible with enabling `gtp-fgsp-s10-only` to synchronize only S10 GTP tunnels, see Selective GTP FGSP sync: only S10 tunnels on page 565. If you disable `session-sync` and enable `gtp-fgsp-s10-only`, FortiOS Carrier FGSP will only synchronize S10 GTP tunnels and also will not synchronize IP sessions.

When `session-sync` is disabled, FGSP support for GTP asymmetric routing is no longer supported and the `gtp-asym-fgsp` system settings CLI option becomes hidden.

Example topology



Default configuration: session-sync enabled

CLI configuration:

```
config system standalone-cluster
   set session-sync enable
end
```

All IP sessions, GTP sessions, and GTP tunnels are received by FortiGate 1 (FGSP primary). IP sessions, GTP sessions, and GTP tunnels are synchronized to FortiGate 2 (FGSP peer).

FortiGate 1 (FGSP primary)

The output of `diagnose sys session list | grep 172.16.200.61` shows IP sessions on FortiGate 1 (FGSP primary):

```
diagnose sys session list | grep 172.16.200.61
hook=pre dir=org act=noop 10.1.100.60:52448->172.16.200.61:22(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.61:22->10.1.100.60:52448(0.0.0.0:0)
```

The output of `diagnose sys session list | grep 2123` shows the GTP sessions on FortiGate 1 (FGSP primary):

```
diagnose sys session list | grep 2123
hook=pre dir=org act=noop 172.16.200.91:2123->10.1.100.91:2123(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.91:2123->172.16.200.91:2123(0.0.0.0:0)
```

The output of the `diagnose firewall gtp tunnel list` command shows the GTP tunnels on FortiGate 1 (FGSP primary):

```
diagnose firewall gtp tunnel list
list gtp tunnels

----------prof=gtpp ref=6 imsi=987654112233445 msisdn=896745214365 mei=43658709.212030.1 ms_
addr=Unknown s11_s4 0----------
----------index=00000003 life=34(sec) idle=34(sec) vd=0  ver=2----------
c_pkt=1 c_bytes=276 u_pkt=0 u_bytes=0
```

```
rat type: utran
downlink cfteid:
    addr=10.10.10.10 teid=0x1ce7eab0 role=control vd=0 intf_type=s5/s8 sgw gtp-c
    1/1 requests shown:
        src=10.1.100.91:2123 dst=172.16.200.91:2123 seq=10017643 msg_type=32 vd=0 ver=2
uplink cfteid:
    addr=Unknown teid=0x00000000 role=control vd=0 intf_type=s1-u enodeb gtp-u
1/1 bearers:
    id=6 linked_id=6 type=regular dead=0 apn=internet2 selection=ms-or-net-provided-apn apn_
restriction=public-2 u_pkt=0 u_bytes=0
    1 fteids:
        addr=10.10.10.10 teid=0x1ce7eab1 role=data vd=0 intf_type=s5/s8 sgw gtp-u
```

FortiGate 2 (FGSP peer)

The output of `diagnose sys session list | grep 172.16.200.61` shows the IP sessions synchronized to FortiGate 2 (FGSP peer):

```
diagnose sys session list | grep 172.16.200.61
hook=pre dir=org act=noop 10.1.100.60:52448->172.16.200.61:22(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.61:22->10.1.100.60:52448(0.0.0.0:0)
```

The output of `diagnose sys session list | grep 2123` shows the GTP sessions synchronized to FortiGate 2 (FGSP peer):

```
diagnose sys session list | grep 2123
hook=pre dir=org act=noop 172.16.200.91:2123->10.1.100.91:2123(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.91:2123->172.16.200.91:2123(0.0.0.0:0)
```

The output of the `diagnose firewall gtp tunnel list` command shows the GTP tunnels synchronized to FortiGate 2 (FGSP peer):

```
diagnose firewall gtp tunnel list
list gtp tunnels

-----------prof=gtpp ref=5 imsi=987654112233445 msisdn=896745214365 mei=43658709.212030.1 ms_
addr=80.80.80.1 s11_s4 0-----------
-----------index=00000001 life=4887(sec) idle=4887(sec) vd=0  ver=2-----------
c_pkt=0 c_bytes=0 u_pkt=0 u_bytes=0
rat type: utran
downlink cfteid:
    addr=10.10.10.10 teid=0x1ce7eab0 role=control vd=0 intf_type=s5/s8 sgw gtp-c
uplink cfteid:
    addr=20.20.20.10 teid=0x1ce7eab2 role=control vd=0 intf_type=s5/s8 pgw gtp-c
1/1 bearers:
    id=6 linked_id=6 type=regular dead=0 apn=internet2 selection=ms-or-net-provided-apn apn_
restriction=public-2 user_addr=80.80.80.1 u_pkt=0 u_bytes=0
    2 fteids:
        addr=10.10.10.10 teid=0x1ce7eab1 role=data vd=0 intf_type=s5/s8 sgw gtp-u
        addr=20.20.20.10 teid=0x1ce7eab3 role=data vd=0 intf_type=s5/s8 pgw gtp-u
```

Only GTP tunnels synchronized: session-sync disabled

FortiGate 1 (FGSP primary)

The output of `diagnose sys session list | grep 172.16.200.61` shows IP sessions on FortiGate 1 (FGSP primary):

```
diagnose sys session list | grep 172.16.200.61
hook=pre dir=org act=noop 10.1.100.60:55136->172.16.200.61:22(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.61:22->10.1.100.60:55136(0.0.0.0:0)
```

The output of `diagnose sys session list | grep 2123` shows the GTP sessions on FortiGate 1 (FGSP primary):

```
diagnose sys session list | grep 2123
hook=pre dir=org act=noop 172.16.200.91:2123->10.1.100.91:2123(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.91:2123->172.16.200.91:2123(0.0.0.0:0)
```

The output of the `diagnose firewall gtp tunnel list` command shows the GTP tunnels on FortiGate 1 (FGSP primary):

```
diagnose firewall gtp tunnel list
list gtp tunnels

-----------prof=gtpp ref=6 imsi=987654112233445 msisdn=896745214365 mei=43658709.212030.1 ms_
addr=Unknown s11_s4 0-----------
-----------index=00000002 life=85(sec) idle=85(sec) vd=0  ver=2-----------
c_pkt=1 c_bytes=276 u_pkt=0 u_bytes=0
rat type: utran
downlink cfteid:
    addr=10.10.10.10 teid=0x1ce7eab0 role=control vd=0 intf_type=s5/s8 sgw gtp-c
    1/1 requests shown:
        src=10.1.100.91:2123 dst=172.16.200.91:2123 seq=10017643 msg_type=32 vd=0 ver=2
uplink cfteid:
    addr=Unknown teid=0x00000000 role=control vd=0 intf_type=s1-u enodeb gtp-u
1/1 bearers:
    id=6 linked_id=6 type=regular dead=0 apn=internet2 selection=ms-or-net-provided-apn apn_
restriction=public-2 u_pkt=0 u_bytes=0
    1 fteids:
        addr=10.10.10.10 teid=0x1ce7eab1 role=data vd=0 intf_type=s5/s8 sgw gtp-u
```

FortiGate 2 (FGSP peer)

The output of `diagnose sys session list | grep 172.16.200.61` shows that IP sessions are not synchronized to FortiGate 2 (FGSP peer):

```
diagnose sys session list | grep 172.16.200.61
```

The output of `diagnose sys session list | grep 2123` shows that GTP sessions are not synchronized to FortiGate 2 (FGSP peer):

```
diagnose sys session list | grep 2123
```

The output of the `diagnose firewall gtp tunnel list` command shows the GTP tunnels synchronized to FortiGate 2 (FGSP peer):

```
diagnose firewall gtp tunnel list
list gtp tunnels

----------prof=gtpp ref=5 imsi=987654112233445 msisdn=896745214365 mei=43658709.212030.1 ms_
addr=80.80.80.1 s11_s4 0----------
----------index=00000005 life=10250(sec) idle=8667(sec) vd=0  ver=2----------
c_pkt=0 c_bytes=0 u_pkt=0 u_bytes=0
rat type: utran
downlink cfteid:
    addr=10.10.10.10 teid=0x1ce7eab0 role=control vd=0 intf_type=s5/s8 sgw gtp-c
uplink cfteid:
    addr=20.20.20.10 teid=0x1ce7eab2 role=control vd=0 intf_type=s5/s8 pgw gtp-c
1/1 bearers:
    id=6 linked_id=6 type=regular dead=0 apn=internet2 selection=ms-or-net-provided-apn apn_
restriction=public-2 user_addr=80.80.80.1 u_pkt=0 u_bytes=0
    2 fteids:
        addr=10.10.10.10 teid=0x1ce7eab1 role=data vd=0 intf_type=s5/s8 sgw gtp-u
        addr=20.20.20.10 teid=0x1ce7eab3 role=data vd=0 intf_type=s5/s8 pgw gtp-u
```

New RAT types

FortiOS Carrier supports filtering for twelve new Radio Access Technology (RAT) types. These RAT types were introduced in 3GPP TS 29.274 Release 17. See section 8.17 for the list of RAT types.

The new supported RAT types are:

- WB-E-UTRAN(LEO)
- WB-E-UTRAN(MEO)
- WB-E-UTRAN(GEO)
- WB-E-UTRAN(OTHERSAT)
- EUTRAN-NB-IoT(LEO)
- EUTRAN-NB-IoT(MEO)
- EUTRAN-NB-IoT(GEO)
- EUTRAN-NB-IoT(OTHERSAT)
- LTE-M(LEO)
- LTE-M(MEO)
- LTE-M(GEO)
- LTE-M(OTHERSAT)

RAT timeout profiles

The new RAT types can be added to RAT timeout profiles using the following command:

```
config gtp rat-timeout-profile
   edit <rat-timeout-profile-name>
```

```
      set wb-eutran-leo-timeout <timeout>
      set wb-eutran-meo-timeout <timeout>
      set wb-eutran-geo-timeout <timeout>
      set wb-eutran-othersat-timeout <timeout>
      set nbiot-leo-timeout <timeout>
      set nbiot-meo-timeout <timeout>
      set nbiot-geo-timeout <timeout>
      set nbiot-othersat-timeout <timeout>
      set ltem-leo-timeout <timeout>
      set ltem-meo-timeout <timeout>
      set ltem-geo-timeout <timeout>
      set ltem-othersat-timeout <timeout>
end
```

For each of the timeouts, the range is 0 to 4294967295 seconds. The default is 0, which means no RAT timeout is set, and the GTP tunnel timeout is determined by the**Tunnel Timeout** option (CLI `tunnel-timeout`) of the GTP profile.

For more information about RAT timeout profiles, see GTP RAT timeout profiles.

GTP v2 policy filtering

You can also add the new RAT types when configuring GTPv2 policy filtering:

```
config firewall gtp
   edit <name>
      set policy-filter enable
      set default-policy-action {allow | deny}
         config policy-v2
            edit <id>
               set rat-type <rat-type>
            end
```

`<rat-type>` can be:

`wb-eutran-leo`

`wb-eutran-meo`

`wb-eutran-geo`

`wb-eutran-othersat`

`nbiot-leo`

`nbiot-meo`

`nbiot-geo`

`nbiot-othersat`

`ltem-leo`

`ltem-meo`

`ltem-geo`

`ltem-othersat`

For more information about GTP v2 policy filtering, see Adding GTPv2 policy filters to a GTP profile.

Viewing the RAT type associated with a GTP tunnel

You can use the `diagnose firewall gtp tunnel list` command to view the RAT type associated with a GTP tunnel:

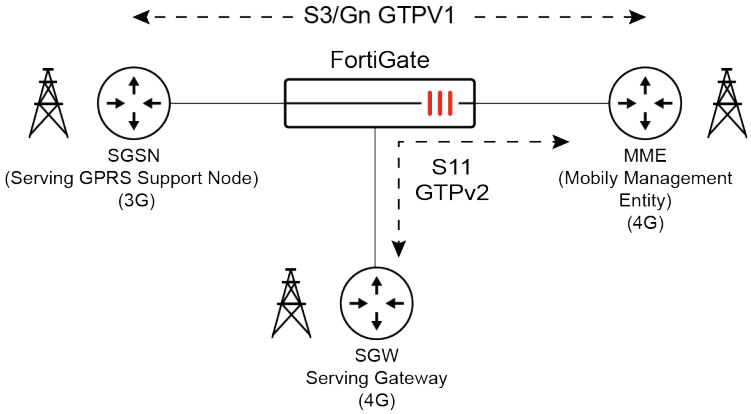
```
diagnose firewall gtp tunnel list
list gtp tunnels

-----------prof=gtpp ref=6 imsi=240011334567001 msisdn=140541334560001 mei=35911300.000001.0 ms_
addr=192.168.4.2 s11_s4 1-----------
-----------index=0000001d life=1340(sec) idle=24(sec) vd=0  ver=2-----------
c_pkt=4 c_bytes=696 u_pkt=0 u_bytes=0
rat type: ltem-geo
downlink cfteid:
    addr=9.8.3.19 teid=0x00000002 role=control vd=0 intf_type=s11 mme gtp-c
uplink cfteid:
    addr=9.8.8.19 teid=0x00000002 role=control vd=0 intf_type=s11/s4 sgw gtp-c
1/1 bearers:
    id=5 linked_id=0 type=regular dead=0 apn=internet selection=ms-or-net-provided-apn apn_
restriction=private-1 user_addr=192.168.4.2 u_pkt=0 u_bytes=0
    2 fteids:
        addr=9.8.8.19 teid=0x00000003 role=data vd=0 intf_type=s1-u sgw gtp-u
        addr=9.8.8.19 teid=0x00000002 role=data vd=0 intf_type=s5/s8 pgw gtp-c
```

3G to 4G/5G handover support

FortiOS Carrier now understands the following GTPv1 mobility management (MM) messages on S3/Gn interfaces that are used to transition a UE from 3G to 4G/5G; including support for smooth inter-RAT handovers.

- Forward Relocation Request (53)
- Forward Relocation Response (54)
- Forward Relocation Complete Notification (135)
- Forward Relocation Complete Acknowledge (136)
- SGSN Context Request (50)
- SGSN Context Response (51)
- SGSN Context Acknowledge (52)

Example topology



When a 3G UE migrates to 4G/5G, GTPv1 signaling messages are exchanged over the S3/Gn interface between the Serving GPRS Support Node (SGSN)(3G) and the Mobility Management Entity (MME)(4G) through the FortiGate.

These messages are required to establish a GTPv2 tunnel in the 4G network. FortiOS Carrier recognizes the GTPv1 signaling, establishes the tunnel, and maps subsequent GTPv2 messages to it, updating the version to GTPv2.

TPv1 SGSN Context Request/Response/Acknowledgment messages received

The output of the `diagnose firewall gtp tunnel list` command shows the GTPv1 tunnel has been established after receiving the SGSN context messages.

```
diagnose firewall gtp tunnel list
list gtp tunnels

-----------prof=gtp685663 ref=5 imsi=206012238331481 msisdn=unknown mei=unknown ms_addr=Unknown
s11_s4 0-----------
-----------index=00000002 life=17(sec) idle=17(sec) vd=1  ver=1-----------
c_pkt=1 c_bytes=55 u_pkt=0 u_bytes=0
downlink cfteid:
    addr=Unknown teid=0x00000000 role=control vd=0 intf_type=s1-u enodeb gtp-u
uplink cfteid:
    addr=213.181.60.88 teid=0x0cca317a role=control vd=1 intf_type=gn-gp ggsn gtp-c
1/1 bearers:
    id=5 linked_id=0 type=regular dead=0 apn=internet.proximus.be.mnc001.mcc206.gprs user_
addr=100.78.125.49/2a02:a020:92:4e06:1:0:a1b4:7bcb u_pkt=0 u_bytes=0
    1 fteids:
        addr=213.181.60.88 teid=0x0cc8317a role=data vd=1 intf_type=gn-gp ggsn gtp-u
```

GTPv2 Modify Bearer messages traverse the FortiGate

When GTPv2 Modify Bearer messages traverse the FortiGate, they are mapped to the existing GTPv1 tunnel managed by the FortiGate, updating the tunnel version to GTPv2:

```
 diagnose firewall gtp tunnel list
list gtp tunnels
```

```
-----------prof=gtp685663 ref=6 imsi=206012238331481 msisdn=unknown mei=35299271.388816.16 ms_
addr=Unknown s11_s4 0-----------
-----------index=00000002 life=93(sec) idle=7(sec) vd=1  ver=2-----------
rat type: eutran
downlink cfteid:
    addr=101.119.242.123 teid=0xb8e0ded0 role=control vd=1 intf_type=s5/s8 sgw gtp-c
uplink cfteid:
    addr=213.181.60.88 teid=0x0cca317a role=control vd=1 intf_type=s5/s8 pgw gtp-c
1/1 bearers:
    id=5 linked_id=5 type=regular dead=0 apn=internet.proximus.be.mnc001.mcc206.gprs apn_
restriction=all user_addr=100.78.125.49/2a02:a020:92:4e06:1:0:a1b4:7bcb u_pkt=0 u_bytes=0
    2 fteids:
        addr=213.181.60.88 teid=0x0cc8317a role=data vd=1 intf_type=s5/s8 pgw gtp-u
        addr=101.119.242.122 teid=0xf1c00b02 role=data vd=1 intf_type=s5/s8 sgw gtp-u
```

Increase GTP tunnel limit per profile to 50M tunnels

The maximum tunnel limit GTP profiles and global GTP settings has been increased from 16,000,000 to 50,000,000.

FortiOS Carrier includes two methods for limiting the number of GTP tunnels that can be operating at one time.

- You can add a tunnel limit to a GTP profile. All traffic processed by that GTP profile cannot open more tunnels than this configured tunnel limit.
- You can create global tunnel limits and add these tunnel limits to GTP profiles. This global tunnel limit applies to all traffic processed by all of the GTP profiles configured with that same global tunnel limit.

In network configurations, that for example support IoT deployments, GTP tunnel limiting can be scaled up to allow up to 50,000,000 GTP tunnels.

GTP profile tunnel limit

Use the following command to set the tunnel limit for a GTP profile:

```
config firewall gtp
   edit <name>
      set tunnel-limit <number-of-tunnels>
   end
```

On the GUI: **General Settings > Tunnel limit**.

Enter the maximum number of tunnels allowed to be open at one time. The range is 1 to 50,000,000 tunnels. The default tunnel limit is 0, which means no limit.

Global GTP tunnel limits

Use the following command to create global shared tunnel limits:

```
config gtp tunnel-limit
   edit <global-tunnel-limit-name>
      set tunnel-limit <number-of-tunnels>
   end
```

Enter the maximum number of global tunnels allowed to be open at one time. The range is 1 to 50,000,000 tunnels. The default limit is 0, which means no limit.

You can add multiple global tunnel limits. Global tunnel limits are added to a VDOM and can be applied to traffic processed by that VDOM

Use the following command to add a shared tunnel limit to a GTP profile:

```
config firewall gtp
   edit <profile-name>
      set global-tunnel-limit <global-tunnel-limit-name>
   end
```

Viewing information about GTP tunnel limits

You can use the following diagnose command to view information about global tunnel limits.

```
diagnose firewall gtp tunnel-limit list
```

The command output includes a line for each global tunnel limit. The output includes the name of the tunnel limit, its configured limit, and the number of tunnels processed by GTP profiles that the global tunnel limit has been added to. For example:

```
name=gtp-tl-1 tunnel_limit=7000 tunnel_count=500
```

You can also use the following command to get another view of GTP tunnel configuration information:

```
diagnose firewall gtp stat
request=0 echo_request=0 tunnel=50000000 tunnel_v0=0 path=222144 bearer=50000000 fteid=100000000
ds_fteid=0 profile=1
mobility=0 pdn_connection=0
imsi=0 apn=0 apn_shaper=0 tunnel_limiter=0 adv_policy=0 ie_remove_policy=0 ip_policy=0
noip_policy=0 ie_wl_entry=0 clash=0 drop=0 apn_hash_node=0
Based on 3GPP TS 29.060 V15.5.0 & 3GPP TS 29.274 V15.9.0
```

FortiASIC

This section includes information about FortiASIC related new features:

- Hardware acceleration on page 579
- Hyperscale firewall on page 582

Hardware acceleration

This section includes information about hardware acceleration related new features:

- Improvements for NP7-offloaded GRE tunnels on page 579
- New SNMP counters to support CGNAT IP pool monitoring
- NP7 support for traffic shaping based on dynamic RADIUS VSAs

See also Hardware Acceleration.

Improvements for NP7-offloaded GRE tunnels

Two new features are available to improve performance and add features for NP7-offloaded GRE tunnels.

L3/L4 header-based hashing for NP7-offloaded GRE tunnels

You can use the following command to configure the algorithm that controls how NP7-offloaded GRE tunnel sessions are load balanced by LAGs. Depending on your traffic profile, you can adjust the algorithm to optimize performance and throughput. The command allows you to use L3 and L4 GRE headers when calculating load balancing hashing.

```
config system npu
    set lag-hash-gre {disable | gre_inner_l3 | gre_inner_l4 | gre_inner_l3l4}
end
```

`disable` (the default) disable using GRE inner headers as hash input.

`gre_inner_l3` use GRE inner L3 header information as hash input.

`gre_inner_l4` use GRE inner L4 header information as hash input.

`gre_inner_l3l4` use GRE inner L3 and L4 header information as hash input.

Once you have configured the `lag-hash-gre` option, you can configure a LAG to use the configuring hashing, by setting the `algorithm` option of the LAG configuration to `NPU-GRE`:

```
config system interface
    edit example-lag
        set type aggregate
        set algorithm NPU-GRE
    end
```

`NPU-GRE` use L4 and GRE inner header information for calculating load balance hashing for GRE tunnels passing through this LAG. The `NPU-GRE` option is only available if you have configured `lag-hash-gre`.

ECMP hashing of NP7-offloaded GRE tunnels

When configuring a GRE tunnel, you can use the `loopback-ecmp-offload` option to support ECMP hashing for offloaded GRE tunnels over software loopback interfaces.

```
config system gre-tunnel
    edit <tunnel-name>
        set loopback-ecmp-offload {disable | enable}
    end
```

`loopback-ecmp-offload` enable or disable ECMP hashing (or multi-tunnel support) for offloaded GRE tunnels.

You cannot change to the `loopback-ecmp-offload` setting after creating the GRE tunnel.

Example configuration

Example loopback interface configuration:

```
config system interface
    edit loopb1
        set vdom vdom1
        set ip 10.10.10.1 255.255.255.255
        set allowaccess ping https ssh snmp http telnet fgfm
        set type loopback
        set snmp-index 57
    end
```

Example GRE tunnel configuration

```
config system gre-tunnel
    edit "gre1"
        set interface "lb01"
        set remote-gw 10.10.10.2
        set local-gw 10.10.10.1
        set loopback-ecmp-offload enable
        set keepalive-interval 1
    end
```

NP7 offloading of TCP and UDP sessions denied by firewall policies to reduce CPU usage

If your FortiGate with NP7 processors is experiencing high CPU usage because the CPU is processing many denied sessions , you can use the following command to offload those denied sessions to NP7 processors and reduce CPU usage:

```
config system npu
   set session-denied-offload enable
end
```

To test the feature:

1. Set action to deny in the policy:

```
config firewall policy
    edit 1
        set action deny
        set srcintf "lan1"
        set dstintf "lan2"
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set logtraffic disable
    next
end
```

2. Generate UDP traffic and check session info:

```
session info: proto=17 proto_state=00 duration=3 expire=27 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=block may_dirty npu f15
statistic(bytes/packets/allow_err): org=5312/4/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=9->10/10->9 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 172.16.200.55:49877->10.1.100.11:5001(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.11:5001->172.16.200.55:49877(0.0.0.0:0)
misc=0 policy_id=1 pol_uuid_idx=1902 auth_info=0 chk_client_info=0 vd=0
serial=00002ec4 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x000400 ofld-O
npu info: flag=0x81/0x00, offload=9/0, ips_offload=0/0, epid=4077/0, ipid=130/0,
vlan=0x0000/0x0000
vlifid=130/0, vtag_in=0x0000/0x0000 in_npu=1/0, out_npu=1/0, fwd_en=0/0, qid=19/0, ha_
divert=0/0
```

```
no_ofld_reason:
hrx info: valid=1/0, qid=20/0, npuid=1/0, sublink=0/0
```

NP7 support for traffic shaping based on dynamic RADIUS VSAs

Dynamic shaping profiles are supported for traffic offloaded by NP7 or NP7Lite (SoC5) processors. This feature allows traffic control policies to be applied per user based on authentication details and bandwidth parameters from the RADIUS server. For more information, see Traffic shaping based on dynamic RADIUS VSAs.

Hyperscale firewall

This section includes information about hyperscale firewall related new features:

- New SNMP counters to support CGNAT IP pool monitoring on page 582
- Hyperscale HA hardware session synchronization improves support for PBR sessions on page 586
- Hyperscale CGNAT EIF session timer options on page 587

See also Hyperscale Firewall Guide.

New SNMP counters to support CGNAT IP pool monitoring

The following new hyperscale firewall CGNAT IP pool SNMP fields have been added:

- .fgFwIppStatsFreePBAs 1.3.6.1.4.1.12356.101.5.3.2.1.1.12 number of free PBAs in ippool list.
- .fgFwIppStatsInusePBAs 1.3.6.1.4.1.12356.101.5.3.2.1.1.10 number of in-use PBAs in ippool list.
- .fgFwIppStatsTotalPBAs 1.3.6.1.4.1.12356.101.5.3.2.1.1.9 number of PBAs in ippool list.
- .fgFwIppStatsInuseIPs 1.3.6.1.4.1.12356.101.5.3.2.1.1.27 number of in-use IPs in ippool.
- .fgFwIppStatsFreeIPs 1.3.6.1.4.1.12356.101.5.3.2.1.1.26 number of free IPs in ippool.

Example configuration and SNMP results

This example includes a basic CGNAT IPpool and hyperscale firewall configuration. Then shows results of running CGNAT traffic using the diagnose firewall ippool list command output and SNMP results for the new SNMP fields.

IP Pool configuration

```
config firewall ippool
    edit test-cgn-ippool64-1
        set type cgn-resource-allocation
        set startip 172.16.201.0
        set endip 172.16.201.3
```

```
        set nat64 enable
        set cgn-block-size 64
        set cgn-port-start 4117
    next
    edit test-cgn-pba-1
        set type cgn-resource-allocation
        set startip 172.16.201.181
        set endip 172.16.201.182
        set cgn-block-size 256
        set cgn-port-start 4117
    next
    edit test-cgn-opba-1
        set type cgn-resource-allocation
        set startip 172.16.201.183
        set endip 172.16.201.184
        set cgn-overload enable
        set cgn-block-size 256
        set cgn-port-start 4117
    next
    edit test-cgn-spba-1
        set type cgn-resource-allocation
        set startip 172.16.201.185
        set endip 172.16.201.186
        set cgn-spa enable
        set cgn-port-start 4117
    next
    edit test-cgn-ospba-1
        set type cgn-resource-allocation
        set startip 172.16.201.187
        set endip 172.16.201.188
        set cgn-spa enable
        set cgn-overload enable
        set cgn-port-start 4117
    next
    edit test-cgn-fpba-1
        set type cgn-resource-allocation
        set startip 172.16.201.191
        set endip 172.16.201.194
        set cgn-fixedalloc enable
        set cgn-client-startip 10.1.100.11
        set cgn-client-endip 10.1.100.22
        set cgn-port-start 4117
    next
    edit test-cgn-pba-11
        set type cgn-resource-allocation
        set startip 172.16.202.181
        set endip 172.16.202.182
        set cgn-block-size 512
        set cgn-port-start 4117
    next
    edit test-cgn-pba-22
        set type cgn-resource-allocation
```

```
        set startip 172.16.202.183
        set endip 172.16.202.184
        set cgn-block-size 64
        set cgn-port-start 1024
    next
    end
```

Hyperscale firewall policy

```
config firewall policy
        edit 1
            set name cgn-hw1-policy4-1
            set srcintf port2
            set dstintf port1
            set action accept
            set srcaddr all
            set dstaddr all
            set service ALL
            set nat enable
            set cgn-log-server-grp test-syslog-svrgrp-1
            set ippool enable
            set poolname test-cgn-pba-1
            set traffic-shaper shaper10M-high
            set traffic-shaper-reverse shaper10M-high
        end
```

Output of the diagnose firewall ippool list command

```
diagnose firewall ippool list
list ippool info:(vf=cgn-hw1)
ippool test-cgn-ippool64-1: id=1, block-sz=64
        ip-range=172.16.201.0-172.16.201.3

ippool test-cgn-pba-1: id=2, block-sz=256
        ip-range=172.16.201.181-172.16.201.182
        num-pba-per-ip=239
        grp=N/A, start-port=4117, end-port=65530
        npu-clients=1, npu-inuse-NAT-IPs=1, total-NAT-IP=2
        npu-total-PBAs=478, npu-inuse-PBAs=4/0, npu-free-PBAs=99.16%/100.00%
        npu-tcp-sess-count=1021, npu-udp-sess-count=0
ippool test-cgn-opba-1: id=3, block-sz=256
        ip-range=172.16.201.183-172.16.201.184

ippool test-cgn-spba-1: id=4,
        ip-range=172.16.201.185-172.16.201.186

ippool test-cgn-ospba-1: id=5,
        ip-range=172.16.201.187-172.16.201.188

ippool test-cgn-fpba-1: id=6,
        ip-range=172.16.201.191-172.16.201.194
```

```
ippool test-cgn-pba-11: id=7, block-sz=512
        ip-range=172.16.202.181-172.16.202.182

ippool test-cgn-pba-22: id=8, block-sz=64
        ip-range=172.16.202.183-172.16.202.184
```

SNMP OID trap results

fgFwIppStatsFreePBAs: number of free PBAs in ippool list

The second line of the SNMP output shows that 474 of the 478 PBAs are free (not in use).

```
snmpwalk -v2c -c hs_public 172.16.200.27 1.3.6.1.4.1.12356.101.5.3.2.1.1.12

FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.12.172.16.201.0.172.16.201.3.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.12.172.16.201.181.172.16.201.182.2 = Gauge32: 474
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.12.172.16.201.183.172.16.201.184.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.12.172.16.201.185.172.16.201.186.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.12.172.16.201.187.172.16.201.188.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.12.172.16.201.191.172.16.201.194.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.12.172.16.202.181.172.16.202.182.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.12.172.16.202.183.172.16.202.184.2 = Gauge32: 0
```

fgFwIppStatsInusePBAs: number of in-use PBAs in ippool list

The second line of the SNMP output shows that 4 of the 478 PBAs are currently being used.

```
snmpwalk -v2c -c hs_public 172.16.200.27 1.3.6.1.4.1.12356.101.5.3.2.1.1.10

FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.10.172.16.201.0.172.16.201.3.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.10.172.16.201.181.172.16.201.182.2 = Gauge32: 4
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.10.172.16.201.183.172.16.201.184.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.10.172.16.201.185.172.16.201.186.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.10.172.16.201.187.172.16.201.188.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.10.172.16.201.191.172.16.201.194.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.10.172.16.202.181.172.16.202.182.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.10.172.16.202.183.172.16.202.184.2 = Gauge32: 0
```

fgFwIppStatsTotalPBAs: toal number of PBAs in ippool list

The second line of the SNMP output shows that there are a total of 478 PBAs.

```
snmpwalk -v2c -c hs_public 172.16.200.27 1.3.6.1.4.1.12356.101.5.3.2.1.1.9

FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.9.172.16.201.0.172.16.201.3.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.9.172.16.201.181.172.16.201.182.2 = Gauge32: 478
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.9.172.16.201.183.172.16.201.184.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.9.172.16.201.185.172.16.201.186.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.9.172.16.201.187.172.16.201.188.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.9.172.16.201.191.172.16.201.194.2 = Gauge32: 0
```

```
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.9.172.16.202.181.172.16.202.182.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.9.172.16.202.183.172.16.202.184.2 = Gauge32: 0
```

fgFwIppStatsInuseIPs: number of in-use IPs in ippool

The SNMP output shows that only the IP pool on the second output line is currently being used, and only one of the IPs in that IP pool is being used.

```
snmpwalk -v2c -c hs_public 172.16.200.27 1.3.6.1.4.1.12356.101.5.3.2.1.1.27

FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.27.172.16.201.0.172.16.201.3.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.27.172.16.201.181.172.16.201.182.2 = Gauge32: 1
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.27.172.16.201.183.172.16.201.184.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.27.172.16.201.185.172.16.201.186.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.27.172.16.201.187.172.16.201.188.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.27.172.16.201.191.172.16.201.194.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.27.172.16.202.181.172.16.202.182.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.27.172.16.202.183.172.16.202.184.2 = Gauge32: 0
```

fgFwIppStatsFreeIPs: number of free IPs in ippool

The SNMP output shows the number of free IPs in each IP pool.

```
snmpwalk -v2c -c hs_public 172.16.200.27 1.3.6.1.4.1.12356.101.5.3.2.1.1.26

FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.26.172.16.201.0.172.16.201.3.2 = Gauge32: 4
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.26.172.16.201.181.172.16.201.182.2 = Gauge32: 2
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.26.172.16.201.183.172.16.201.184.2 = Gauge32: 2
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.26.172.16.201.185.172.16.201.186.2 = Gauge32: 2
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.26.172.16.201.187.172.16.201.188.2 = Gauge32: 2
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.26.172.16.201.191.172.16.201.194.2 = Gauge32: 4
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.26.172.16.202.181.172.16.202.182.2 = Gauge32: 2
FORTINET-FORTIGATE-MIB::fgFwIppStatsEntry.26.172.16.202.183.172.16.202.184.2 = Gauge32: 2
```

Hyperscale HA hardware session synchronization improves support for PBR sessions

Hyperscale HA hardware session synchronization improves support for policy based routing (PBR) session failover. To achieve this improvement, HA hardware session synchronization adds PBR-corrected reverse-path information to hardware synchronized sessions.

Previously, PBR hardware synchronized sessions were not corrected. Because of this, if your hyperscale FortiGate used policy routing, after a failover, HA hardware sessions would have to re-learn PBR reverse-path information. This re-learning would delay failover for PBR sessions.

Hyperscale CGNAT EIF session timer options

On FortiGates licensed for Hyperscale firewall, the following new options are available to improve control over timers related to EIF sessions.

```
config system npu
    set eif-tcp-refresh-dir {both | outgoing | incoming}
    set eif-udp-refresh-dir {both | outgoing | incoming}
    set eif-tcp-ttl <time>
    set eif-udp-ttl <time>
    set extra-timeout-tcp <time>
    set extra-timeout-udp <time>
end
```

`eif-tcp-refresh-dir` the SSE timeout TCP refresh direction for EIF sessions.

`eif-udp-refresh-dir` the SSE timeout TCP refresh direction for EIF sessions.

`both` refresh the timer in both directions, this is the default option.

`outgoing` refresh the outgoing timer, the timer setup with the session is first started.

`incoming` refresh the incoming timer.

`eif-tcp-ttl` SSE timeout TCP TTL for EIF sessions in seconds. The range can be 300 to 72000 seconds. The default timeout is 3600 seconds.

`eif-udp-ttl` SSE timeout TCP TTL for EIF sessions in seconds. The range can be 300 to 72000 seconds. The default timeout is 180 seconds.

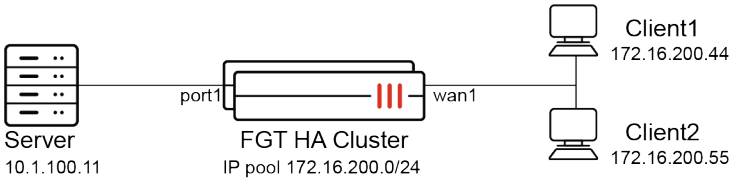
`extra-timeout-tcp` extra timeout for TCP with `eif-tcp-refresh-dir` set to `incoming` or `both` and the `scan-stale` option of the `config background-sse-scan` command is set to 1. The range can be 0 to 7200 sec, default 0 sec. You can use this option to close an incoming TCP EIF session that has been open longer than intended.

`extra-timeout-udp` extra timeout for UDP with `eif-udp-refresh-dir` set to `incoming` or `both` and the `scan-stale` option of the `config background-sse-scan` command is set to 1. The range can be 0 to 7200 sec, default 0 sec. You can use this option to close an incoming UDP EIF session that has been open longer than intended.

Example configurations

The following example configurations use this example topology:

Example topology



Example 1: setting the EIF refresh timer and timeout for outgoing EIF sessions

Set NP7 processors to refresh and set a timeout for outgoing EIF sessions.

```
config system npu
    set eif-tcp-refresh-dir outgoing
    set eif-udp-refresh-dir outgoing
    set eif-tcp-ttl 600
    set eif-udp-ttl 300
end
```

Create a firewall policy with EIF enabled:

```
config firewall policy
    edit 1
        set name cgn-hw1-policy4-1
        set srcintf port1
        set dstintf wan1
        set action accept
        set srcaddr all
        set dstaddr all
        set service ALL
        set nat enable
        set cgn-eif enable
        set cgn-log-server-grp test-syslog-svrgrp-1
        set ippool enable
        set poolname test-cgn-pba-1
    next
end
```

Generate a SNAT UDP session from the server to Client1. The following session appears on the FortiGate:

```
session info: proto=6 proto_state=11 duration=19 expire=3580 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=0
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log hw_ses log-start
statistic(bytes/packets/allow_err): org=112/2/0 reply=60/1/0 tuples=2
tx speed(Bps/kbps): 5/0 rx speed(Bps/kbps): 3/0
orgin->sink: org pre->post, reply pre->post dev=70->71/71->70 gwy=172.16.200.44/10.1.100.11
hook=post dir=org act=snat 10.1.100.11:4155->172.16.200.44:80(172.16.201.181:34325)
hook=pre dir=reply act=dnat 172.16.200.44:80->172.16.201.181:34325(10.1.100.11:4155)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=20028 auth_info=0 chk_client_info=0 vd=500
serial=7c845805 tos=00/00 app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
  setup by offloaded-policy: origin=native
  O: npid=2/0, in: OID=215/VID=20, out: NHI=23028 OID=215/VID=30
  R: npid=0/2, in: OID=215/VID=30, out: NHI=17110 OID=215/VID=20
# hardware-session = 1
```

Generate a TCP EIF session from Client 1:

- Verify that the `eif-tcp-refresh-dir and eif-tcp-ttl` settings apply to TCP EIF sessions. This example uses host logging, so you need to use the `diagnose sys npu-session list-full` command. The session

should show `timeout=600` and `refresh_dir=org`.

- Verify the org packet will refresh the session's expire time.
- Verify TCP EIF sessions will expire after timeout (600 seconds) has been reached.

```
diagnose sys npu-session list-full

session info: proto=6 proto_state=11 duration=36 expire=563 timeout=600 refresh_dir=org
flags=00000000 socktype=0 sockport=0 av_idx=0 use=0
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log hw_ses log-start
statistic(bytes/packets/allow_err): org=112/2/0 reply=60/1/0 tuples=2
tx speed(Bps/kbps): 3/0 rx speed(Bps/kbps): 1/0
orgin->sink: org pre->post, reply pre->post dev=71->70/70->71 gwy=10.1.100.11/172.16.200.55
hook=pre dir=org act=dnat 172.16.200.55:12346->172.16.201.181:34325(10.1.100.11:4155)
hook=post dir=reply act=snat 10.1.100.11:4155->172.16.200.55:12346(172.16.201.181:34325)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=20028 auth_info=0 chk_client_info=0 vd=500
serial=55e72604 tos=00/00 app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
  setup by offloaded-policy: origin=native
  O: npid=1/2, in: OID=215/VID=30, out: NHI=17110 OID=215/VID=20
  R: npid=2/1, in: OID=215/VID=20, out: NHI=23028 OID=215/VID=30
# hardware-session = 1
```

After the timeout, the `diagnose sys npu-session list-full` command shows no hardware sessions:

```
diagnose sys npu-session list-full
# hardware-session = 0
```

Generate UDP EIF sessions from Client 1:

- Verify that the `eif-udp-refresh-dir and eif-udp-ttl` settings apply to UDP EIF sessions. This example uses host logging, so you need to use the `diagnose sys npu-session list-full` command. The session should show `timeout=300` amd `refresh_dir=org`.
- Verify the org packets will refresh the session's expire time.
- Verify UDP EIF sessions will expire after timeout (300 seconds) has been reached.

```
diagnose sys npu-session list-full

session info: proto=17 proto_state=00 duration=123 expire=176 timeout=300 refresh_dir=org
flags=00000000 socktype=0 sockport=0 av_idx=0 use=0
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log hw_ses log-start
statistic(bytes/packets/allow_err): org=28/1/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=71->70/70->71 gwy=10.1.100.11/172.16.200.55
hook=pre dir=org act=dnat 172.16.200.55:4117->172.16.201.181:34325(10.1.100.11:4155)
```

```
hook=post dir=reply act=snat 10.1.100.11:4155->172.16.200.55:4117(172.16.201.181:34325)
misc=0 policy_id=1 pol_uuid_idx=20028 auth_info=0 chk_client_info=0 vd=500
serial=fcea3dc7 tos=00/00 app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
  setup by offloaded-policy: origin=native
  O: npid=1/2, in: OID=215/VID=30, out: NHI=17110 OID=215/VID=20
  R: npid=2/1, in: OID=215/VID=20, out: NHI=23028 OID=215/VID=30
```

Example 2: using the extra-timeout options

With the following configuration, when `eif-udp-refresh-dir` is set to `incoming`, and when you set a timeout using the `extra-timeout-udp` option, even when incoming traffic is received, if there isn't any corresponding outgoing traffic the EIF session is deleted. This example uses UDP traffic, but would work the same for TCP traffic.

```
config system npu
    config background-sse-scan
        set scan-stale 1
    end
```

```
config system npu
    set eif-udp-refresh-dir incoming
    set eif-udp-ttl 100
    set extra-timeout-udp 30
end
```

Create a firewall policy with EIF enabled:

```
config firewall policy
    edit 1
        set name cgn-hw1-policy4-1
        set srcintf port1
        set dstintf wan1
        set action accept
        set srcaddr all
        set dstaddr all
        set service ALL
        set nat enable
        set cgn-eif enable
        set cgn-log-server-grp test-syslog-svrgrp-1
        set ippool enable
        set poolname test-cgn-pba-1
    next
end
```

Generate a SNAT UDP session from the server to Client1. Then send EIF traffic from Client 1 to the server. After a time interval typically slightly longer than the `eif-udp-ttl` time of 100 plus `extra-timeout-udp` time of 30 seconds the EIF session should be deleted and client 1 can no longer connect to the server.

```
diagnose sys npu-session list

session info: proto=17 proto_state=00 duration=195 expire=91 timeout=100 refresh_dir=reply
flags=00000000 socktype=0 sockport=0 av_idx=0 use=0
```

```
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log hw_ses log-start
statistic(bytes/packets/allow_err): org=28/1/0 reply=1008/36/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 5/0
orgin->sink: org pre->post, reply pre->post dev=71->70/70->71 gwy=10.1.100.11/172.16.200.55
hook=pre dir=org act=dnat 172.16.200.55:5155->172.16.201.181:34325(10.1.100.11:1155)
hook=post dir=reply act=snat 10.1.100.11:1155->172.16.200.55:5155(172.16.201.181:34325)
misc=0 policy_id=1 pol_uuid_idx=20029 auth_info=0 chk_client_info=0 vd=500
serial=099654e7 tos=00/00 app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
  setup by offloaded-policy: origin=native
  O: npid=1/2, in: OID=217/VID=30, out: NHI=17110 OID=217/VID=20
  R: npid=2/1, in: OID=217/VID=20, out: NHI=23028 OID=217/VID=30
# hardware-session = 1
```

Operational Technology

This section includes information about Operational Technology related new features:

- System on page 592

System

This section includes information about system related Operational Technology new features:

- MACsec support for FortiGate Rugged Models with hardware switch on page 592
- FortiGate Rugged digital I/O trigger logging enhancement on page 593

MACsec support for FortiGate Rugged Models with hardware switch

MACsec support is added across all applicable FortiGate Rugged (FGR) models with hardware switch. This enables secure Layer-2 encryption and meets mandatory compliance requirements for regulated OT environments where MACsec is required.

1. Enable MACsec:

```
# diagnose debug macsec set wan1 0x1234 0x4321 1 2
```

2. Examine the configured MACsec:

```
# diagnose debug macsec get wan1
Interface                wan1
MACsec enabled           1
macsec->encrypt_key      0x00000000000000000000000000001234
macsec->decrypt_key      0x00000000000000000000000000004321
macsec->sci              0x0000000000000001
macsec->ingress_sci      0x0000000000000002
```

3. Disable MACsec:

```
# diagnose debug macsec clear wan1
```

4. Verify that MACsec is disabled:

```
# diagnose debug macsec get wan1
Interface              wan1
MACsec enabled         0
```

```
macsec->encrypt_key      0x00000000000000000000000000000000
macsec->decrypt_key      0x00000000000000000000000000000000
macsec->sci              0x0000000000000000
macsec->ingress_sci      0x0000000000000000
```

FortiGate Rugged digital I/O trigger logging enhancement

FortiOS now includes event log triggers for status changes in the GPIO (general purpose input output) module on FortiGate Rugged (FGR) when digital I/O triggers are activated. See FGR-70F/FGR-70F-3G4G GPIO/DIO module for more information about the module. The event log triggers can be used for automation stitches. This enhancement allows FGR to execute user-defined stitch actions, such as enabling or disabling specific policies or VPNs, based on input from an external switch or button.

FortiOS now includes the following event log triggers for FGR:

```
show system automation-trigger GPIO ?
name Name.
GPIO Input1 Off event-based event-log forticare-support posture
GPIO Input1 On event-based event-log forticare-support posture
GPIO Input2 Off event-based event-log forticare-support posture
GPIO Input2 On event-based event-log forticare-support posture
```

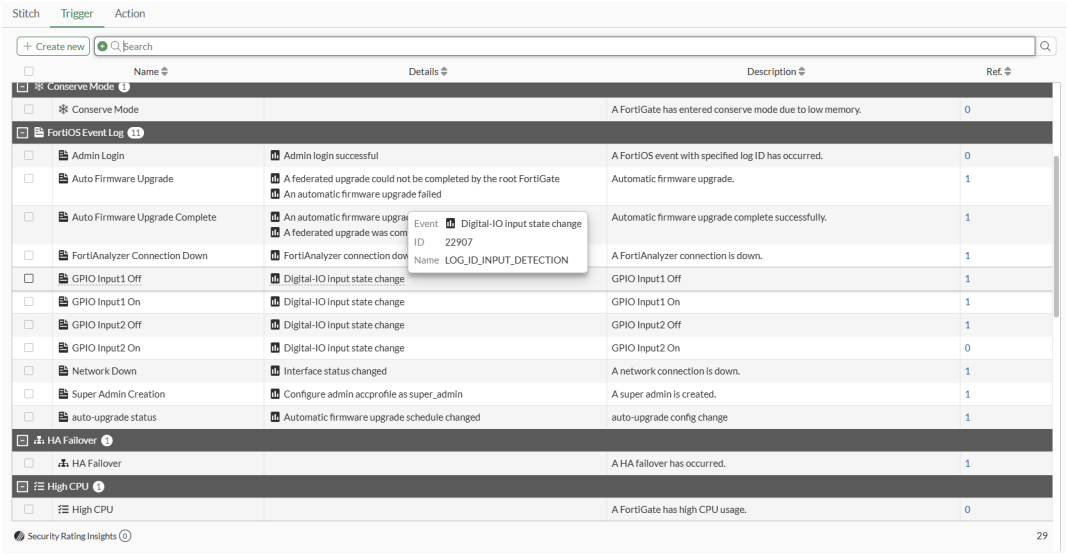
Example

This example describes how to use the GUI and CLI to create an automation stitch with the following actions for a GPIO trigger:

- Disable firewall policy 1
- Email a notification

To configure automation stitches in the GUI:

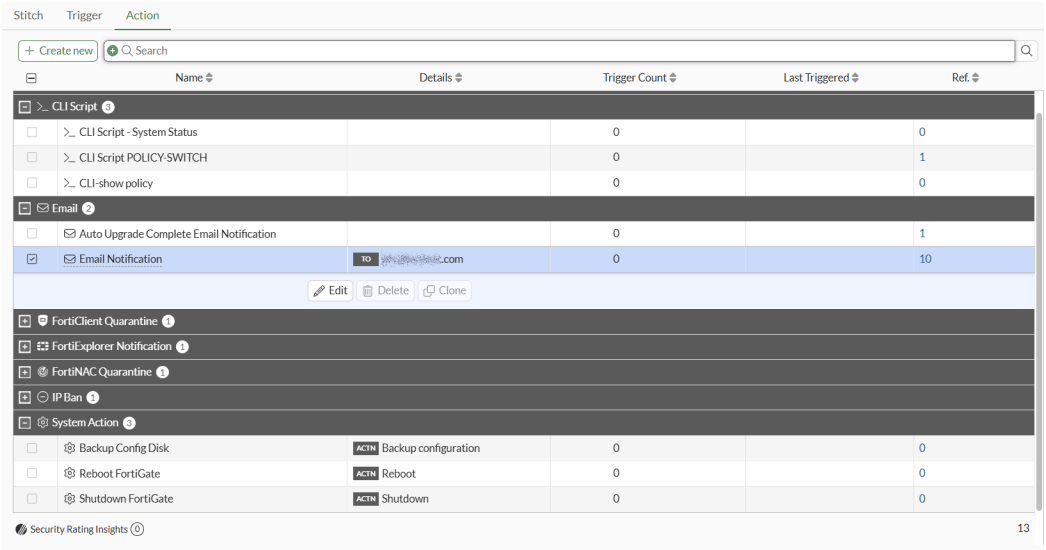
1. Go to *Security Fabric > Automation*.
2. View the available triggers:
   a. Click the *Trigger* tab.
   b. Expand *FortiOS Event Log* and view the following triggers: *GPIO Input1 Off*, *GPIO Input2 On*, *GPIO Input2 Off*, and *GPIO Input2 On*.



3. Configure actions:

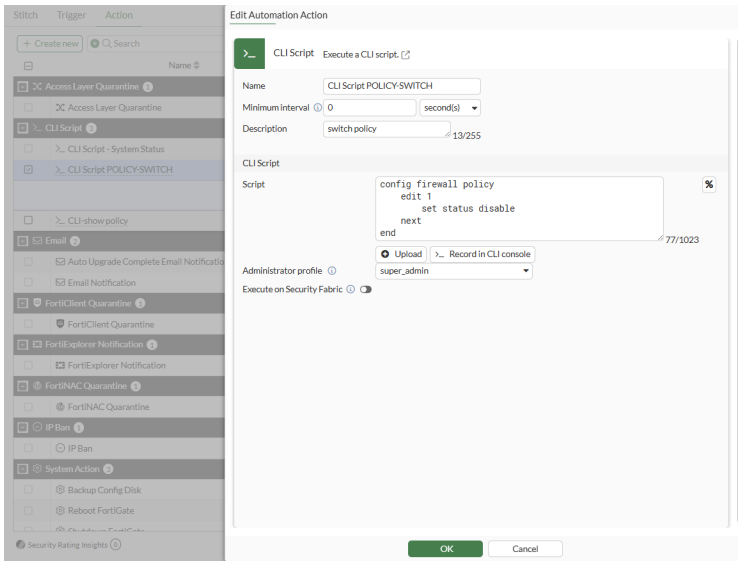
   This example describes how to add the following actions: email notification and CLI script.

   a. Click the *Actions* tab.
   b. Click *Create new*.
   c. Select *Email*, define the options, and click *OK*. The action is displayed.



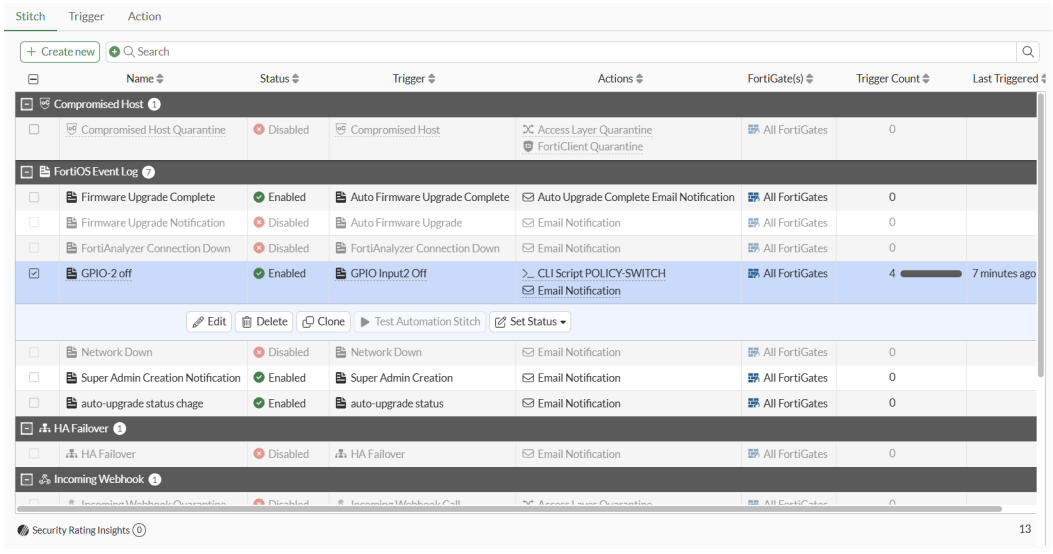
   d. Click *Create new*.
   e. Select *CLI Script*, define the options, and click *OK*.

      In this example, the automation stitch disables the firewall policy.



The action is displayed.

4. Configure a stitch and select the actions you created:
   a. Click the *Stitch* tab.
   b. Click *Create new*.
   c. Enter a *Name*, such as *GPIO-2 off*.
   d. Click *Add Trigger*, select a GPIO trigger, and click *Apply*.
      This example uses the *GPIO Input2 Off* trigger.
   e. Click *Action*, select the email action you created, and click *OK*.
   f. Click *Action*, select the CLI script you created, and click *OK*.
   g. Click *OK* to save the stitch.



To configure automation stitches in the CLI:

1. View the GPIO trigger settings:
   - Input1 on:

```
config system automation-trigger
    edit "GPIO Input1 On"
        set description "GPIO Input1 On"
        set event-type event-log
        set logid 22907
        config fields
            edit 1
                set name "connector"
                set value "input_IN1_REF"
            next
            edit 2
                set name "state"
                set value "open"
            next
        end
    next
end
```

   - Input1 off:

```
config system automation-trigger
    edit "GPIO Input1 Off"
        set description "GPIO Input1 Off"
        set event-type event-log
        set logid 22907
        config fields
            edit 1
                set name "connector"
                set value "input_IN1_REF"
            next
            edit 2
                set name "state"
                set value "closed"
            next
        end
    next
end
```

   - Input2 off:

```
config system automation-trigger
    edit "GPIO Input2 Off"
        set description "GPIO Input2 Off"
        set event-type event-log
        set logid 22907
        config fields
            edit 1
```

```
                set name "connector"
                set value "input_IN2_REF"
            next
            edit 2
                set name "state"
                set value "closed"
            next
        end
    next
end
```

- Input2 on:

```
config system automation-trigger
    edit "GPIO Input2 On"
        set description "GPIO Input2 Off"
        set event-type event-log
        set logid 22907
        config fields
            edit 1
                set name "connector"
                set value "input_IN2_REF"
            next
            edit 2
                set name "state"
                set value "open"
            next
        end
    next
end
```

2. Configure a stitch for a GPIO event:
   The automation stitch `GPIO-2 off` is configured to use the following actions: CLI script and email notification.

```
config system automation-stitch
    edit "GPIO-2 off"
        set trigger "GPIO Input2 Off"
        config actions
            edit 1
                set action "CLI Script POLICY-SWITCH"
                set required enable
            next
            edit 2
                set action "Email Notification"
                set required enable
            next
        end
    next
end
```

3. Configure the actions for the stitch:
   The CLI script is configured to disable policy `1`:

```
config system automation-action
    edit "CLI Script POLICY-SWITCH"
        set description "switch policy"
        set action-type cli-script
        set script "config firewall policy
            edit 1
                 set status disable
            next
        end"
        set accprofile "super_admin"
    next
end
```

The email notification is configured:

```
config system automation-action
    edit "Email Notification"
        set description "Send a custom email to the specified recipient(s)."
        set action-type email
        set forticare-email enable
        set email-to "<username>@<company>.com"
        set email-subject "%%log.logdesc%%"
    next
end
```

To view the results:

1. When a triggered event occurs, a log is generated:

```
3: date=2026-02-18 time=14:48:53 eventtime=1771454932961997569 tz="-0800" logid="0100022907"
type="event" subtype="system" level="notice" vd="root" logdesc="Digital-IO input state change"
connector="input_IN2_REF" mode="default" state="closed" msg="The state of IN2_REF terminals
has changed from open to closed"
```

2. The automation stitch is triggered:

```
2: date=2026-02-18 time=14:48:54 eventtime=1771454933072245950 tz="-0800" logid="0100046600"
type="event" subtype="system" level="notice" vd="root" logdesc="Automation stitch triggered"
stitch="GPIO-2 off" trigger="GPIO Input2 Off" stitchaction="CLI Script POLICY-SWITCH,Email
Notification" from="log" msg="stitch:GPIO-2 off is triggered."

1: date=2026-02-18 time=14:48:54 eventtime=1771454933105206552 tz="-0800" logid="0100032549"
type="event" subtype="system" level="information" vd="root" logdesc="Autoscript stop
automatically" msg="script autod.20 stopped automatically"
```

3. The script disables the firewall policy 1:

```
show firewall policy 1
config firewall policy
    edit 1
        set status disable
        set uuid b3cee3aa-0cfe-51f1-500c-9b12a2cd8fa0
```

```
        set srcintf "wan1"
        set dstintf "lan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

4. An email notification is sent:

```
ALERT : This message originated outside of the Fortinet network.  BE CAUTIOUS before clicking
any link or opening an attachment.

Time: 2026-02-18 14:48:52
Device: FGR70GTK25000389 (root)
Level: notice
Event: Digital-IO input state change
Raw log:
date=2026-02-18 time=14:48:52 devid="FGR70GTK25000389" devname="FortiGateRugged-70G"
eventtime=1771454932961997569 tz="-0800" logid="0100022907" type="event" subtype="system"
level="notice" vd="root" logdesc="Digital-IO input state change" connector="input_IN2_REF"
mode="default" state="closed" msg="The state of IN2_REF terminals has changed from open to
closed"
```

5. Events triggered by FGR GPIO status changes are captured in event log ID 22907. Sample reference logs:
   - GPIO IN1: When opened (unlink IN1 and REF), FGR generates a new log:

     ```
     3: date=2026-02-18 time=14:08:44 eventtime=1771452524062323521 tz="-0800"
     logid="0100022907" type="event" subtype="system" level="notice" vd="root"
     logdesc="Digital-IO input state change" connector="input_IN1_REF" mode="default"
     state="open" msg="The state of IN1_REF terminals has changed from closed to open"
     ```

   - GPIO IN1: When closed (link IN1 and REF), FGR generates a new log:

     ```
     9: date=2026-02-18 time=14:08:42 eventtime=1771452522023646550 tz="-0800"
     logid="0100022907" type="event" subtype="system" level="notice" vd="root"
     logdesc="Digital-IO input state change" connector="input_IN1_REF" mode="default"
     state="closed" msg="The state of IN1_REF terminals has changed from open to closed"
     ```

   - GPIO IN2: When opened (unlink IN2 and REF), FGR generates a new log:

     ```
     18: date=2026-02-18 time=14:08:00 eventtime=1771452480343143518 tz="-0800"
     logid="0100022907" type="event" subtype="system" level="notice" vd="root"
     logdesc="Digital-IO input state change" connector="input_IN2_REF" mode="default"
     state="open" msg="The state of IN2_REF terminals has changed from closed to open"
     ```

   - GPIO IN2: When closed (link IN2 and REF), FGR generates a new log:

     ```
     22: date=2026-02-18 time=14:07:59 eventtime=1771452479321079860 tz="-0800"
     logid="0100022907" type="event" subtype="system" level="notice" vd="root"
     ```

```
logdesc="Digital-IO input state change" connector="input_IN2_REF" mode="default"
state="closed" msg="The state of IN2_REF terminals has changed from open to closed"
```

Index

The following index provides a list of all new features added to FortiOS 8.0. The index allows you to quickly identify the version where the feature first became available in FortiOS.

Select a version number to navigate in the index to the new features available for that patch:

- 8.0.0 on page 601

8.0.0

GUI

General usability enhancements	• Summary panel in Log Details on page 10 • GUI support for preferred outbound route map options on page 13 • Seven-day setup period for GUI and CLI configuration on page 14 • VDOM selection for Central Management in the GUI on page 15 • GUI support for configuring proxy ARP on VLAN interface on page 16 • Improve switch management on page 17 • Display Known Exploited Vulnerabilities from FortiClient on page 20 • Custom GUI themes and admin-level personalization on page 26 • Enhanced GUI support for Inline-CASB SaaS Application on page 32
FortiView	• Graphical display for application performance monitoring analytics on page 35

Network

General	• IPv6 Multicast Routing Enhancement with BSR Support Added on page 41 • BGP neighbor naming support on page 48 • Increased AS path item for BGP route maps on page 51 • IPv6 PCP support for DNAT46 in NAT64 deployments on page 52 • Customizable CoS marking for locally generated ARP packets on page 55 • Firewall address support for IPAM rules on page 59 • Graceful BGP shutdown on page 64 • TCP congestion control enhancement with BBR on page 72 • Session helper statistics on page 74

	• Enhanced EVPN support for VXLAN Anycast gateway and type-5 routing on page 75 • Enhanced PIM support for IPv6 across all VRFs on page 91 • GPON support for FortiGates with SFP/SFP+ ports on page 97 • 802.1X, MAB, dynamic VLAN, and address object support for software switch on page 99 • Dynamic BGP learning of ISDB reputation IPs on page 116 • Enhanced multicast session key for virtual wire pair on page 119 • Next-hop self-support for VPNv4 and VPNv6 route reflectors on page 121
IPv6	• IPv6 Link-local configuration enhancement on page 122 • IPv6 proxy address and address group object support on page 123
Explicit and transparent proxy	• Secure explicit proxy with client certificate blocklist enforcement on page 125

SD-WAN

Overlays and underlays	• SD-WAN underlay bandwidth steers traffic on page 128 • ToS-based SD-WAN duplication matching on page 135 • Stop packet duplication upon reaching bandwidth utilization threshold on page 141 • Selective SD-WAN duplication on page 147 • Use secondary tunnel for FEC redundant parity packets on page 152 • ToS matching and negate options on adaptive FEC profiles on page 159 • Control SD-WAN interface usage based on monthly traffic volume (quota) on page 166 • Confine packet duplication across multiple overlays to the same spoke on page 173 • On-demand duplication at the Hub using remote health-check on page 178
Performance SLA	• SD-WAN speed test scheduling improvements on page 188

Policy and objects

Policies	• Upstream SSL/TLS support for virtual load-balancing servers on page 196
Objects	• Custom tags for addresses, policies, and dynamic tag address groups on page 199

Zero Trust Network Access

General	• ZTNA configuration simplification on page 212 • ZTNA service connector on page 235
Security posture and EMS connectors	• Support tags in dual stack IPv4/IPv6 ZTNA policies on page 242

Security Profiles

Others	• File filter warning action for HTTP downloads on page 246 • Local URL filtering with custom FortiGuard categories on page 250 • New classification framework for application signatures and FortiView support on page 256 • Support for hybrid PQC SSL deep inspection in proxy mode on page 272 • Enhanced UTM protection with WebSocket traffic inspection on page 275 • FortiGuard-free rating for local and external categories on page 282 • Expanded CASB SaaS application database on page 284
Protecting LLM and GenAI	• Application control support for generative AI on page 286 • Agentic AI protocol support in FortiOS on page 286

VPN

IPsec and Agentless VPN	• New Cloud SDN Orchestration VPN wizard for AWS on page 293 • Support matching firewall policies and policy routes based on source IP geography of dial-up IPsec remote users on page 307 • Send security posture tags from FortiClient to FortiOS directly as JWT on page 311 • TLS 1.3 based VPN over TCP on page 316 • Native VPN remote access support in VPN wizard on page 323 • Default to IPAM addressing for remote access VPNs in VPN Wizard on page 341 • Display interface and tunnel alias in various diagnose and get commands on page 346 • Support SM3/SM4 cryptographic algorithms for IKEv1/ IKEv2 on page 350 • Show only recommended IKE proposals by default on page 356

User & Authentication

Authentication	• Enable secure LDAP connection by default in the GUI on page 358

LAN Edge

Wireless	• Support FortiAP management through IPv6 on page 360 • Improve CAPWAP stability on page 362 • Support dynamic redirect URLs from Cisco ISE on page 364 • Support IPsec traffic offloading from the FortiAP on page 367 • Enhance DARRP with FortiAIOps on page 369 • Support up to 16 VAPs on FortiAP G and K-series models on page 372 • Support Wi-Fi 7 MLO on FortiAP K-series models on page 374 • Support FQDN in Layer 3 ACLs on page 377 • Add GUI support for 6 GHz Channel Utilization on page 379 • Support customizable DHCP Option 82 configurations on page 380 • Optimize and update default DARRP parameters on page 383 • Add GUI support for capturing wireless packets on page 384 • Support tuning Clear Channel Assessment threshold values on page 387 • Allow FortiAP radio and FortiWiFi local radio to be disabled on page 391 • Support DVLAN assignments from RADIUS and VLAN pooling together on page 394 • Support multiple FortiAP groups per VLAN entry for VLAN pooling on page 396 • Add RADIUS authentication survivability for 802.1X SSIDs on page 396 • Automated FortiAP location positioning system on page 399 • Support FortiGuest assignments of RADIUS attributes to MPSK groups on page 404
Switch Controller	• Add maximum power control for PoE ports on page 408 • Increase number of supported switches on page 409 • Advertise and control EEE using LLDP in the switch controller on page 409 • Support trusted host settings for managed switches on page 410 • Control when custom commands are pushed on page 412 • Provide private data encryption for managed switches on page 414 • Support MAC move for managed switches on page 415 • Enable advanced switching features in the GUI on page 420 • Control the number of 802.1X-authenticated clients per port on page 422 • Support IGMP-snooping static group configuration on page 422 • Dynamic MAB session handling on page 423

	• Enhance NAC/DPP port control and QoS/PoE support on page 424 • Add global port-selection criteria for some switch models on page 426 • Using NAC with 802.1X authentication in the FortiOS GUI on page 427

System

General	• Configurable timeout for log file system check on page 431 • Administrator access hardening on page 433 • SCP file transfers on page 434 • VM license grace period monitoring enhancement on page 437 • Configurable FQDN Host header for FortiGate proxy communication on page 438 • Additional commands supported before device registration on page 440 • Legal third party disclosure panel on page 440 • Per VDOM replacement message customization on page 442 • Configuration revisions and logout backup default changes on page 444 • FortiAI assistant and CLI Code Lab on page 446 • Enhancements to required upgrade when firmware license is invalid or device is EOES on page 458
High availability	• IPv6 support for in-band management IP on page 468 • Support IPv6 multicast route synchronization in HA on page 469 • FGSP source IP and interface selection on page 470 • HA monitoring support for software switch member interfaces on page 472 • FGCP and FGSP synchronizes full-cone expectation sessions when session-pickup-expectation enabled on page 475 • Enhanced FGCP monitoring with interface group awareness on page 481 • GUI support for HA actions, health status, and config diff on page 485 • FGSP synchronizes firewall-authenticated users on page 487 • HA role shown in CLI prompt on page 492
SNMP	• SNMP support for new IP pool statistics OIDs on page 492
FortiGuard	• Enhanced FortiGuard server selection and load balancing on page 495 • FortiGuard rating server prioritization improvements on page 503
API	• API support to manage firmware for FortiGate extended devices on page 506

Security Fabric

Fabric settings and connectors	• Cloud-based Fabric Feed synchronization on page 512

	• Label-based sandbox exemptions for sensitive data control on page 517 • Simultaneous FortiSandbox Cloud and on-premise integration on page 520
External SDN connectors	• Support for Cisco ACI External EPG Subnets in Direct Connector on page 536 • Negation support in ACI direct connector address filters on page 542

Log & Report

Logging	• Enhanced TACACS+ accounting log detail on page 547 • Custom log format support for syslog server on page 549 • Secure log upload enhancements with SFTP and LZ4 support on page 558

Cloud

Public and private cloud	• AWS SDN connector EKS filtering support on page 562

FortiOS Carrier

FortiOS Carrier	• Selective GTP FGSP sync: only S10 tunnels on page 565 • Selective GTP FGSP sync: only synchronize GTP tunnels on page 569 • New RAT types on page 573 • 3G to 4G/5G handover support on page 575 • Increase GTP tunnel limit per profile to 50M tunnels on page 577

FortiASIC

Hardware acceleration	• Improvements for NP7-offloaded GRE tunnels on page 579 • NP7 offloading of TCP and UDP sessions denied by firewall policies to reduce CPU usage on page 581 • NP7 support for traffic shaping based on dynamic RADIUS VSAs on page 582
Hyperscale firewall	• New SNMP counters to support CGNAT IP pool monitoring on page 582 • Hyperscale HA hardware session synchronization improves support for PBR sessions on page 586 • Hyperscale CGNAT EIF session timer options on page 587

Operational Technology

System	• MACsec support for FortiGate Rugged Models with hardware switch on page 592 • FortiGate Rugged digital I/O trigger logging enhancement on page 593

www.fortinet.com

Copyright© 2026 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's Chief Legal Officer, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.