

New Features Guide

**SD-WAN with FortiOS, FortiManager, and
FortiAnalyzer 7.2.x**



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



March 26, 2024

SD-WAN with FortiOS, FortiManager, and FortiAnalyzer 7.2.x New Features Guide

01-72X-792697-20240326

TABLE OF CONTENTS

Change Log	5
Overview	6
What's new in 7.2.0	6
What's new in 7.2.1	7
What's new in 7.2.2	7
What's new in 7.2.3	8
What's new in 7.2.4	8
What's new in 7.2.6	8
ADVPN	9
Phase 2 selectors and ADVPN shortcut tunnels	9
SD-WAN members' local cost exchange on ADVPN shortcut tunnels	9
Exchange underlay link cost property with remote peer in IPsec VPN phase 1 negotiation	
7.2.1	9
Monitoring	15
VPN Monitoring displays IPsec VPN tunnels created by IPsec templates and SD-WAN overlay wizard FMG 7.2.3	15
SD-WAN application monitor using FortiMonitor 7.2.4	15
Example	16
Improve client-side settings for SD-WAN network monitor 7.2.6	18
Provisioning	19
SD-WAN overlay templates FMG	19
Prerequisites and network planning	20
Using the SD-WAN overlay template	20
Configuring an SD-WAN overlay template	20
Metafield support on dynamic objects FMG	25
Model device blueprints FMG	27
Creating a device blueprint	27
Adding model devices using a blueprint	28
Application categories in SD-WAN rules FMG	30
Fabric Authorization Template automatically provisions and authorizes LAN Edge devices on managed FortiGates FMG 7.2.1	36
Internet service database version checked for model devices FMG 7.2.1	39
FortiManager supports BYOL installation on managed FortiGate VMs FMG 7.2.1	40
Pre-built route-maps used for SD-WAN self-healing with BGP routing FMG 7.2.2	43
FortiManager supports multiple interface members in the SD-WAN neighbor configuration FMG 7.2.2	46
Factory default firewall addresses and address group for private IP space (RFC1918) FMG 7.2.2	47
Interface-based traffic shaping can display real time dropped packets FMG 7.2.2	49
Add Fabric Overlay Orchestrator for SD-WAN overlay configurations 7.2.4	52
Prerequisites	52
Network topology	53

Using the Fabric Overlay Orchestrator	53
Reporting	55
SD-WAN chart to include more ADVPN shortcut information FAZ	55
SD-WAN chart for MOS scoring FAZ	57
Bandwidth and applications report update FAZ 7.2.1	61
Routing	64
SD-WAN segmentation over a single overlay	64
New SD-WAN options	64
New IPsec options	65
New VPN configuration for BGP	66
Display BGP routes by VRF and neighbor	67
Examples	67
Multiple members per SD-WAN neighbor configuration	79
Example	80
SD-WAN in large scale deployments	85
Route map rules and BGP routes	97
BGP socket limit increase	97
IKE embryonic limit	97
GUI support for advanced BGP options 7.2.1	97
Support BGP AS number input in asdot and asdot+ format 7.2.1	100
Example	101
Support cross-VRF local-in and local-out traffic for local services 7.2.1	102
Example	102
Matching BGP extended community route targets in route maps 7.2.4	104
Example	105
Add static route tag and BGP neighbor password 7.2.4	109
Example 1	109
Example 2	111
SD-WAN steering	112
Allow application category as an option for SD-WAN rule destination	112
Example	113
Add mean option score calculation and logging in performance SLA health checks	117
Allow application category as a GUI option for SD-WAN rule destination 7.2.1	119
Embedded SD-WAN SLA information in ICMP probes 7.2.1	121
Example with BGP on loopback SD-WAN	123
SD-WAN Template added the health-check embedded SLA information FMG 7.2.2	129
Visibility	132
Traffic shaping charts FAZ 7.2.1	132
High bandwidth application usage report update FAZ 7.2.1	135
WAN remediation	138
Duplication on-demand when SLAs in the configured service are matched	138
Results	140

Change Log

Date	Change Description
2022-04-11	Initial release for 7.2.0. See Overview on page 6 .
2022-08-04	Updated for FortiOS 7.2.1 release. See Overview on page 6 .
2022-08-31	Added Traffic shaping charts FAZ 7.2.1 on page 132 .
2022-09-06	Added: <ul style="list-style-type: none">• Bandwidth and applications report update FAZ 7.2.1 on page 61• High bandwidth application usage report update FAZ 7.2.1 on page 135
2022-09-26	Added <ul style="list-style-type: none">• Fabric Authorization Template automatically provisions and authorizes LAN Edge devices on managed FortiGates FMG 7.2.1 on page 36 FortiManager supports BYOL installation on managed FortiGate VMs FMG 7.2.1 on page 40• FortiManager supports BYOL installation on managed FortiGate VMs FMG 7.2.1 on page 40
2022-12-22	Added: <ul style="list-style-type: none">• Allow application category as a GUI option for SD-WAN rule destination 7.2.1 on page 119• Internet service database version checked for model devices FMG 7.2.1 on page 39.
2023-01-09	Updated: <ul style="list-style-type: none">• SD-WAN in large scale deployments on page 85• Route map rules and BGP routes on page 97
2023-01-31	Updated for FortiOS 7.2.4. See Overview on page 6 .
2023-02-02	Updated for FortiManager 7.2.2. See Overview on page 6 .
2023-04-14	Updated Allow application category as an option for SD-WAN rule destination on page 112 .
2023-07-19	Updated Embedded SD-WAN SLA information in ICMP probes 7.2.1 on page 121 .
2024-03-26	Updated for FortiManager 7.2.3. See What's new in 7.2.3 on page 8 . Updated for FortiOS 7.2.6. See What's new in 7.2.6 on page 8 .

Overview

This guide provides details of new features for SD-WAN introduced in FortiOS 7.2, FortiManager 7.2, and FortiAnalyzer 7.2.

- [What's new in 7.2.0 on page 6](#)
- [What's new in 7.2.1 on page 7](#)
- [What's new in 7.2.2 on page 7](#)
- [What's new in 7.2.3 on page 8](#)
- [What's new in 7.2.4 on page 8](#)
- [What's new in 7.2.6 on page 8](#)

For each feature, the guide provides detailed information on configuration, requirements, and limitations, as applicable. For features introduced in FortiManager or FortiAnalyzer, the short product name is appended to the end of the topic heading, for example FMG or FAZ.

For features introduced in 7.2.1 and later versions, the version number is appended to the end of the topic heading. For example, [Support cross-VRF local-in and local-out traffic for local services 7.2.1 on page 102](#) was introduced in 7.2.1. If a topic heading has no version number at the end, the feature was introduced in 7.2.0.

For features introduced in FortiManager or FortiAnalyzer 7.2.1 and later versions, the short product name and version number are appended to the end of the topic heading.

What's new in 7.2.0

Feature	Details
ADVPN	<ul style="list-style-type: none">• Phase 2 selectors and ADVPN shortcut tunnels on page 9• SD-WAN members' local cost exchange on ADVPN shortcut tunnels on page 9
Provisioning	<ul style="list-style-type: none">• SD-WAN overlay templates FMG on page 19• Metafield support on dynamic objects FMG on page 25• Model device blueprints FMG on page 27• Application categories in SD-WAN rules FMG on page 30
Reporting	<ul style="list-style-type: none">• SD-WAN chart to include more ADVPN shortcut information FAZ on page 55• SD-WAN chart for MOS scoring FAZ on page 57
Routing	<ul style="list-style-type: none">• SD-WAN segmentation over a single overlay on page 64• Multiple members per SD-WAN neighbor configuration on page 79• SD-WAN in large scale deployments on page 85• Route map rules and BGP routes on page 97• BGP socket limit increase on page 97• IKE embryonic limit on page 97

Feature	Details
SD-WAN steering	<ul style="list-style-type: none"> Allow application category as an option for SD-WAN rule destination on page 112 Add mean option score calculation and logging in performance SLA health checks on page 117
WAN remediation	<ul style="list-style-type: none"> Duplication on-demand when SLAs in the configured service are matched on page 138

What's new in 7.2.1

Feature	Details
ADVPN	<ul style="list-style-type: none"> Exchange underlay link cost property with remote peer in IPsec VPN phase 1 negotiation 7.2.1 on page 9
Provisioning	<ul style="list-style-type: none"> Fabric Authorization Template automatically provisions and authorizes LAN Edge devices on managed FortiGates FMG 7.2.1 on page 36 Internet service database version checked for model devices FMG 7.2.1 on page 39 FortiManager supports BYOL installation on managed FortiGate VMs FMG 7.2.1 on page 40
Reporting	<ul style="list-style-type: none"> Bandwidth and applications report update FAZ 7.2.1 on page 61
Routing	<ul style="list-style-type: none"> GUI support for advanced BGP options 7.2.1 on page 97 Support BGP AS number input in asdot and asdot+ format 7.2.1 on page 100 Support cross-VRF local-in and local-out traffic for local services 7.2.1 on page 102
SD-WAN steering	<ul style="list-style-type: none"> Allow application category as a GUI option for SD-WAN rule destination 7.2.1 on page 119 Embedded SD-WAN SLA information in ICMP probes 7.2.1 on page 121
Visibility	<ul style="list-style-type: none"> Traffic shaping charts FAZ 7.2.1 on page 132 High bandwidth application usage report update FAZ 7.2.1 on page 135

What's new in 7.2.2

Feature	Details
Provisioning	<ul style="list-style-type: none"> Pre-built route-maps used for SD-WAN self-healing with BGP routing FMG 7.2.2 on page 43 FortiManager supports multiple interface members in the SD-WAN neighbor configuration FMG 7.2.2 on page 46

Feature	Details
	<ul style="list-style-type: none">• Factory default firewall addresses and address group for private IP space (RFC1918) FMG 7.2.2 on page 47• Interface-based traffic shaping can display real time dropped packets FMG 7.2.2 on page 49
SD-WAN steering	<ul style="list-style-type: none">• SD-WAN Template added the health-check embedded SLA information FMG 7.2.2 on page 129

What's new in 7.2.3

Feature	Details
Monitoring	<ul style="list-style-type: none">• VPN Monitoring displays IPsec VPN tunnels created by IPsec templates and SD-WAN overlay wizard FMG 7.2.3 on page 15

What's new in 7.2.4

Feature	Details
Monitoring	<ul style="list-style-type: none">• SD-WAN application monitor using FortiMonitor 7.2.4 on page 15
Provisioning	<ul style="list-style-type: none">• Add Fabric Overlay Orchestrator for SD-WAN overlay configurations 7.2.4 on page 52
Routing	<ul style="list-style-type: none">• Matching BGP extended community route targets in route maps 7.2.4 on page 104• Add static route tag and BGP neighbor password 7.2.4 on page 109

What's new in 7.2.6

Feature	Details
Monitoring	<ul style="list-style-type: none">• Improve client-side settings for SD-WAN network monitor 7.2.6 on page 18

ADVPN

7.2.0

- [Phase 2 selectors and ADVPN shortcut tunnels on page 9](#)
- [SD-WAN members' local cost exchange on ADVPN shortcut tunnels on page 9](#)

7.2.1

- [Exchange underlay link cost property with remote peer in IPsec VPN phase 1 negotiation 7.2.1 on page 9](#)

Phase 2 selectors and ADVPN shortcut tunnels

Phase 2 selectors can be used to inject IKE routes on the ADVPN shortcut tunnel. When configuration method (`mode-cfg`) is enabled in IPsec phase 1 configuration, enabling `mode-cfg-allow-client-selector` allows custom phase 2 selectors to be configured. By also enabling the addition of a route to the peer destination selector (`add-route`) in the phase 1 configuration, IKE routes based on the phase 2 selectors can be injected. This means that routes do not need to be reflected on the hub to propagate them between spokes, avoiding possible BGP daemon process load issues and improving network scalability in a large-scale ADVPN network.

For details, see [SD-WAN in large scale deployments on page 85](#).

SD-WAN members' local cost exchange on ADVPN shortcut tunnels

SD-WAN members' local cost can be exchanged on the ADVPN shortcut tunnel so that spokes can use the remote cost as tiebreak to select a preferred shortcut. If multiple shortcuts originate from the same member to different members on the same remote spoke, then the remote cost on the shortcuts is used as the tiebreak to decide which shortcut is preferred.

For details, see [SD-WAN in large scale deployments on page 85](#).

Exchange underlay link cost property with remote peer in IPsec VPN phase 1 negotiation - 7.2.1



This information is also available in the FortiOS 7.2 Administration Guide:

- [SD-WAN in large scale deployments](#)

The underlay link cost property has been added to the IPsec VPN tunnel phase 1 configuration and enhances the IPsec VPN to exchange the link cost with a remote peer as a private notify payload in IKEv1 and IKEv2 phase 1 negotiations.

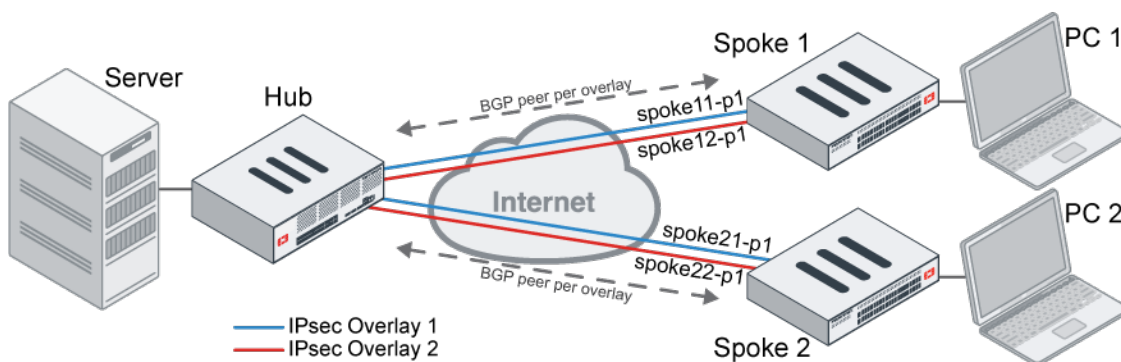
This avoids possible health check daemon process load issues in the previous implementation of the link cost exchange feature, and it improves network scalability in a large-scale SD-WAN network with ADVPN.

```
config vpn ipsec phase1-interface
  edit <name>
    set link-cost <integer>
  next
end
```

link-cost <integer>

Set the VPN underlay link cost (0 - 255, default = 0).

If multiple shortcuts originate from the same SD-WAN member to different members on the same remote spoke, learned remote IPsec link costs on shortcuts will be used as a tie-breaker to decide which shortcut is preferred.



In this example, SD-WAN is configured on an ADVPN network with a BGP neighbor per overlay.

Instead of reflecting BGP routes with the route-reflector on the hub, when the shortcuts are triggered, IKE routes on the shortcuts are directly injected based on the configured phase 2 selectors to allow routes to be exchanged between spokes.

Routes between the hub and the spokes are exchanged by BGP, and the spokes use the default route to send spoke-to-spoke traffic to the hub and trigger the shortcuts.

To configure Spoke 1:

1. Configure the VPN remote gateway:

```
config vpn ipsec phase1-interface
  edit "spoke11-p1"
    ...
    set mode-cfg-allow-client-selector enable
    set link-cost 11
  next
  edit "spoke12-p1"
    ...
    set mode-cfg-allow-client-selector enable
    set link-cost 21
  next
end
```

2. Configure the SD-WAN settings:

```
config system sdwan
  set status enable
```

```
config zone
    edit "virtual-wan-link"
    next
end
config members
    edit 1
        set interface "spoke11-p1"
        set cost 10
    next
    edit 2
        set interface "spoke12-p1"
        set cost 20
    next
end
config health-check
    edit "1"
        set server "9.0.0.1"
        set members 0
        config sla
            edit 1
                next
            end
        next
    next
end
config service
    edit 1
        set name "1"
        set mode sla
        set dst "all"
        set src "10.1.100.0"
        config sla
            edit "1"
                set id 1
            next
        end
        set priority-members 1 2
    next
end
end
```

To configure Spoke 2:

1. Configure the VPN remote gateway:

```
config vpn ipsec phase1-interface
    edit "spoke21-p1"
        ...
        set link-cost 101
    next
    edit "spoke22-p1"
        ...
        set link-cost 201
    next
end
```

2. Configure the SD-WAN settings:

```

config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "spoke21-p1"
            set cost 10
        next
        edit 2
            set interface "spoke22-p1"
            set cost 20
        next
    end
    config health-check
        edit "1"
            set server "9.0.0.1"
            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 "192.168.5.0"
            config sla
                edit "1"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
end

```

To test the configuration:

1. Verify the service diagnostics on Spoke 1:

```

# diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Tie break: cfg
Gen(4), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
Members(2):
    1: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(10),
selected
    2: Seq_num(2 spoke12-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(20),

```

```

selected
  Src address(1):
    10.1.100.0-10.1.100.255

  Dst address(1):
    0.0.0.0-255.255.255.255

```

2. Verify the service diagnostics on Spoke 2:

```

# diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Tie break: cfg
  Gen(2), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 spoke21-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(10),
selected
    2: Seq_num(2 spoke22-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(20),
selected
  Src address(1):
    192.168.5.0-192.168.5.255

  Dst address(1):
    0.0.0.0-255.255.255.255

```

3. Trigger shortcuts between Spoke 1 and Spoke 2:

- Shortcuts spoke11-p1_1 and spoke11-p1_0 originate from spoke11-p1.
- spoke11-p1_1 corresponds to spoke21-p1_0 on Spoke 2.
- spoke11-p1_0 corresponds to spoke22-p1_0 on Spoke 2.

Spoke 1:

```

# diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Tie break: cfg
  Gen(11), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Member sub interface(4):
    3: seq_num(1), interface(spoke11-p1):
      1: spoke11-p1_0(80)
      2: spoke11-p1_1(81)
  Members(4):
    1: Seq_num(1 spoke11-p1_1), alive, sla(0x1), gid(0), remote cost(101), cfg_order(0),
local cost(10), selected
    2: Seq_num(1 spoke11-p1_0), alive, sla(0x1), gid(0), remote cost(201), cfg_order(0),
local cost(10), selected
    3: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(10),
selected
    4: Seq_num(2 spoke12-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(20),
selected
  Src address(1):
    10.1.100.0-10.1.100.255

  Dst address(1):
    0.0.0.0-255.255.255.255

```

Spoke 2:

```
# diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Tie break: cfg
Gen(15), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
Member sub interface(4):
  2: seq_num(1), interface(spoke21-p1):
    1: spoke21-p1_0(75)
  4: seq_num(2), interface(spoke22-p1):
    1: spoke22-p1_0(74)
Members(4):
  1: Seq_num(1 spoke21-p1_0), alive, sla(0x1), gid(0), remote cost(11), cfg_order(0),
local cost(10), selected
  2: Seq_num(1 spoke21-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(10),
selected
  3: Seq_num(2 spoke22-p1_0), alive, sla(0x1), gid(0), remote cost(11), cfg_order(1),
local cost(20), selected
  4: Seq_num(2 spoke22-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(20),
selected
Src address(1):
  192.168.5.0-192.168.5.255

Dst address(1):
  0.0.0.0-255.255.255.255
```

The spoke11-p1_1 shortcut on Spoke 1 is preferred over spoke11-p1_0 due to the lower remote link cost of 101 when they have the same local SD-WAN member cost of 10.

4. Verify the policy route list on Spoke 1:

```
# diagnose firewall proute list
list route policy info(vf=root):

id=2131755009(0x7f100001) vwl_service=1(1) vwl_mbr_seq=1 1 1 2 dscp_tag=0xfc 0xfc
flags=0x0 tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path
(4) oif=81(spoke11-p1_1) oif=80(spoke11-p1_0) oif=54(spoke11-p1) oif=55(spoke12-p1)
source(1): 10.1.100.0-10.1.100.255
destination(1): 0.0.0.0-255.255.255.255
hit_count=176 last_used=2022-07-12 11:56:08
```

Monitoring

7.2.3

- [VPN Monitoring displays IPsec VPN tunnels created by IPsec templates and SD-WAN overlay wizard FMG 7.2.3 on page 15](#)

7.2.4

- [SD-WAN application monitor using FortiMonitor 7.2.4 on page 15](#)

7.2.5

- [Improve client-side settings for SD-WAN network monitor 7.2.6 on page 18](#)

VPN Monitoring displays IPsec VPN tunnels created by IPsec templates and SD-WAN overlay wizard - FMG 7.2.3

VPN Monitoring displays IPsec VPN tunnels created by IPsec Templates and the SD-WAN Overlay Wizard with specific device icon identification for HUBs and the ability to drilldown to a device group level.

For more information about this feature, see [VPN Monitoring displays IPsec VPN tunnels created by IPsec templates and the SD-WAN overlay wizard](#).

SD-WAN application monitor using FortiMonitor - 7.2.4



This information is also available in the FortiOS 7.2 Administration Guide:

- [SD-WAN application monitor using FortiMonitor](#)

The agent-based health check detection mode works with FortiMonitor to provide more accurate user level performance statistics. FortiMonitor acts as an agent and sends health check probes on behalf of the monitored FortiGate interface. FortiMonitor mimics a real user, and the probes return a more accurate application level performance. The SLA information collected from FortiMonitor is sent back to the FortiGate as the monitored interface's SLA information.

```
config system sdwan
  config health-check
    edit <name>
      set detect-mode agent-based
    next
  end
  config service
    edit <id>
      set agent-exclusive {enable | disable}
    next
```

```
end
end
```

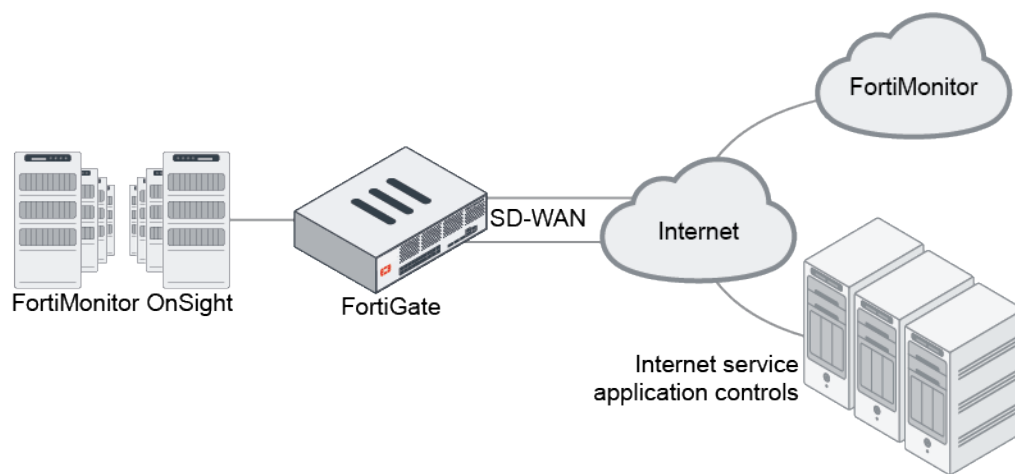
The following diagnostic commands can be used to view agent related metrics:

```
# diagnose sys link-monitor-passive agent <option>
```

list	List all the collected reports.
list-app	List the details of each application.
flush	Flush all the collected reports.
flush-app	Flush the details of all the applications.
agent-oif-map	List the agent and interface maps.

Example

In this example, routing is achieved through SD-WAN rules. The agent-based health check detection mode creates the FortiMonitor IP address and FortiGate SD-WAN interface map.



This example assumes that the FortiMonitor has already been added to the Security Fabric (see [Configuring FortiMonitor](#) in the FortiOS Administration Guide for detailed instructions). The FortiMonitor OnSight (client) can be configured for two or more IP addresses, and each IP address is capable of sending application probes to user-specified applications.

Specific routing is implemented on the FortiGate to ensure each FortiMonitor client collects performance statistics for only one SD-WAN member interface. The FortiMonitor is configured to send application-specific probes to measure that application's performance on a given SD-WAN member. The FortiGate uses the FortiMonitor performance statistics to determine link quality based on application performance by mapping the health check. The link quality for a given application can then be used to steer the matching application traffic with greater accuracy.

To configure the FortiGate:

1. Configure the address objects for each FortiMonitor client:

```
config firewall address
  edit "FMR_OnSight1"
```

```

        set subnet 10.2.1.80 255.255.255.255
    next
    edit "MR_OnSight2"
        set subnet 10.2.1.81 255.255.255.255
    next
end

```

2. Configure the SD-WAN rules to ensure each OnSight client uses only one SD-WAN member, and map the FortiMonitor IP to an SD-WAN member (interface):

```

config system sdwan
    config service
        edit 1
            set dst "all"
            set src "FMR_OnSight1"
            set priority-members 1
            set agent-exclusive enable
        next
        edit 2
            set dst "all"
            set src "FMR_OnSight2"
            set priority-members 2
            set agent-exclusive enable
        next
    end
end

```

3. Configure the SD-WAN health check:

```

config health-check
    edit "FMR"
        set detect-mode agent-based
        set members 1 2
        config sla
            edit 1
                next
            end
        next
    end
end

```

To verify the SD-WAN member performance:

1. Verify the health check diagnostics:

```

# diagnose sys sdwan health-check
Health Check(FMR):
Seq(1 v1236): state(alive), packet-loss(0.000%) latency(183.214), jitter(0.124), mos
(4.225), bandwidth-up(999992), bandwidth-dw(999976), bandwidth-bi(1999968) sla_map=0x0
Seq(2 v1237): state(alive), packet-loss(0.000%) latency(182.946), jitter(0.100), mos
(4.226), bandwidth-up(999998), bandwidth-dw(999993), bandwidth-bi(1999991) sla_map=0x0

```

2. Verify the collected reports:

```

# diagnose sys link-monitor-passive agent list
v1236( 23) | src=10.2.1.80 | latency=183.2    20:27:24 | jitter=0.1    20:27:24 |
pktloss=0.0  % 20:27:24
v1237( 24) | src=10.2.1.81 | latency=182.9    20:27:24 | jitter=0.1    20:27:24 |
pktloss=0.0  % 20:27:24

```

3. Verify the details of each application:

```
# diagnose sys link-monitor-passive agent list-app
app_id=0x00000000, app=fortinet.com, dev=v1236(23)
    latency=183.2, jitter=0.1, pktloss=0.0, ntt=99.2, srt=384.8, app_err=0.0, 20:28:25
app_id=0x00000000, app=fortinet.com, dev=v1237(24)
    latency=183.1, jitter=0.5, pktloss=0.0, ntt=104.4, srt=377.8, app_err=0.0, 20:28:25
```

4. Verify the agent and interface maps:

```
# diagnose sys link-monitor-passive agent agent-oif-map
oif=v1236(23), src=10.2.1.80
oif=v1237(24), src=10.2.1.81
```

Improve client-side settings for SD-WAN network monitor - 7.2.6



This information is also available in the FortiOS 7.2 Administration Guide:

- [Speed test examples](#)

Improvements have been made to the client-side settings of the SD-WAN network bandwidth monitoring service to increase the flexibility of the speed tests, and to optimize the settings to produce more accurate measurements. The changes include:

- Support UDP speed tests.
- Support multiple TCP connections to the server instead of a single connection.
- Measure the latency to speed test servers and select the server with the smallest latency to perform the test.
- Support the auto mode speed test, which selects either UDP or TCP testing automatically based on the latency threshold.

For more information about this feature, see [Improve client-side settings for SD-WAN network monitor](#).

Provisioning

7.2.0

- [SD-WAN overlay templates FMG on page 19](#)
- [Metafield support on dynamic objects FMG on page 25](#)
- [Model device blueprints FMG on page 27](#)
- [Application categories in SD-WAN rules FMG on page 30](#)

7.2.1

- [Fabric Authorization Template automatically provisions and authorizes LAN Edge devices on managed FortiGates FMG 7.2.1 on page 36](#)
- [Internet service database version checked for model devices FMG 7.2.1 on page 39](#)
- [FortiManager supports BYOL installation on managed FortiGate VMs FMG 7.2.1 on page 40](#)

7.2.2

- [Pre-built route-maps used for SD-WAN self-healing with BGP routing FMG 7.2.2 on page 43](#)
- [FortiManager supports multiple interface members in the SD-WAN neighbor configuration FMG 7.2.2 on page 46](#)
- [Factory default firewall addresses and address group for private IP space \(RFC1918\) FMG 7.2.2 on page 47](#)
- [Interface-based traffic shaping can display real time dropped packets FMG 7.2.2 on page 49](#)

7.2.4

- [Add Fabric Overlay Orchestrator for SD-WAN overlay configurations 7.2.4 on page 52](#)

SD-WAN overlay templates - FMG



This information is also available in the FortiManager 7.2 Administration Guide:

- [SD-WAN overlay templates](#)

Most SD-WAN deployments require complex overlay configurations for datacenter or cloud connectivity. FortiManager 7.2.0 includes an SD-WAN overlay template with a wizard to automate and simplify the process using Fortinet's recommended IPsec and BGP templates.

This topic includes the following.

- [Prerequisites and network planning on page 20](#)
- [Using the SD-WAN overlay template on page 20](#)
- [Configuring an SD-WAN overlay template on page 20](#)

For more information, including editing a template and onboarding new SD-WAN branch devices, see the [FortiManager Administration Guide](#).

Prerequisites and network planning

Prerequisites

- Import the FortiGate devices that will make up the hub and branch devices into FortiManager.
- Configure the ISP links and other interfaces on your imported devices.
- Create a device group for your branch devices.

Network planning

- Allocate the overlay network address space. By default, the template uses 10.10.0.0/16.
- Allocate the loopback IP address space. By default, the template uses 172.16.0.0/16.
- Select an AS number for BGP for the new SD-WAN overlay region. By default, the template uses 65000.

Using the SD-WAN overlay template

To use the SD-WAN overlay template:

1. Pre-configure your network and SD-WAN devices.
2. Create an SD-WAN overlay template.
3. Assign metadata variables to devices. The branch_id variable is automatically created by the template and each branch device must be assigned a unique value. Additional custom metadata variables can be used if required.
4. Configure the SD-WAN rules to be used in your SD-WAN environment by editing the SD-WAN template.
5. Create the Policy Package for your branch and hub devices.
6. Install the changes to SD-WAN devices using the Install Wizard.
7. (Optional) Edit the SD-WAN overlay template.
8. (Optional) Add new branch devices.

Configuring an SD-WAN overlay template

To create an SD-WAN overlay template:

1. Go to *Device Manager > Provisioning Templates > SD-WAN Overlay Templates*.
2. Click *Create New*.
The Create New SD-WAN Overlay Template wizard opens.
3. Enter a name and description for the new SD-WAN overlay template, and click *OK*.

4. For the *Region Settings*, select a topology type, and click *Next*.

Edit SD-WAN Overlay Template - Region Settings (1/5)

Name: SD-WAN-TEMPLATE

Description:

Select New Topology

Single HUB

Dual HUB
(Primary & Secondary)

Dual HUB
(Primary & Primary)

Advanced

Loopback IP Address: 172.16.0.0/255.255.0.0

Overlay Network: 10.10.0.0/255.255.0.0

BGP-AS Number: 65000

Auto-Discovery VPN: ☐

Next > Last Cancel

Select New Topology

Select a topology type based on your environment. Topologies include the following:

- Single Hub
- Dual Hub (Primary/Secondary)
- Dual Hub (Primary/Primary)

The options presented in the wizard change based on the topology selected.



Primary/Secondary and Primary/Primary are the same configuration, with the difference being that in a Primary/Secondary deployment, the Secondary hub is given a higher cost than the Primary. This cost is controlled by the SDWAN rule.

Advanced

Expand to view additional configurable settings.

These fields are preconfigured with settings that will work in many situations, but you may need to adjust these to match your own networking environment. They should match the addresses you identified when considering the SD-WAN overlay template prerequisites.

Loopback IP Address

Optionally, you can configure the loopback IP address.
By default, this setting is set to 172.16.0.0/255.255.0.0.

Overlay Network

Optionally, you can configure the overlay network.
By default, this setting is set to 10.10.0.0/255.255.0.0.

BGP-AS Number

Optionally, you can configure the BGP AS number.
By default, this setting is set to 65000.

Auto-Discovery VPN

Optionally, you can toggle this setting ON to enable Auto Discovery VPN (ADVPN).

5. For the *Role Assignment*, configure the following settings and click *Next*.

Edit SD-WAN Overlay Template - Role Assignment (2/5)

Name: SD-WAN-TEMPLATE

Topology: Single HUB Dual HUB (Primary & Secondary) Dual HUB (Primary & Primary)

HUB

Primary HUB: Enterprise_HUB1

Secondary HUB: Enterprise_HUB2

Branch

Device Group Assignment: sd-wan-branches

< Back Next > Cancel

Topology

Optionally, you can change the topology type that you selected on the previous screen.

Hub

Select the SD-WAN hubs. The number of hubs required depend on the topology selected:

- *Single Hub*: One standalone hub.
- *Dual Hub (Primary & Secondary)*: One primary and one secondary hub.
- *Dual Hub (Primary & Primary)*: Two primary hubs.

Hub devices must be added to SD-WAN with FortiOS, FortiManager, and FortiAnalyzer before creating the SD-WAN overlay template.

Branch

Select the device group containing your SD-WAN branch devices.

Devices included in this device group are configured as SD-WAN branch devices as a part of this template.

Additional devices can be added to the selected device group later to receive the SD-WAN branch configuration when performing an installation on that device. This simplifies the onboarding of new branch devices.

6. For the *Network Configuration*, configure the following settings and click *Next*.

Edit SD-WAN Overlay Template - Network Configuration (3/5)

Name: SD-WAN-TEMPLATE

HUB

Primary HUB

WAN Underlay 1: Enterprise_HUB1

Private Link: ☐ 10.0.11.2

Override IP: ☐

Network Advertisement: **Connected** Static

Interface: +

Advanced >

Secondary HUB

WAN Underlay 1: Enterprise_HUB2

Private Link: ☐ 10.0.12.3

Override IP: ☐

Network Advertisement: **Connected** Static

Interface: +

Advanced >

Branch Route Maps

Route map in: ☐

Route map out: ☐

Branch

Branch Device Group: sd-wan-branches

WAN Underlay 1: 192.185.50.1

Private Link: ☐

Network Advertisement: **Connected** Static

Network Prefix: +

Advanced >

< Back Next > Cancel

Hub

Configure the network settings for each hub in your configuration. The number and types of hubs present depend on the topology you selected.

WAN Underlay

Type the interfaces for each WAN underlay. You can add additional WAN underlays by clicking the add icon.

For each WAN underlay, you can optionally enable the following settings:

- *Private Link*: No overlays will be created on private links.
- *Override IP*: Override the IP address for the WAN underlay with the provided IP address. This option is not available when *Private Link* is enabled.

Network Advertisement

1. Configure network advertisement for the hub. Network advertisement can be set to one of the following:
 - *Connected*: Type the network interface to advertise. Additional interfaces can be added by clicking the add icon.
 - *Static*: Type the network prefix to advertise. Additional network prefixes can be added by clicking the add icon.

Advanced	Expand to view advanced settings, including configuration of SD-WAN neighbors. Click <i>Neighbors > Create New</i> to add a new SD-WAN neighbor for the hub.
Branch Route Maps	Optionally, move the toggle to the ON position to enable branch maps, and then select the corresponding route map. You can create a new route map by clicking the add icon.
Branch	Configure the network settings for the branch devices in your configuration.
WAN Underlay	Type the interfaces for the SD-WAN branch WAN underlay. You can add additional WAN underlays by clicking the add icon. For each WAN underlay, you can optionally enable the following settings: <ul style="list-style-type: none"> • <i>Private Link</i>: No overlays will be created on private links.
Network Advertisement	Configure network advertisement for the branch. Network advertisement can be set to one of the following: <ul style="list-style-type: none"> • <i>Connected</i>: Type the network interface to advertise. Additional interfaces can be added by clicking the add icon. • <i>Static</i>: Type the network prefix to advertise. Additional network prefixes can be added by clicking the add icon.
Advanced	Expand to view advanced settings, including configuration of route maps for hub overlays. You can apply the route map settings to all hub overlays or specify them individually.

7. For the *Template Options*, configure the following settings and click *Next*.

Edit SD-WAN Overlay Template - SD-WAN Template Options (4/5)

Add Overlay Objects to SD-WAN Template

Add Overlay Interfaces and Zones

Add Healthcheck Servers for Each HUB as Performance SLA

< Back

Next >

Cancel

Add Overlay Objects to SD-WAN Template	Optionally, you can toggle this setting ON to automatically add the overlay objects configured by this template to a new or existing SD-WAN template. Select an existing SD-WAN template or click the add icon to create a new SD-WAN template.
Add Overlay Interfaces and Zones	Optionally, you can toggle this setting ON to add overlay interfaces and zones.
Add Healthcheck Servers for Each HUB as Performance SLA	Optionally, you can toggle this setting ON to add health check servers for each hub as performance SLAs.

8. The summary window displays a summary of the SD-WAN overlay configurations that will be created by this template.

9. When you click *Finish*, multiple provisioning templates are created based on the information you provided. The templates are automatically assigned to the devices specified by the wizard.

Edit SD-WAN Overlay Template - Summary (5/5)

Please review the summary of SD-WAN Overlay configurations

NOTE: By clicking "Finish", multiple related provisioning templates will be automatically created based on the configurations. You could also re-run the SD-WAN Overlay wizard to re-generate the provisioning templates later.

Template Name	SD-WAN-TEMPLATE
Topology	Dual HUB (Primary & Secondary)
Region Network Settings	Loopback Allocated: 172.16.0.0/255.255.0.0 Overlay Network: 10.10.0.0/255.255.0.0 BGP AS Number: 65000 Auto-Discovery VPN: <input type="checkbox"/>
Device Assignment	Primary HUB Enterprise_HUB1 (10.100.88.101, Platform: FortiGate-VM64-KVM) Secondary HUB Enterprise_HUB2 (10.100.88.102, Platform: FortiGate-VM64-KVM) Assign to sd-wan-branches
Underlay Assignment	Primary HUB Underlays 10.0.11.2 Secondary HUB Underlays 10.0.12.3 Branch Underlays 192.185.50.1
Network Advertisement	Primary HUB Connected: None Secondary HUB Connected: None Branch Static: None
SD-WAN Template Options	Add Overlay Objects to SD-WAN Template <input type="checkbox"/> Add Overlay Interfaces and Zones <input type="checkbox"/> Add Healthcheck Servers for Each HUB as Performance SLA <input type="checkbox"/>

< Back

Finish

Cancel

10. When complete, you can deploy the SD-WAN provisioning templates in your environment.

Metafield support on dynamic objects - FMG



This information is also available in the FortiManager 7.2 Administration Guide:

- [Managing objects and dynamic objects](#)
- [Meta Fields](#)

In FortiManager 7.2.0, metadata variables can be used in dynamic objects in place of per-device mappings.

To use a metadata variable in a dynamic objects:

1. Go to *Policy & Objects > Object Configurations*.
2. Create or edit a firewall address, IP pool, or virtual IP.
3. Add the metadata in a supported text field using the following format: `$<metadata_variable_name>`.
When `$` is typed into a supported text field, available metadata variables are displayed for selection. You can click the add button to create a new metadata variable.

The screenshot shows the 'Create New Firewall Address' form. The 'Name' field is 'Branch-NET', 'Color' is '#4', 'Type' is 'Subnet', and 'Interface' is 'any'. The 'IP/Netmask' field has a search icon and a 'Resolve from name' button. A dropdown menu is open for the 'IP/Netmask' field, showing options: '(branch_id)' and '(metadata_v1)'. The 'Static Route Configuration' toggle is off.

- For firewall addresses (subnet type), you can use metadata variables in the *IP/Netmask* field.

The screenshot shows the 'Create New Firewall Address' form. The 'Name' field is 'Branch-NET', 'Color' is '#4', 'Type' is 'Subnet', and 'Interface' is 'any'. The 'IP/Netmask' field contains the value '10.1.\${branch_id}.0/24' and has a search icon and a 'Resolve from name' button. The 'Static Route Configuration' toggle is off.

- For IP pools, you can use metadata variables in the *External IP Range* field.

The screenshot shows the 'Create New IPv4 Pool' form. The 'Name' field is 'IP_pool'. The 'Comments' field is empty. The 'Configure Default Value' toggle is on. The 'Type' is 'Overload'. The 'External IP Range' field contains the value '10.1.\${branch_id}.0 - 10.1.\${branch_id}.100' and has a search icon and a 'Resolve from name' button. The 'NAT64' toggle is off. The 'Enable ARP Reply' toggle is on. The 'Advanced Options' section is expanded. The 'Per-Device Mapping' toggle is off. The 'Revision' section is expanded. The 'Change Note' field is empty.

0/1023

- For virtual IPs, you can use metadata variables in the *External IP Address/Range*, *Mapped IPv4 Address/Range*, and *Mapped IPv6 Address/Range* fields.

Create New Virtual IP

Name

VIP

Comments

Color

Interface

any

Configure Default Value

☒

Network

Type

Static NAT DNS Translation FQDN Load balance

External IP Address/Range

10.1.\${branch_id}.0 10.1.\${branch_id}.100

Mapped IPv4 Address/Range

10.2.\${branch_id}.0 10.2.\${branch_id}.100

Mapped IPv6 Address/Range

\${branch_id} \${branch_id}

Source Interface Filter

Click to select

Model device blueprints - FMG



This information is also available in the FortiManager 7.2 Administration Guide:

- [Using device blueprints for model devices](#)

In FortiManager 7.2.0, you can create device blueprints to simplify configuration of certain device settings, including device groups, configuring pre-run templates, policy packages, provisioning templates, and more. Once a device blueprint has been created, it can be selected when adding a model device or when importing multiple model devices from a CSV file.

The following information is available:

- [Creating a device blueprint on page 27](#)
- [Adding model devices using a blueprint on page 28](#)

Creating a device blueprint

To create a new device blueprint:

- Go to Device Manager, and select *Device Blueprint* from the *Add Device* dropdown menu. Previously configured blueprints are displayed in the table below and can be edited or deleted.

Device Name	Config Status	Host Name	IP Address	Platform	Description	Firmware Version
Branch_Office_01	Auto-update	Branch_Office_01	10.10.10.1	FortiGate-VM64-KVM	FortiGate 7.0.2,build0234 (GA)	
Branch_Office_02	Auto-update	Branch_Office_02	10.10.10.2	FortiGate-VM64-KVM	FortiGate 7.0.2,build0234 (GA)	
fduncan-tech72	Auto-update	fduncan-tech72	10.10.10.3	FortiGate-VM64-KVM	FortiGate 7.0.2,build0234 (GA)	
Enterprise_HUB1	Auto-update	Enterprise_First_Floor	10.10.10.4	FortiGate-VM64-KVM	FortiGate 7.0.2,build0234 (GA)	
Enterprise_HUB2	Auto-update	Enterprise_Second_Floor	10.10.10.5	FortiGate-VM64-KVM	FortiGate 7.0.2,build0234 (GA)	
vdom-1 [NAT]	Synchronized	vdom				

- Click *Create New* to add a new blueprint.
- Select the model devices to which the blueprint can be applied.
- Configure the device setting details for the blueprint. For example, you can specify a device group and provisioning template for the devices using this blueprint.

Create New Device Blueprint

Name:

Device Model:

Enforce Firmware Version: ☐ 7.0 (by default)

Add to Device Group: ☐

Add to Folder: ☐

Pre-Run CLI Template: ☐

Assign Policy Package: ☐

Provisioning Templates:

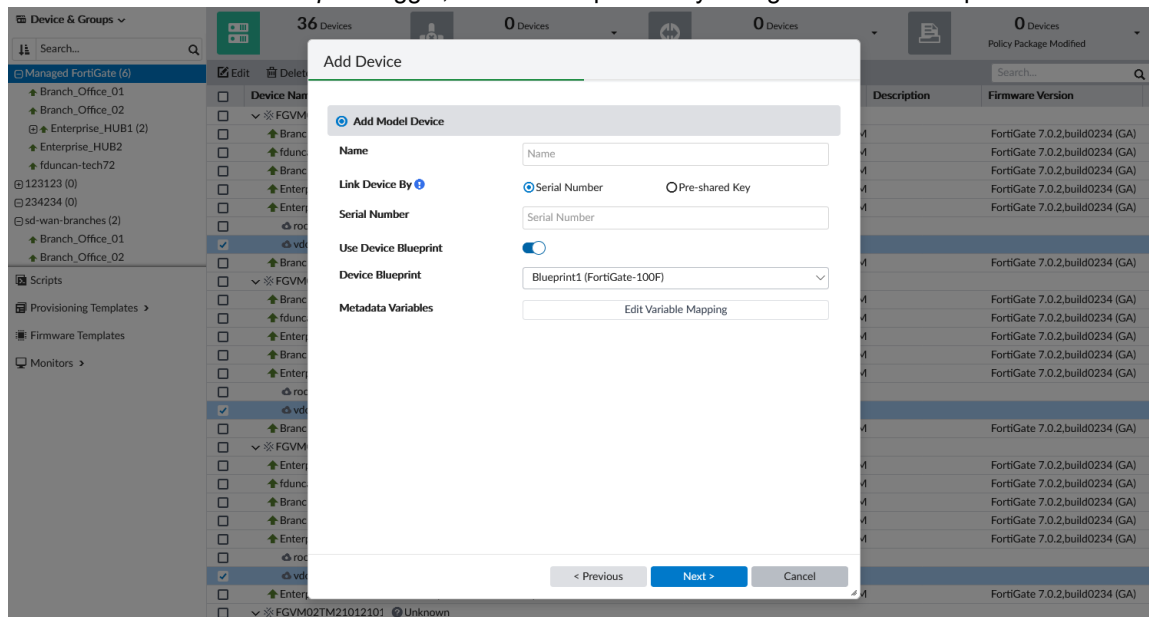
- Click *OK*.

Adding model devices using a blueprint

To use a blueprint when adding a model device:

- Go to *Device Manager > Device & Groups*.
- Click *Add Device*. The *Add Device* wizard displays.
- Click *Add Model Device*.
The *Add Device* window is displayed.
- Enter the name and serial number or pre-shared key for the device.

5. Enable the *Use Device Blueprint* toggle, and select a previously configured device blueprint.

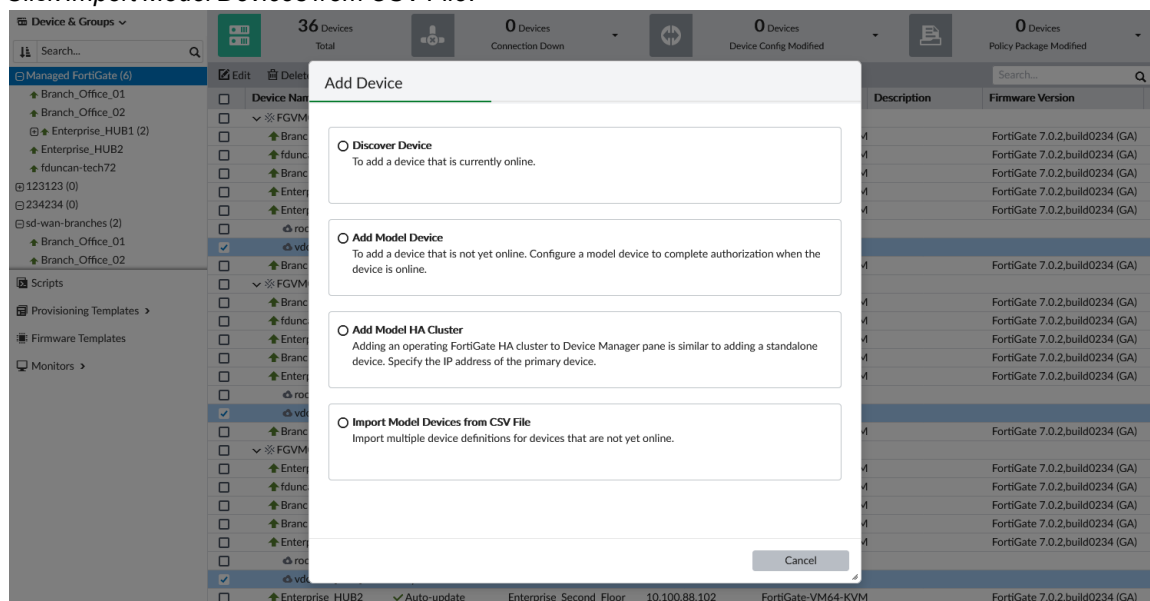


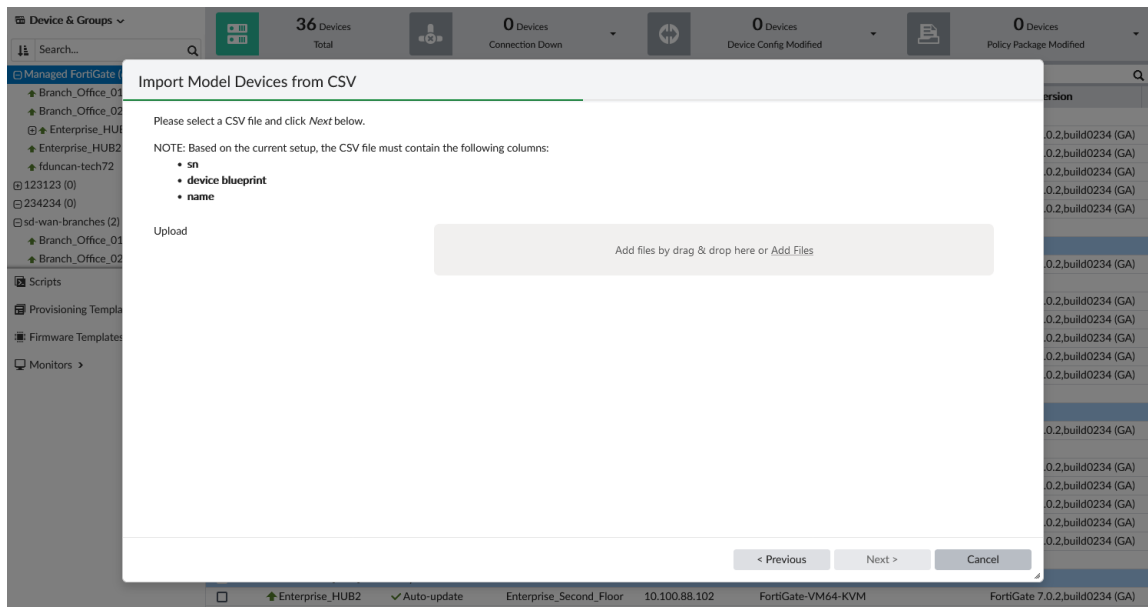
You can alternatively click the add icon to create a new device blueprint.

6. Optionally, configure the metadata variables for this device.
7. Click *Next* to continue importing the device.

To import model devices from a CSV File:

1. If ADOMs are enabled, select the ADOM to which you want to add the device.
2. Go to *Device Manager > Device & Groups*.
3. Click *Add Device*.
The Add Device window is displayed.
4. Click *Import Model Devices from CSV File*.





5. Configure your local CSV file for the devices that you want to import. CSV files must contain the following columns: `sn`, `device blueprint`, and `name`, with the respective data listed in the cells below. Additional columns can be added for each metadata variable that you want to specify. In the following image, the `branch_id` metadata variable has been added to specify this variable for each imported device.

	A	B	C	D	E	F	G	H	I
1	sn	device blueprint	name	branch_id					
2	FGVM02TM2101234	branch_blueprint	br3	3					
3	FGVM02TM2101235	branch_blueprint	br4	4					
4	FGVM02TM2101236	branch_blueprint	br5	5					
5	FGVM02TM2101237	branch_blueprint	br6	6					
6	FGVM02TM2101238	branch_blueprint	br7	7					
7	FGVM02TM2101239	branch_blueprint	br8	8					
8	FGVM02TM2101240	branch_blueprint	br9	9					
9	FGVM02TM2101241	branch_blueprint	br10	10					
10	FGVM02TM2101242	branch_blueprint	br11	11					
11	FGVM02TM2101243	branch_blueprint	br12	12					
12	FGVM02TM2101244	branch_blueprint	br13	13					
13	FGVM02TM2101245	branch_blueprint	br14	14					
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									

6. Drag and drop the CSV file into the *Upload* area, or select the CSV file location on your computer. The model devices' serial numbers, names, blueprints, and optional metadata variables are displayed in the table.
7. Review the device list, and click *Next* to begin importing the devices.
8. Click *Finish* when the import process is complete.

Application categories in SD-WAN rules - FMG



This information is also available in the FortiManager 7.2 Administration Guide:

- [SD-WAN templates](#)
- [SD-WAN rules](#)

In FortiManager 7.2.0, the *Internet Services > Application Category* option has been added when configuring SD-WAN rules.

The application category uses the default internet service database (ISDB) categories received from FortiGuard. This feature is available in a FortiManager 7.2 ADOM with 7.2 or later FortiGate devices.

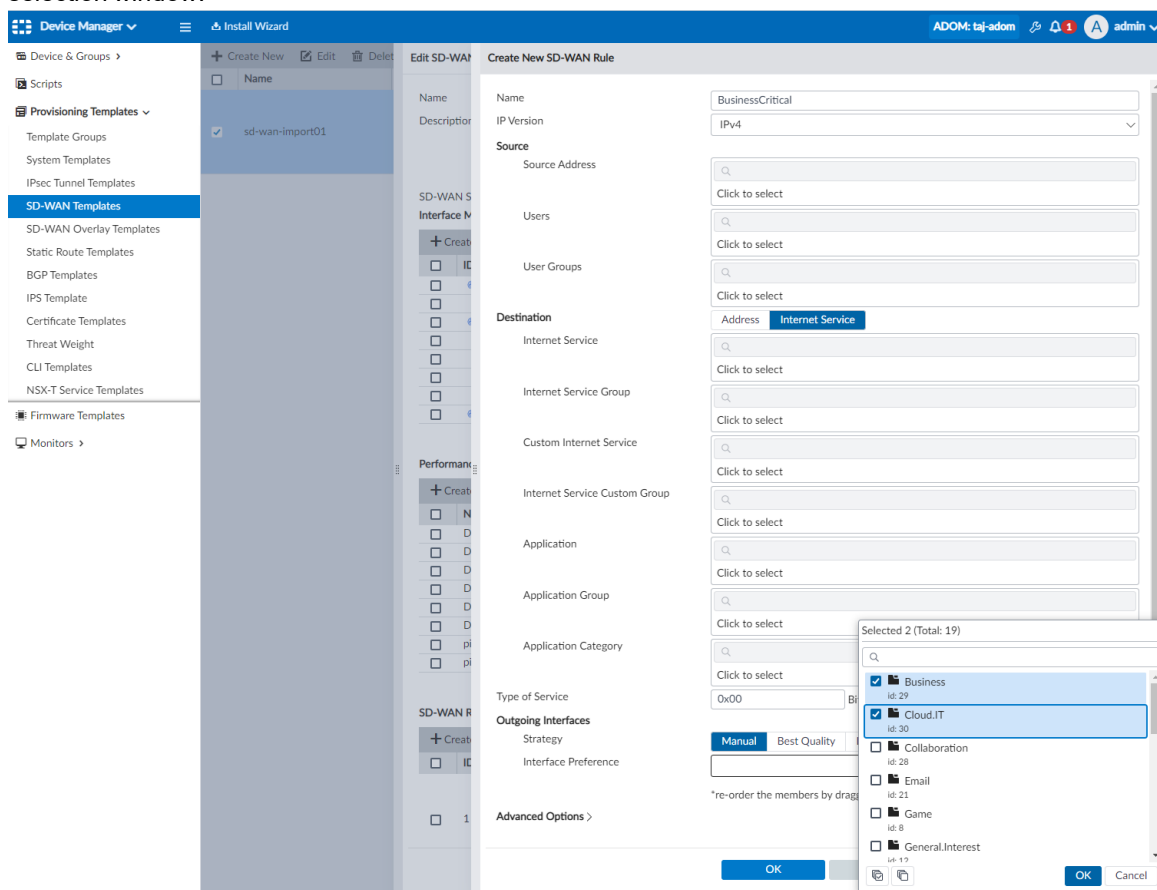
To configure application groups for SD-WAN rules in a template:

1. In FortiManager, make sure you're in a 7.2 ADOM.
2. Go to *Device Manager > Provisioning Templates > SD-WAN Templates*, and create or edit a template.
3. Under *SD-WAN Rules*, create a new rule.
4. Set the *Destination* as *Internet Service*.

The new destination type *Application Group* has been added.

The screenshot displays the FortiManager interface for configuring an SD-WAN rule. The left sidebar shows the navigation tree with 'SD-WAN Templates' selected. The main panel shows the 'Edit SD-WAN Rule' configuration for a rule named 'sd-wan-import01'. The 'Destination' section is expanded, showing 'Internet Service' selected. The 'Application Category' field is highlighted in blue. The 'Outgoing Interfaces' section shows 'Manual' selected. The 'Advanced Options' section is collapsed.

5. Select categories from the default ISDB list. New categories can be created by clicking the add button in the selection window.



6. Click OK to save the SD-WAN rule.

Performance SLA

Name	Health-Check Server	Detect Protocol	Failure Threshold	Recovery Threshold
Default_AWS	aws.amazon.com	HTTP	5	10
Default_DNS	(System DNS)	DNS	5	10
Default_FortiGuard	fortiguard.com	HTTP	5	10
Default_Gmail	gmail.com	Ping	5	10
Default_Google Search	www.google.com	HTTP	5	10
Default_Office_365	www.office.com	HTTP	5	10
ping	8.8.8.8	Ping	5	5
ping6	2004:10:100:1::1	Ping	5	5

SD-WAN Rules

ID	Name	Source	Destination	Criteria	Members
1	rule01	ALL	Microsoft-Skype_Teams Microsoft-Office365 Facebook-Whatsapp Business Cloud.IT	SLA (ping#1)	port3 port1 port2
4	BusinessCritical	ALL	Cloud.IT Business		port1 port2
	sd-wan	ALL	ALL	Volume	ALL

Neighbor

Neighbor	Role	Interface Member	Performance SLA	SLA
10.254.0.2	Standalone	port2-1	ping	1
10.254.30.1	Standalone	port2	ping	1

Duplication

ID	Packet Discard Duplication
No record found.	

Advanced Options >

OK Cancel

To configure application groups for SD-WAN rules in the device database:

1. In FortiManager, make sure you're in a 7.2 ADOM.
2. Go to *Device Manager* > *Device & Groups*.
3. Select a FortiGate device (7.2 or later) to manage the device database.
4. Go to *System* > *SD-WAN* > *SD-WAN Rules*, and create a new rule.

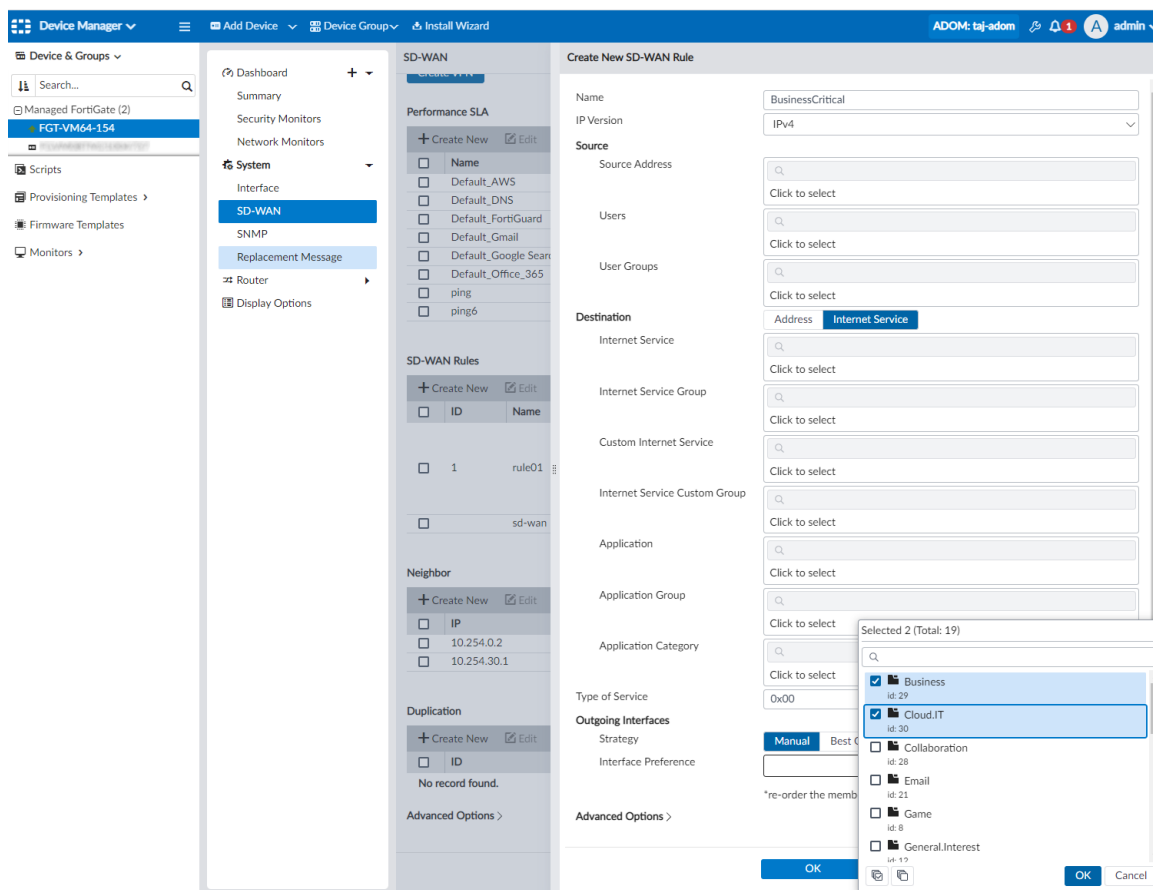
5. Set the *Destination* as Internet Service.
The new destination type *Application Group* has been added.

The screenshot displays the FortiManager interface for configuring SD-WAN rules. The left sidebar shows the navigation menu with 'SD-WAN' selected under the 'System' category. The main panel is titled 'Create New SD-WAN Rule' and contains the following sections:

- Name:** BusinessCritical
- IP Version:** IPv4
- Source:**
 - Source Address: Click to select
 - Users: Click to select
 - User Groups: Click to select
- Destination:**
 - Address: Internet Service (selected)
 - Internet Service: Click to select
 - Internet Service Group: Click to select
 - Custom Internet Service: Click to select
 - Internet Service Custom Group: Click to select
- Application:** Click to select
- Application Group:** Click to select
- Application Category:** Click to select
- Type of Service:** 0x00 (selected) | Bit Mask: 0x00
- Outgoing Interfaces:**
 - Strategy: Manual (selected), Best Quality, Lowest Cost (SLA), Maximize Bandwidth (SLA)
 - Interface Preference: +
- Advanced Options:** (expandable section)

At the bottom of the dialog are 'OK' and 'Cancel' buttons. A note at the bottom states: '*re-order the members by dragging and dropping the item'.

6. Select categories from the default ISDB list. New categories can be created by clicking the add button in the selection window.



7. Click OK to save the SD-WAN rule.

Performance SLA

Name	Health-Check Server	Detect Protocol	Failure Threshold	Recovery Threshold
Default_AWS	aws.amazon.com	HTTP	5	10
Default_DNS	96.45.45.45, 0.0.0.0 (System DNS)	DNS	5	10
Default_FortiGuard	fortiguard.com	HTTP	5	10
Default_Gmail	gmail.com	Ping	5	10
Default_Google Search	www.google.com	HTTP	5	10
Default_Office_365	www.office.com	HTTP	5	10
ping	8.8.8.8	Ping	5	5
ping6	2004:10:100:1::1	Ping	5	5

SD-WAN Rules

ID	Name	Source	Destination	Criteria	Members
1	rule01	ALL	Microsoft-Skype_Teams Microsoft-Office365 Facebook-Whatsapp Business Cloud.IT	SLA (ping#1)	port3 port1 (wan1) port2 (wan2)
2	BusinessCritical	ALL	Cloud.IT Business		port1 (wan1) port2 (wan2)
	sd-wan	ALL	ALL	Volume	ALL

Neighbor

IP	Role	Interface Member	Performance SLA	SLA
10.254.0.2	Standalone	port2-1	ping	1
10.254.30.1	Standalone	port2 (wan2)	ping	1

Duplication

ID	Packet Discard Duplication
No record found.	

Advanced Options >

Apply

Fabric Authorization Template automatically provisions and authorizes LAN Edge devices on managed FortiGates - FMG 7.2.1



This information is also available in the FortiManager 7.2 Administration Guide:

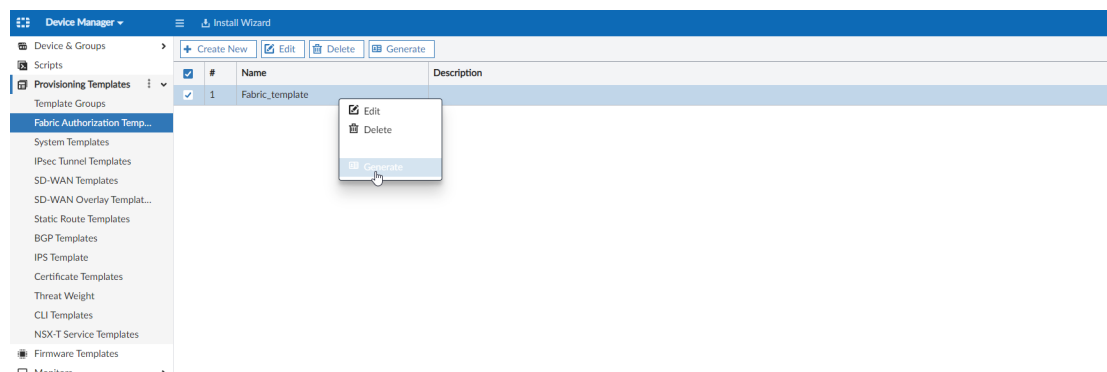
- [Fabric authorization templates](#)

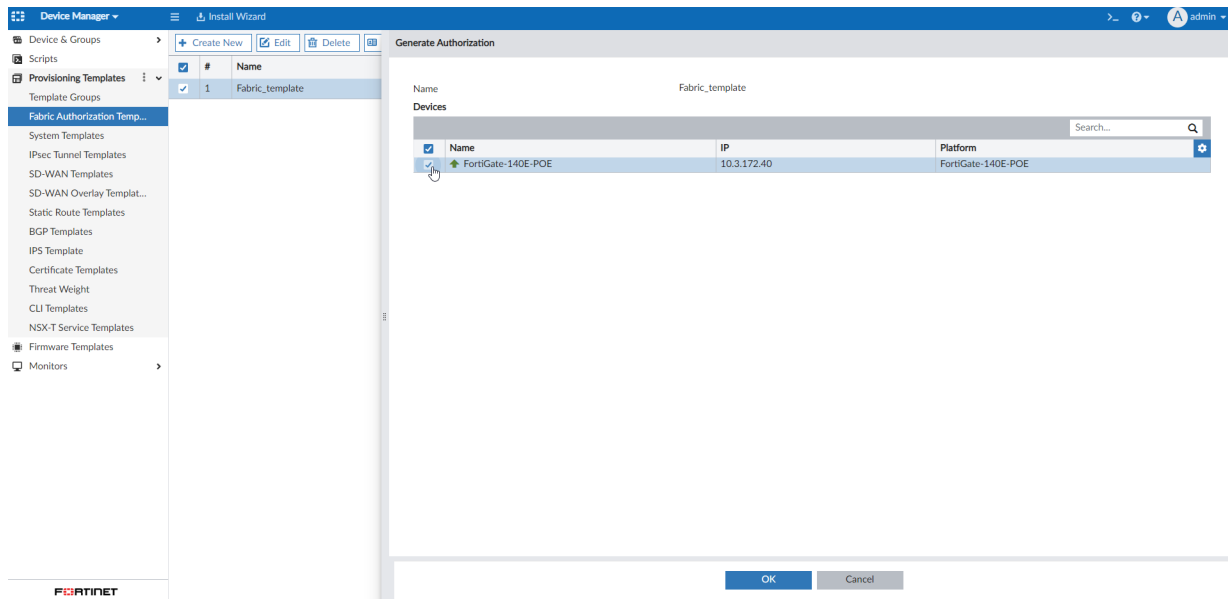
Fabric Authorization Template automatically provisions and authorizes LAN Edge devices on managed FortiGates. Within the template we can enable the wireless and switch controllers and configure FortiLink interfaces.

To configure the Fabric Authorization Template:

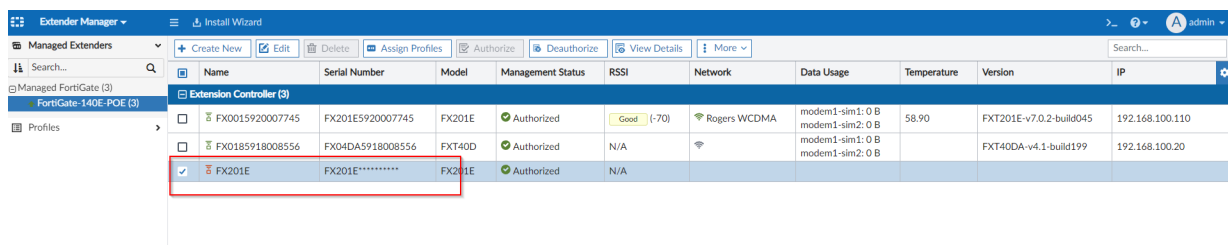
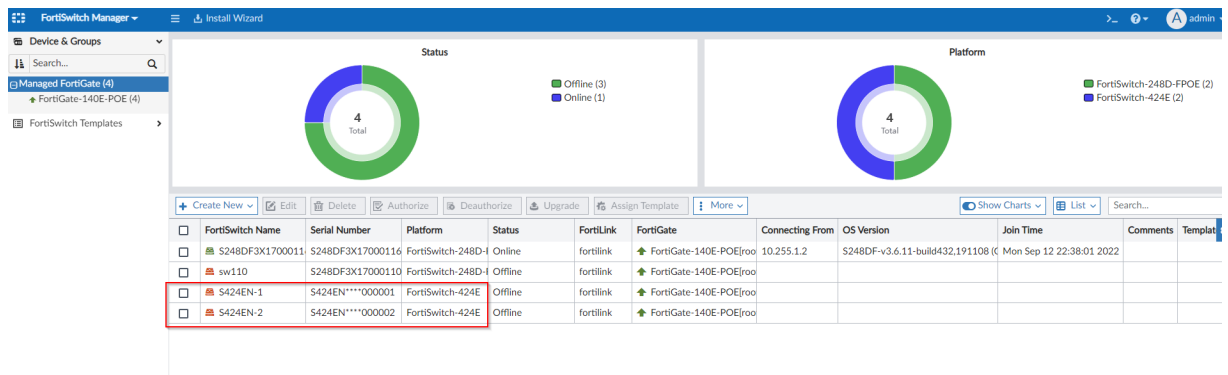
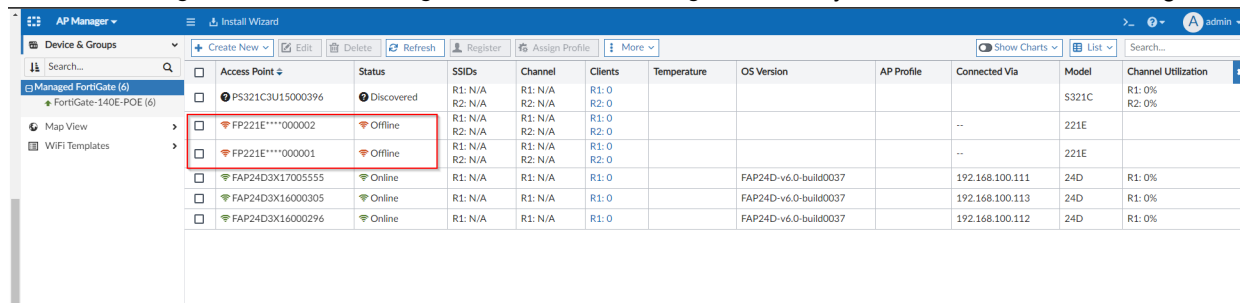
1. Go to *Device Manager > Provisioning Templates > Fabric Authorization Template*.
2. Create a new template, and specify the FortiAP, FortiSwitch, and/or FortiExtender settings, then save the template.

3. Right-click the template, and select *Generate* from the context menu, and then select the FortiGate to generate the wildcard entries.

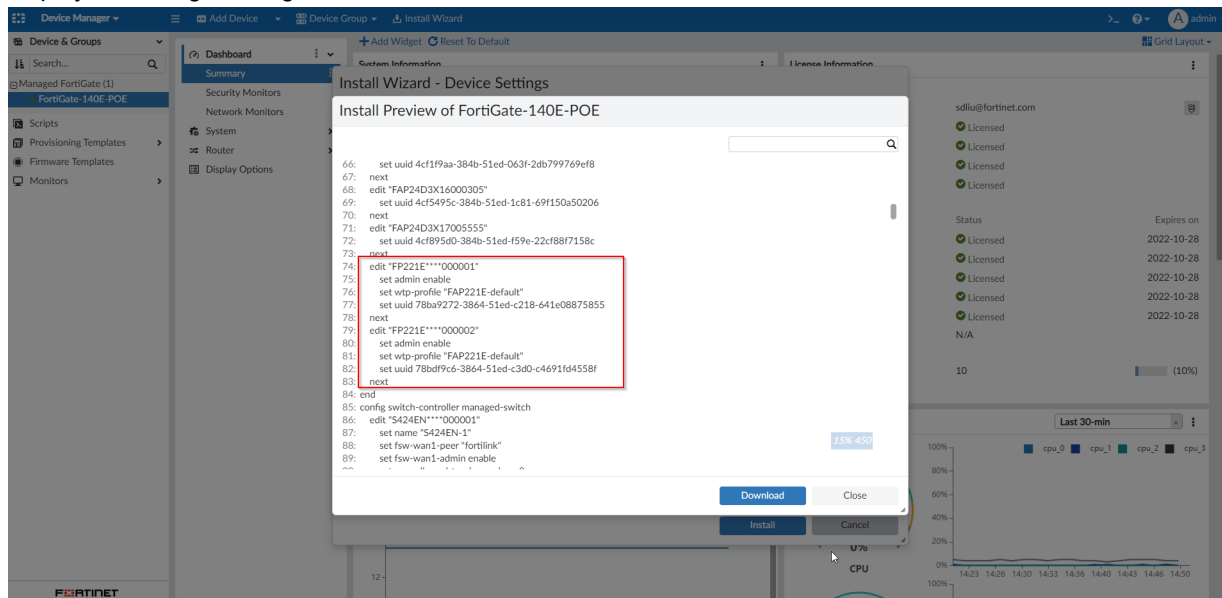




4. Go to AP Manager, FortiSwitch Manager, and Extender Manager, and verify that the wildcard entries are generated.



5. Deploy the changes using the Install Wizard.



Internet service database version checked for model devices - FMG 7.2.1



This information is also available in the FortiManager 7.2 Administration Guide:

- [Model devices](#)

For model devices, FortiManager checks the ISDB (internet service database) version on FortiGates before installing the configuration to the FortiGate. When the ISDB version on the FortiGate is older than the ISDB version on FortiManager, FortiManager triggers an ISDB upgrade on FortiGate before installing the configuration.

If the ISDB version is not updated after three minutes, FortiManager still installs the configuration to FortiGate.

You can observe the behavior in Task Monitor.

To observe ISDB upgrade behavior:

1. Go to *System Settings > Task Monitor*.
2. Select the *Install Configuration* task, and click *View Details*. The details are displayed.
3. Select a detail, and click *View Progress Report*.

The following example shows the internet service database version being updated:

View Progress Report	
Search...	
Name	Status
fg11	start to install dev (fg11)
fg11	start to update internet service from version 0.0 to 7.2803
fg11	FDS objects on device are updated, current version: 7.2803
fg11	init state: start to get pre-checksum
fg11	get pre-checksum state: start get diff (chkout=0)
fg11	script done state: start to FGFM install
fg11	fgfm install state: prepare to post-checksum
fg11	post-checksum state: start verification
fg11	install and save finished status=OK

If the update to the internet service database fails, the configuration installation still proceeds, for example:

Task 54: Push config to device. 80%

Total: 0/1, Pending: 0, In Progress: 1, Completed: 0

View Installation Log View Progress Report Column Settings

#	Name	Time Used	Status
1	49	3m 40s	80%

View Progress Report

Name	Progress %	Time Used	Status
49	0%	<1s	start to install dev (49)
49	5%	2s	start to update internet service from version 0.0 to 7.2429
49	10%	3m 0s	FDS update on device FGVM08TM21004801 is time-out
49	15%	<1s	init state: start to get pre-checksum
49	25%	5s	get pre-checksum state: start get diff (chkout=0)

Close

4. Click *Close*.

FortiManager supports BYOL installation on managed FortiGate VMs - FMG 7.2.1



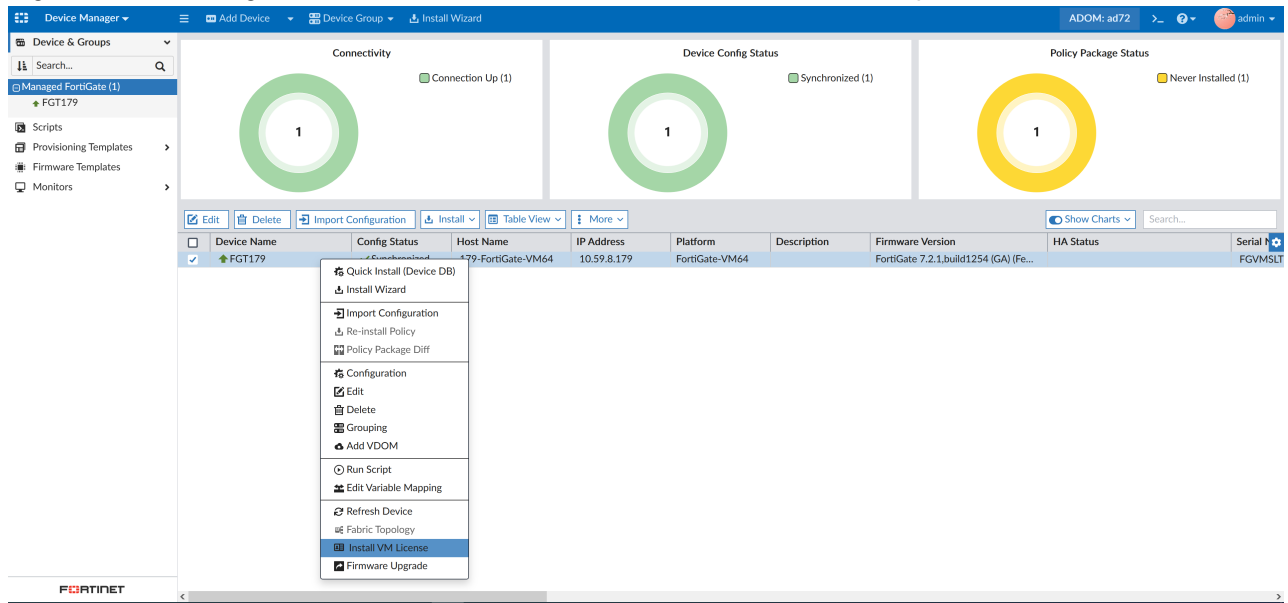
This information is also available in the FortiManager 7.2 Administration Guide:

- [Installing VM licenses](#)

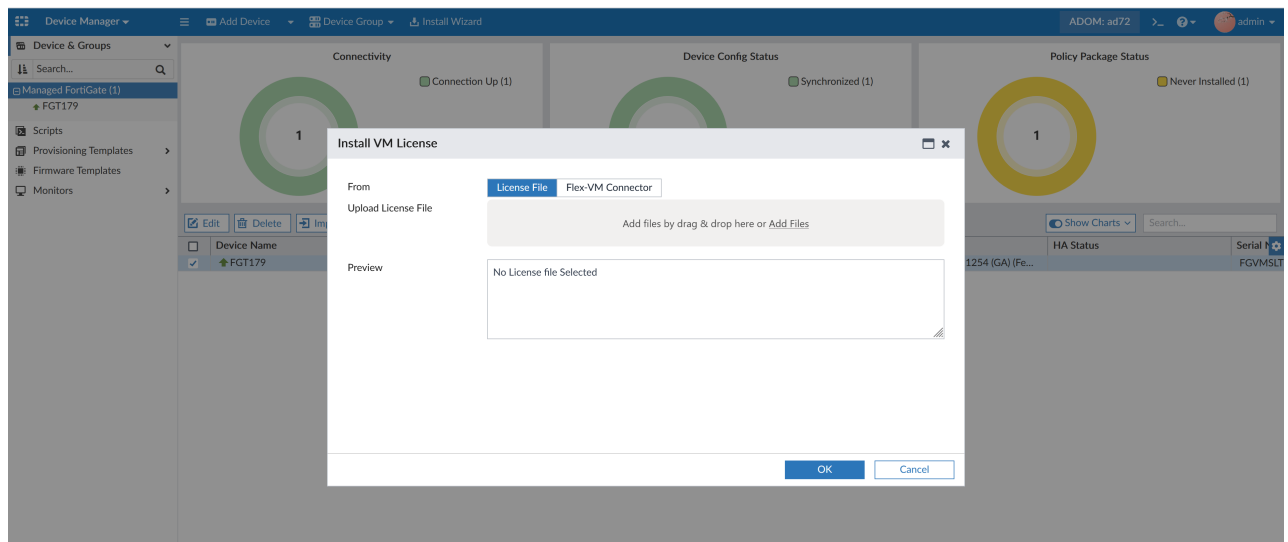
FortiManager supports BYOL installation on managed FortiGate VMs.

To install a BYOL license to a managed FortiGate VM on FortiManager:

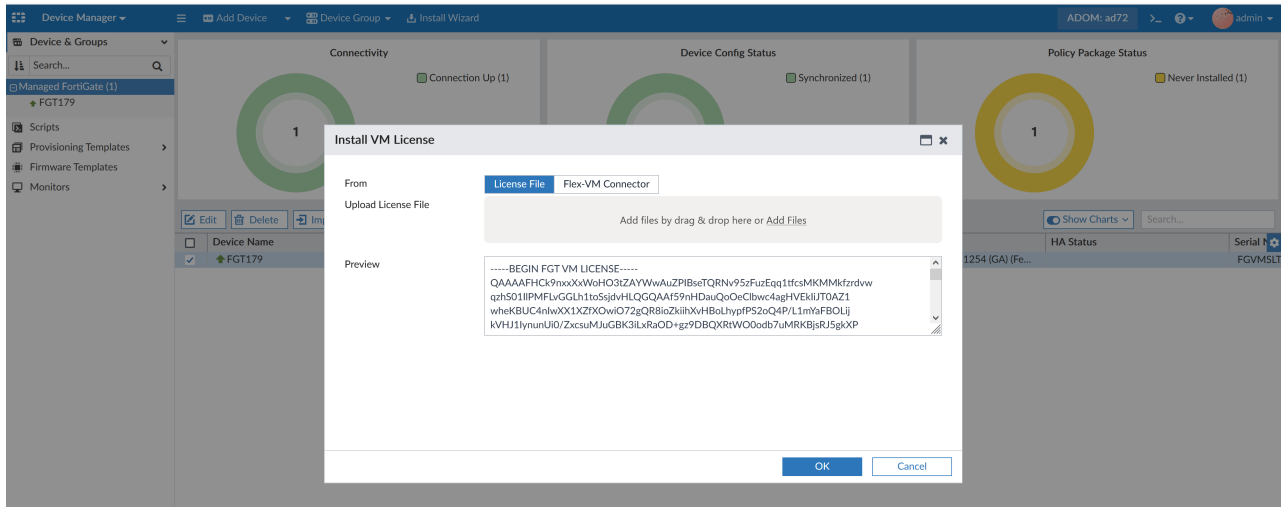
1. Go to *Device Manager > Device & Groups*.
2. Right click on a managed FortiGate VM, and select *Install VM License* from the dropdown menu.



The Install VM License dialog appears with two options for the license: *License File* and *Flex-VM Connector*.

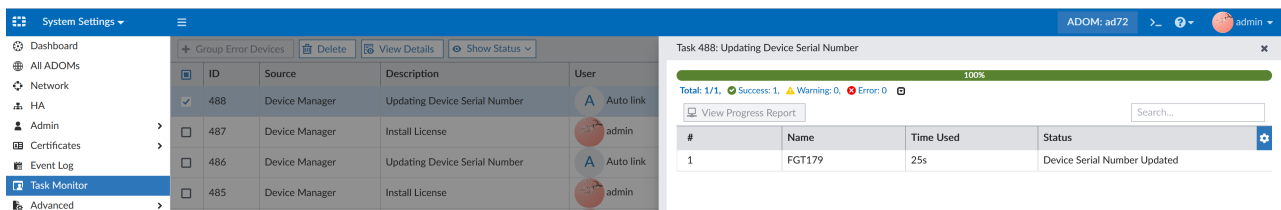
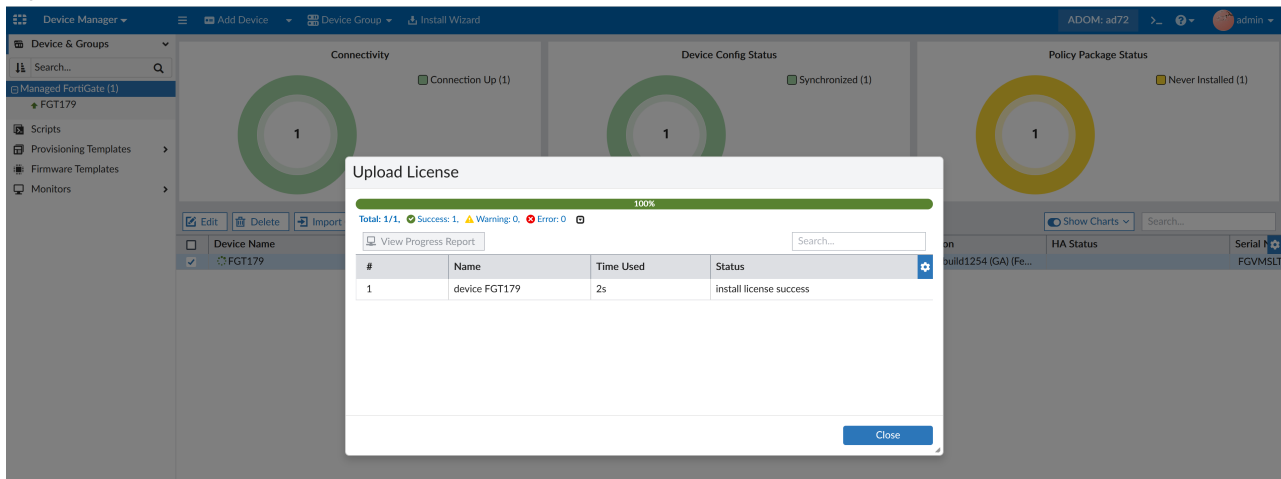


3. Select *License File*, and then upload the license by dragging and dropping the file into the selection box, or clicking *Add Files* to browse to its location.



4. Click OK.

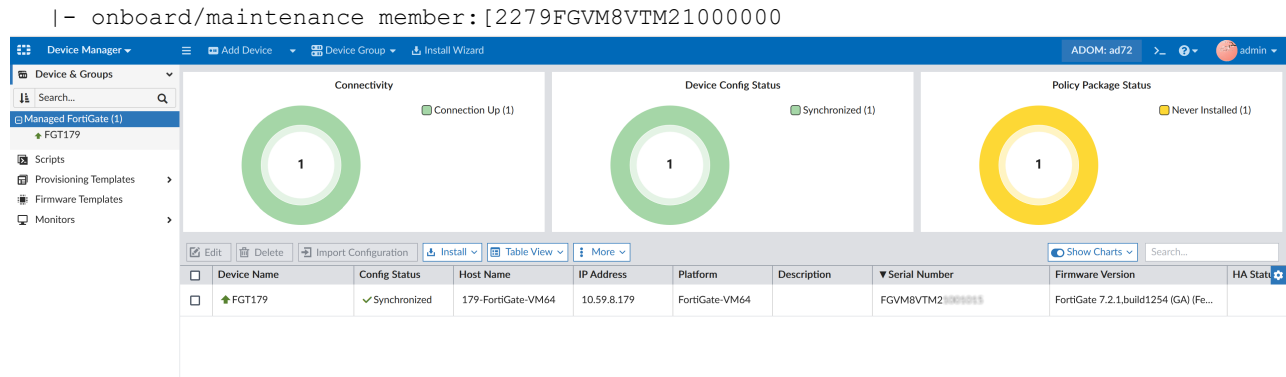
FortiManager uploads the license and updates the Serial Number for the FortiGate device. The FortiGate license is replaced with the new license.



Check the device list, onboard/maintenance member for new the FortiGate serial number.

For example:

```
diagnose dvm device list FGT179
--- There are currently 49 devices/vdoms managed ---
--- There are currently 42 devices/vdoms count for license ---
TYPE OID SN HA IP NAME ADOM
IPS FIRMWARE
fmgfaz-managed 2267 FGVMSLTm2200001 - 10.59.8.179 179-FortiGate-VM64 ad72
6.00741 (extended) 7.0 MR2 (1254)
|- STATUS: dev-db: not modified; conf: in sync; cond: OK; dm: retrieved; conn: up
|- vdom:[3]root flags:0 adom:ad72 pkg:[never-installed]
```



Pre-built route-maps used for SD-WAN self-healing with BGP routing - FMG 7.2.2



This information is also available in the FortiManager 7.2 Administration Guide:

- [Using preconfigured route maps for self-healing with BGP](#)

FortiManager 7.2.2 includes pre-built route-maps used for SD-WAN self-healing with BGP routing.

name	comments
In-SLA	
Out-SLA	
Priority_1	
Priority_2	
Priority_3	
Priority_4	
Priority_9999	
RM-VPN-Priority	
port3_only	

An option is available in the *SD-WAN Overlay Template* to automatically configure BGP neighbors based on HUB overlays and SLAs created by the overlay template.

The branch includes five (5) preconfigured route maps that the user may select, including: *Priority_1*, *Priority_2*, *Priority_3*, *Priority_4*, *Priority_9999* (used as a catch-all), and *RM-VPN-Priority*. Each route map will advertise a given community based on the *SD-WAN Overlay Template AS*.

Top Screenshot: Edit SD-WAN Overlay Template - Network Configuration (3/5)

The left sidebar shows the navigation menu with 'Provisioning Templates' expanded. The main area shows the configuration for the 'sd-wan-overlay' template. A dropdown menu is open for 'Priority_9999' under the 'Advanced' section.

Bottom Screenshot: Edit router route-map

The left sidebar shows the navigation menu with 'Policy & Objects' expanded. The main area shows the configuration for the 'route-map' object. The 'rule' section contains a table of route map rules.

id	action	match-as-path	match-community	match-community-exact	match-flags
1	permit			disable	0

Each HUB maps the route map to a priority. Established by the advertised community from the branch (based on the SLA information), the priority value will decide the preferred routing.

Device Manager | **Install Wizard** | ADOM: fgt72-overlay | admin

Provisioning Templates

- Template Groups
- Fabric Authorization Temp...
- System Templates
- IPsec Tunnel Templates
- SD-WAN Templates
- SD-WAN Overlay Templat...**
- Static Route Templates
- BGP Templates
- IPS Template
- Certificate Templates
- Threat Weight
- CLI Templates
- NSX-T Service Templates

Firmware Templates

- Monitors

Edit SD-WAN Overlay Template - Network Configuration (3/5)

Name: sd-wan-overlay

HUB

Standalone HUB: HUB3

WAN Underlay 1: ☐ Private Link ☐ Override IP ☐ Private Link ☐ Override IP

WAN Underlay 2: ☐ Private Link ☐ Override IP ☐ Private Link ☐ Override IP

Network Advertisement: **Connected** Static

Interface 1: port3

Advanced

Branch Route Maps

Route map in: ☒ RM-VPN-Priority

Route map out: ☐ RM-VPN-Priority

Branch

Branch Device Group: WAN Underlay 1, WAN Underlay 2

Network Advertisement: Interface 1

Search Results:

- port3_only
- Priority_1
- Priority_2
- Priority_3
- Priority_4
- Priority_9999
- ☒ RM-VPN-Priority

Policy & Objects | **Policy Package** | **Install Wizard** | ADOM: fgt72-overlay | admin

Object Configurations

- Normalized Interface
- Firewall Objects
- Security Profiles
- Fabric Connectors
- User & Authentication
- WAN Optimize
- Dynamic Object
- Advanced
- CLI Configurations
- Objects**

Edit router route-map

name: RM-VPN-Priority (maximum 35 characters)

comments: (maximum 127 characters)

rule

Table:

id	action	match-as-path	match-community	match-community-exact	match-flags
1	permit		Priority_1	disable	0
2	permit		Priority_2	disable	0
3	permit		Priority_3	disable	0

OK **Cancel**

FortiManager supports multiple interface members in the SD-WAN neighbor configuration - FMG 7.2.2



This information is also available in the FortiManager 7.2 Administration Guide:

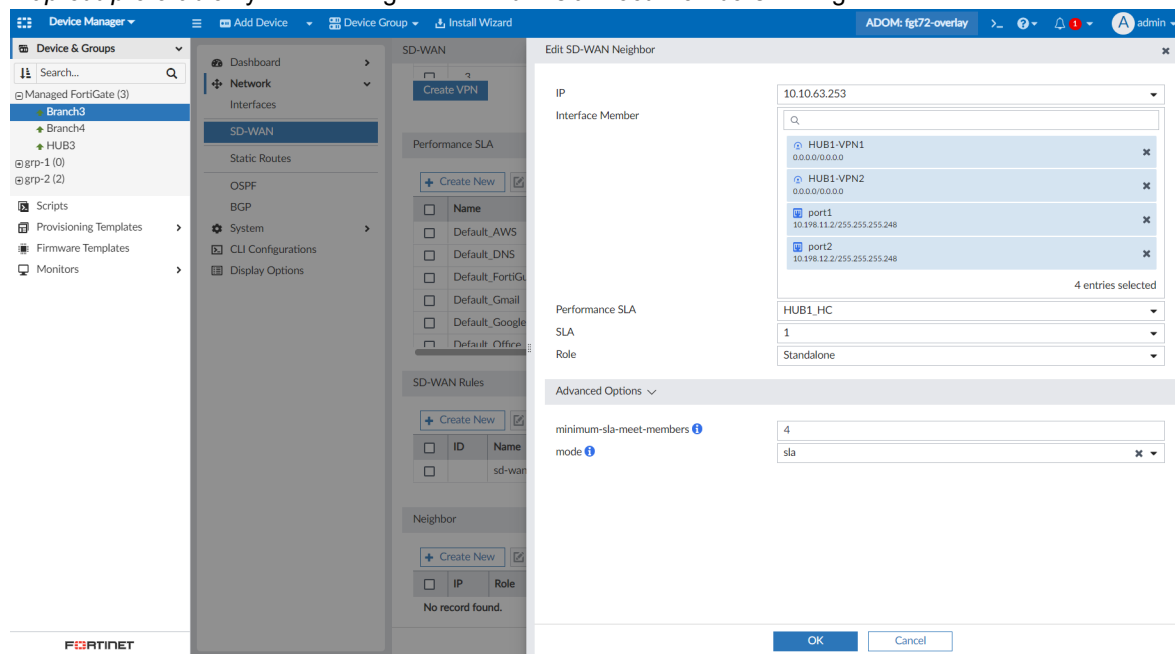
- [Neighbors](#)

FortiManager supports multiple interface members in the SD-WAN neighbor configuration.

This setting can be configured per-device or using an SD-WAN Templates.

To configure per-device:

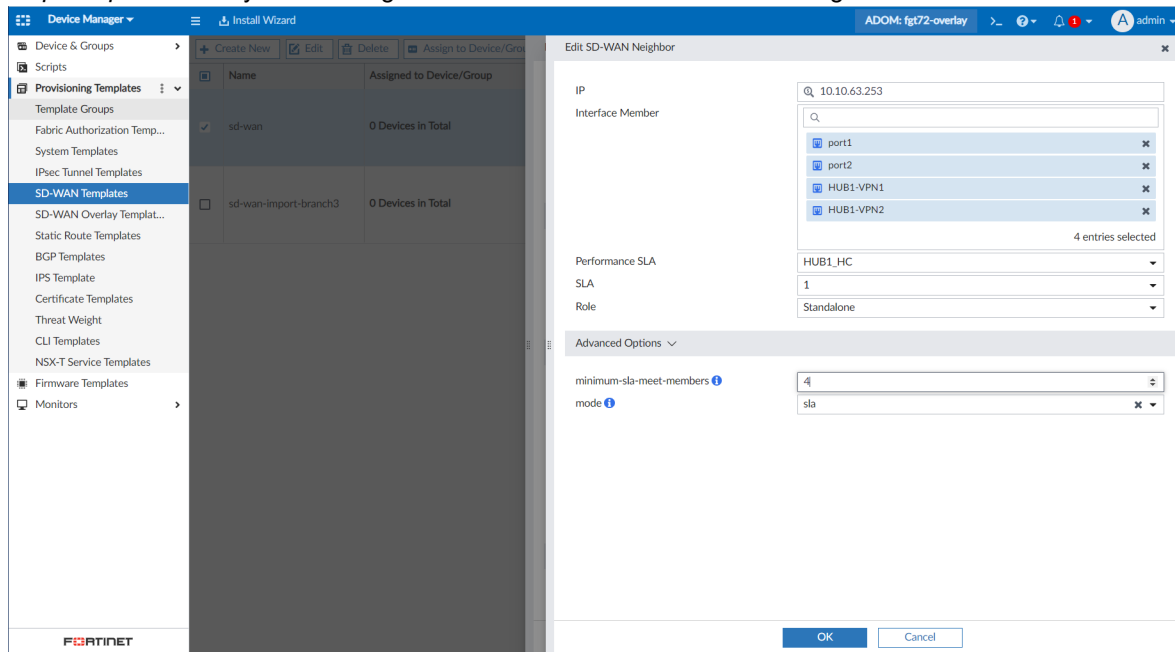
1. In a FortiManager 7.2 ADOM, go to *Device Manager > Managed Devices*, and select a managed device.
2. In the device database, go to *Network > SD-WAN*, and create or edit a *Neighbor*. Multiple *Interface Members* can be configured.
3. Open *Advanced Options*. You can configure the minimum number of members that must be in SLA to utilize *route-map-out-preferable* by customizing the *minimum-sla-meet-members* setting.



To configure using an SD-WAN Template:

1. In a FortiManager 7.2 ADOM, go to *SD-WAN Templates*, and edit or create a template.
2. Edit an *SD-WAN Neighbor*. Multiple *Interface Members* can be configured.

3. Open *Advanced Options*. You can configure the minimum number of members that must be in SLA to utilize *route-map-out-preferable* by customizing the *minimum-sla-meet-members* setting.



Factory default firewall addresses and address group for private IP space (RFC1918) - FMG 7.2.2



This information is also available in the FortiManager 7.2 Administration Guide:

- [Default address space objects](#)

FortiManager includes factory default firewall addresses and address group for private IP space (RFC1918).

The following new default firewall addresses objects are available:

- **RFC1918-10**: 10.0.0/8
- **RFC1918-172**: 172.16.0.0/12
- **RFC1918-192**: 192.168.0.0/16

The following new default firewall address group is available:

- **RFC1918-GRP**: Includes the *RFC1918-10*, *RFC1918-172*, and *RFC1918-192* address objects.

To use the new default private IP space address objects in FortiManager:

1. Go to *Policy & Objects > Object Configurations > Firewall Objects > Addresses*.
The default RFC1918 address objects are available.

Policy & Objects	Policy Packages	Install Wizard	ADOM Revisions	Tools	ADOM: root	admin
Policy Packages	Create New	Columns	More	Column Settings		
Object Configurations	Name	Type	Details	Interface	Comments	Created Time
Normalized Interface	none	Firewall Address	IP Netmask: 0.0.0.0/255.255.255.255	any		admin / 2022-08-05 11
Firewall Objects	login.microsoftonline.com	Firewall Address	FQDN:login.microsoftonline.com	any		admin / 2022-08-05 11
Addresses	login.microsoft.com	Firewall Address	FQDN:login.microsoft.com	any		admin / 2022-08-05 11
Wildcard FQDN Addresses	login.windows.net	Firewall Address	FQDN:login.windows.net	any		admin / 2022-08-05 11
Services	gmail.com	Firewall Address	FQDN:gmail.com	any		admin / 2022-08-05 11
Schedules	wildcard.google.com	Firewall Address	FQDN:*.google.com	any		admin / 2022-08-05 11
Virtual IPs	wildcard.dropbox.com	Firewall Address	FQDN:*.dropbox.com	any		admin / 2022-08-05 11
IP Pools	SSLVPN_TUNNEL_ADDR1	Firewall Address	IP Range: 10.212.134.200-10.212.134.210	any		admin / 2022-08-05 11
Traffic Shapers	all	Firewall Address	IP Netmask: 0.0.0.0/0.0.0.0	any		admin / 2022-08-05 11
Shaping Profile	FIREWALL_AUTH_PORTAL_ADDRESS	Firewall Address	IP Netmask: 0.0.0.0/0.0.0.0	any		admin / 2022-08-05 11
Security Profiles	FABRIC_DEVICE	Firewall Address	IP Netmask: 0.0.0.0/0.0.0.0	any	IPv4 addresses of Fabric Devices.	admin / 2022-08-05 11
Fabric Connections	metadata-server	Firewall Address	IP Netmask: 169.254.169.254/255.255.255.255	any		admin / 2022-08-05 11
User & Authentication	RFC1918-10	Firewall Address	IP Netmask: 10.0.0.0/255.0.0.0	any		admin / 2022-08-05 11
	RFC1918-172	Firewall Address	IP Netmask: 172.16.0.0/255.240.0.0	any		admin / 2022-08-05 11
	RFC1918-192	Firewall Address	IP Netmask: 192.168.0.0/255.255.0.0	any		admin / 2022-08-05 11
	G Suite	Address Group	gmail.com, wildcard.google.com			admin / 2022-08-05 11
	Microsoft Office 365	Address Group	login.microsoftonline.com, login.microsoft.co			admin / 2022-08-05 11
	RFC1918-GRP	Address Group	RFC1918-10, RFC1918-172, RFC1918-192			admin / 2022-08-05 11
	SSLVPN_TUNNEL_IPV6_ADDR1	IPv6 Address	IPv6 Subnet: ffff:ffff::120			admin / 2022-08-05 11
	all	IPv6 Address	IPv6 Subnet: ::0			admin / 2022-08-05 11
	none	IPv6 Address	IPv6 Subnet: ::120			admin / 2022-08-05 11
	IPv6-address	Proxy Address	Host Regexp Match: *[0-9a-f]{0,4}:[0-9a-f]{0,4}:[0-9a-f]{0,4}:[0-9a-f]{0,4}			admin / 2022-08-05 11
	IPv4-address	Proxy Address	Host Regexp Match: *[0-9]{1,3}:[0-9]{1,3}:[0-9]{1,3}:[0-9]{1,3}			admin / 2022-08-05 11

2. Go to **Policy & Objects > Policy Packages**, and select a **Firewall Policy**.

You can select the firewall address objects for use in the policy. For example, the RFC1918-GRP address group object is selectable as an IPv4 Destination Address.

3. Install the policy package to FortiGate.

To edit the default private IP space address objects using the CLI:

1. In the FortiManager CLI, use the config firewall address command.

For example:

```
config firewall address
edit "RFC1918-10"
set subnet 10.0.0.0 255.0.0.0
next
edit "RFC1918-172"
set subnet 172.16.0.0 255.240.0.0
next
edit "RFC1918-192"
set subnet 192.168.0.0 255.255.0.0
next
end
config firewall addrgrp
edit "RFC1918-GRP"
set member "RFC1918-10" "RFC1918-172" "RFC1918-192"
next
```

end

Interface-based traffic shaping can display real time dropped packets - FMG 7.2.2



This information is also available in the FortiManager 7.2 Administration Guide:

- [Viewing the traffic shaping widget](#)

Interface-based traffic shaping can display real time dropped packets.

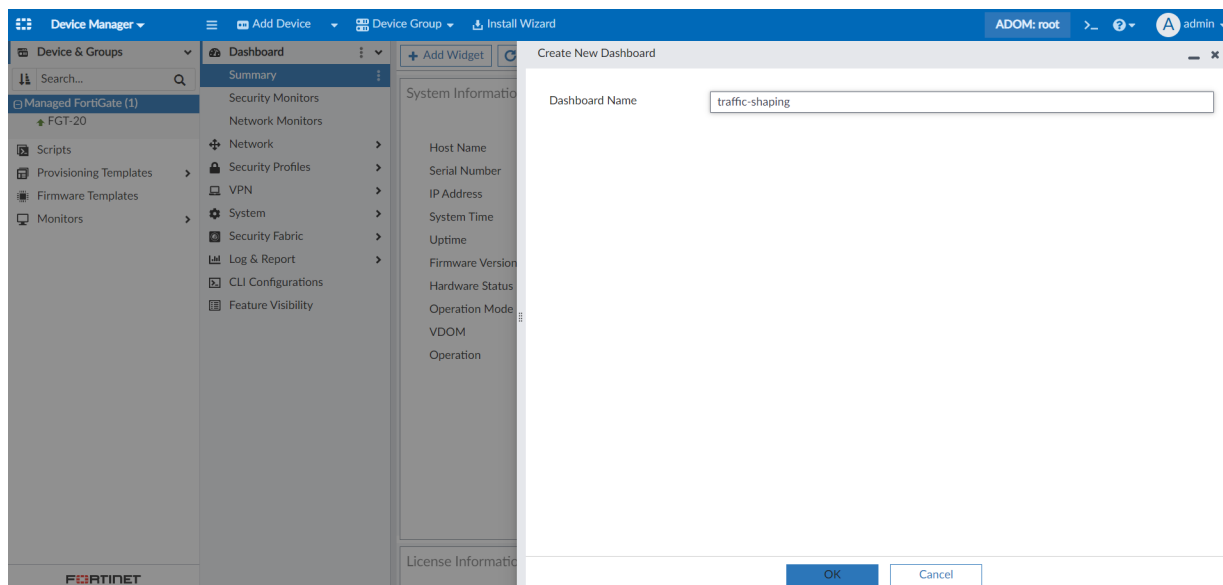
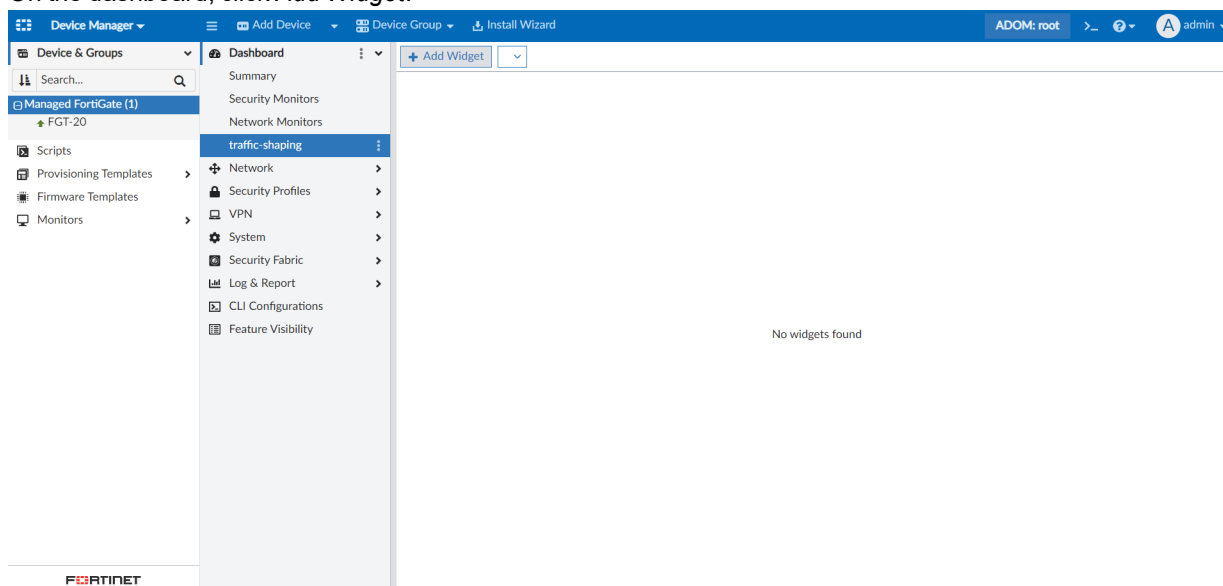
To view real-time dropped packets in the Traffic Shaping widget:

1. Go to *Device Manager > Device Groups*, and select a managed device.

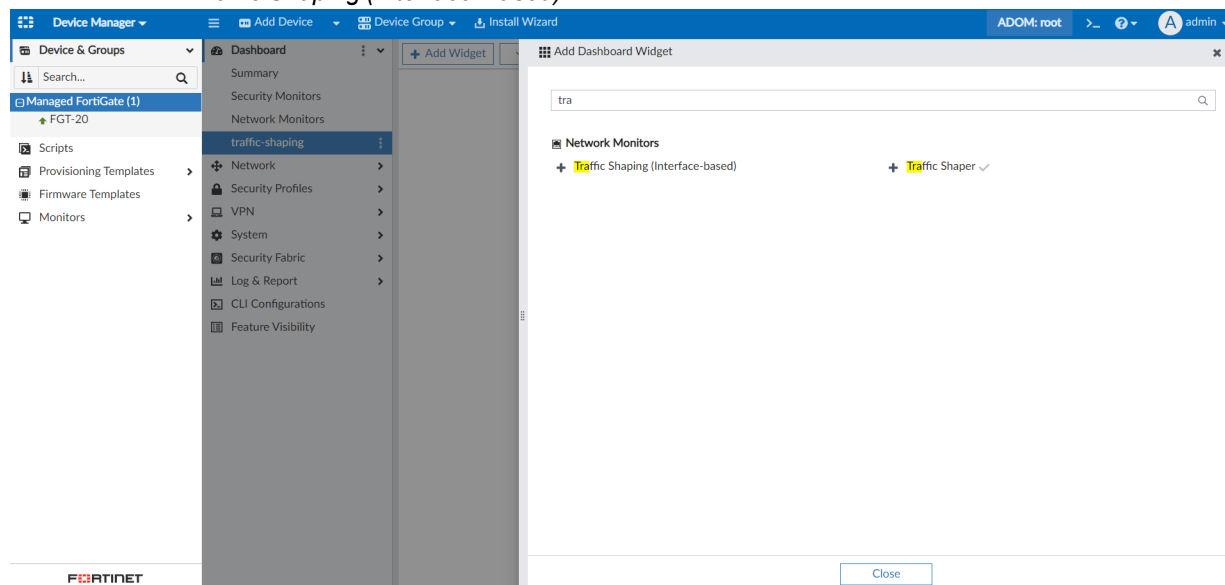
Device Name	Config Status	Host Name	IP Address	Platform	Description	Firmware Version	FGSP
FGT-20	✓ Synchronized	FGT-20	10.2.170.20	FortiGate-VM64	FortiGate 7.2.4.build1366 (Interim)	FortiGate 7.2.4.build1366 (Interim)	Disabled

2. In the toolbar, click *Create New*.

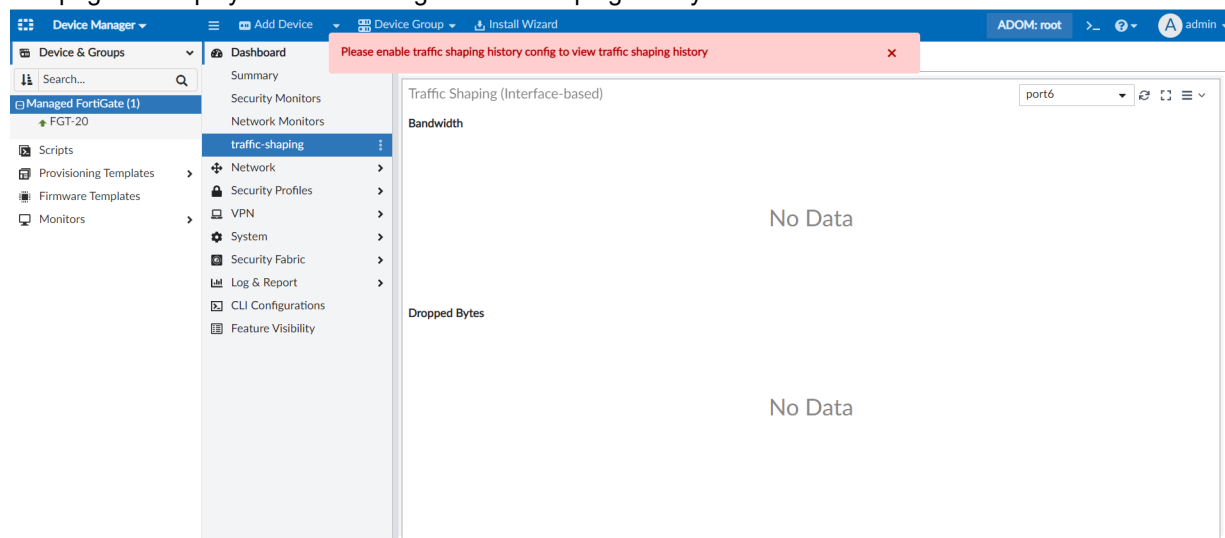
Host Name	Serial Number	IP Address	System Time	Uptime	Firmware Version	Hardware Status	Operation Mode	VDOM	Operation
FGT-20	FGT-20	10.2.170.20 (port1)	Thu Dec 01 11:37:18 2022 PST	2 days 1 hour 2 minutes 2 seconds	FortiGate 7.2.4.build1366 (Interim)	1 CPU, 1994 MB RAM	NAT	VDOM Disabled	> < < >

3. Enter a name for the dashboard.**4. On the dashboard, click *Add Widget*.**

5. Search and add *Traffic Shaping (Interface-Based)*.



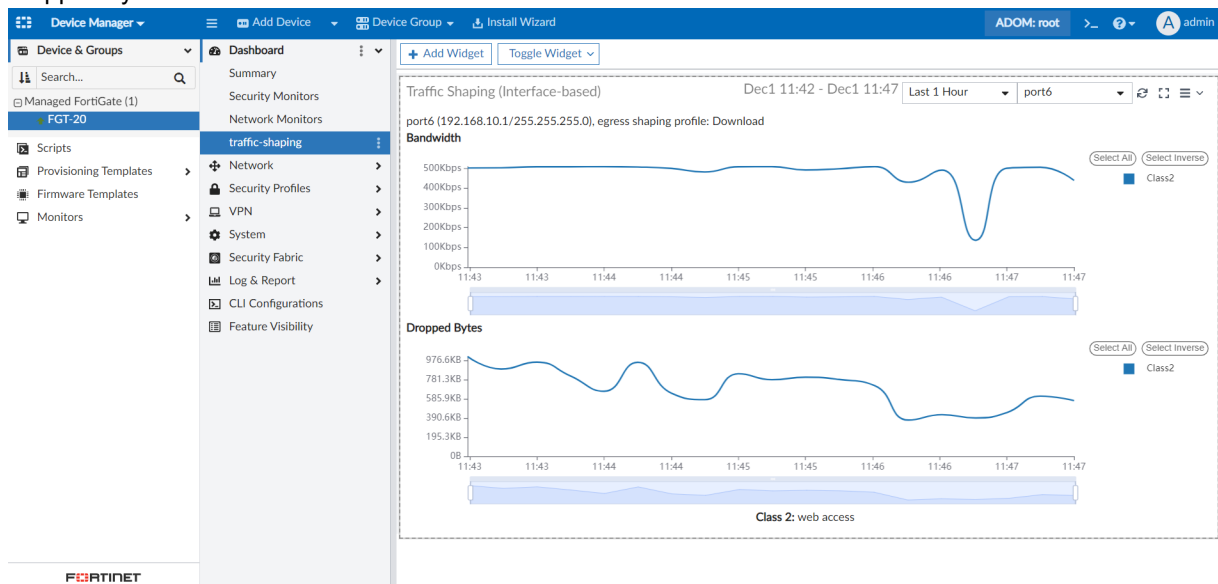
The page will display an error message if traffic shaping history is not enabled.



6. To enable traffic shaping history, open the CLI console and enter the following commands:

```
config system admin setting
  set traffic-shaping-history enable
end
```

7. After FortiManager receives data from FortiGate, the widget will display the real-time information of bandwidth and dropped bytes for each class.



Add Fabric Overlay Orchestrator for SD-WAN overlay configurations - 7.2.4



This information is also available in the FortiOS 7.2 Administration Guide:

- [Fabric Overlay Orchestrator](#)

The Fabric Overlay Orchestrator feature is an easy-to-use GUI wizard that simplifies the process of configuring a self-orchestrated SD-WAN overlay within a single Security Fabric. This feature is self-orchestrated since no additional tool or device, aside from the FortiGate devices themselves, is required to orchestrate this configuration. An SD-WAN overlay configuration consists of IPsec and BGP configuration settings.

Currently, the Fabric Overlay Orchestrator supports a single hub architecture and builds upon an existing Security Fabric configuration. This feature configures the root FortiGate as the SD-WAN overlay hub and the downstream first-level FortiGates as the spokes.

After configuring the Fabric Overlay, you can complete the SD-WAN deployment by configuring SD-WAN rules.

Prerequisites

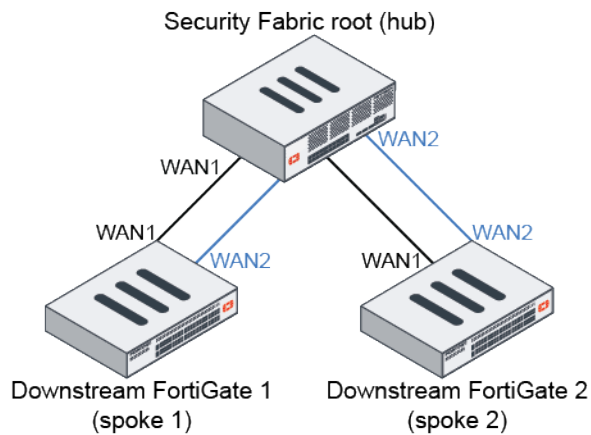
Create a single Fortinet Security Fabric with the following components:

- A root FortiGate and one or more downstream FortiGates all running FortiOS 7.2.4 or later
- A FortiAnalyzer, or cloud logging using FortiAnalyzer Cloud or FortiGate Cloud
 - For FortiGate Cloud, all downstream devices must belong to the same FortiCloud account

For more information about configuring these components, see [Configuring the root FortiGate and downstream FortiGates](#), [Configuring FortiAnalyzer](#), and [Configuring cloud logging](#) in the FortiOS Administration Guide.

Network topology

The Fabric Overlay Orchestrator supports configuring an overlay for the following hub and spoke topology using ADVPN and a single hub.



This topology corresponds to the [single datacenter \(active-passive gateway\)](#) design using the [IPsec overlay](#) design of one-to-one overlay mapping per underlay. For more details on these topics, see the [SD-WAN Architectures for Enterprise](#) guide.

Using the Fabric Overlay Orchestrator

The following steps should be used to configure a self-orchestrated SD-WAN overlay within a single Security Fabric. These steps must be followed in order, and assume that the prerequisites and network topology are in place.

1. Configure the root FortiGate using the Fabric Overlay Orchestrator.
2. Configure one or more downstream FortiGates using the Fabric Overlay Orchestrator.
3. Configure an overlay on the spoke for an additional incoming interface on the hub (if applicable).
4. Verify the firewall policies on the hub FortiGate.
5. Verify the Fabric Overlay created by the Fabric Overlay Orchestrator:
 - a. Verify the IPsec VPN tunnels on the hub FortiGate.
 - b. Verify BGP routing on the hub FortiGate.
 - c. Verify the performance SLAs on the hub FortiGate.
 - d. Verify the firewall policies on a spoke FortiGate.
 - e. Verify the IPsec VPN tunnels on a spoke FortiGate.
 - f. Verify BGP routing on a spoke FortiGate.
 - g. Verify the performance SLAs on a spoke FortiGate.
 - h. Verify the spoke-to-spoke ADVPN communication.
6. Configure SD-WAN rules on the hub FortiGate.
7. Configure SD-WAN rules on the spoke FortiGates.

When configuring the root and downstream FortiGates, the Fabric Overlay Orchestrator configures the following settings in the background:

- IPsec overlay configuration (hub and spoke ADVPN tunnels)
- BGP configuration
- Policy routing
- SD-WAN zones
- SD-WAN performance SLAs

The FortiGate's role in the SD-WAN overlay is automatically determined by its role in the Security Fabric. The Fabric root will be the hub, and any first-level downstream devices from the Fabric root will be spokes.

After using the Fabric Overlay Orchestrator on all FortiGates and verifying the overlay settings, complete the SD-WAN deployment configuration using steps 3 (if applicable), and steps 6 and 7. See [SD-WAN rules](#) in the FortiOS Administration Guide for more information.



For a detailed example configuration, see [Using the Fabric Overlay Orchestrator](#) in the FortiOS Administration Guide.

Creating firewall policies

The Fabric Overlay Orchestrator can create firewall policies to allow all traffic through the SD-WAN overlay, or firewall policies to just allow health check traffic through it instead. When the Fabric Overlay Orchestrator is enabled on the root FortiGate, there are three *Policy creation* options:

- *Automatic*: automatically create policies for the loopback interface and tunnel overlays.
- *Health check*: automatically create a policy for the loopback interface so the SD-WAN health checks are functional.
- *Manual*: no policies are automatically created.



The *Automatic* policy creation option creates wildcard allow policies for the tunnel overlays. For some cases, these policies do not provide the necessary granularity to restrict overlay traffic to specific subnets or hosts.



When the Fabric Overlay Orchestrator is configured on a device, changing the policy creation rule will create new policies based on the rule, but it will not delete existing policies. Deleting existing policies must be performed manually.

Reporting

7.2.0

- SD-WAN chart to include more ADVPN shortcut information FAZ on page 55
- SD-WAN chart for MOS scoring FAZ on page 57

7.2.1

- Bandwidth and applications report update FAZ 7.2.1 on page 61

SD-WAN chart to include more ADVPN shortcut information - FAZ



This information is also available in the FortiAnalyzer 7.2 Administration Guide:

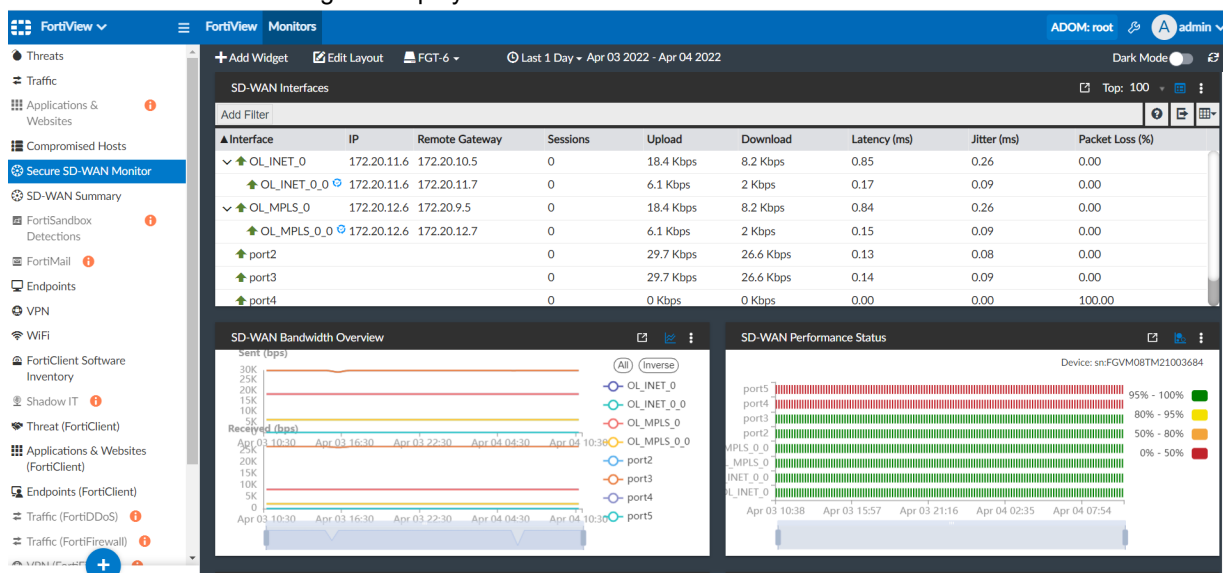
- [Secure SD-WAN Monitor](#)

The *SD-WAN Interfaces* widget is available in *FortiView > Monitors > Secure SD-WAN Monitor*.

This widget displays the following information for SD-WAN interfaces: IP, Remote Gateway, Sessions, Upload, Download, Latency (ms), Jitter (ms), and Packet Loss (%). The Upload and Download columns can be used to show outbound and inbound bandwidth. For a VPN tunnel interface, IP and Remote Gateway are the local IP and Remote Gateway IP of the VPN tunnel.

To view the SD-WAN interface information:

1. Go to *FortiView > Monitors > Secure SD-WAN Monitor*.
The SD-WAN Interfaces widget is displayed.



2. If there is an expand icon in the row, click the icon to view the ADVPN shortcut information in a row below. The IP and Remote Gateway are the local spoke IP and remote spoke IP of the shortcut VPN tunnel.

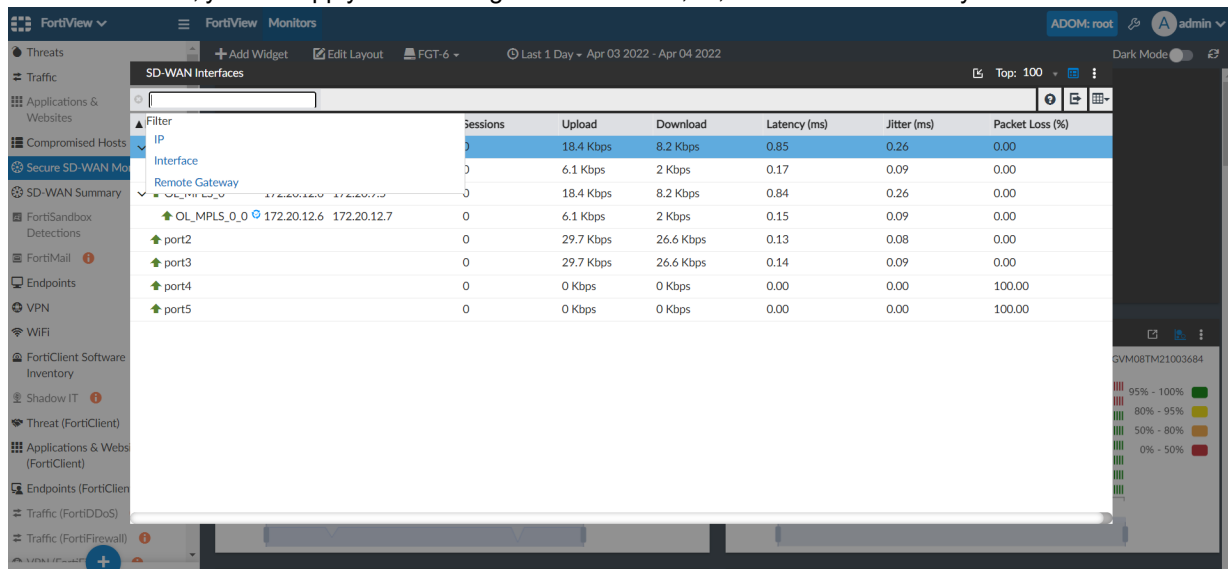
The screenshot shows the FortiView SD-WAN Interfaces widget. The table displays the following data:

Interface	IP	Remote Gateway	Sessions	Upload	Download	Latency (ms)	Jitter (ms)	Packet Loss (%)
OL_INET_0	172.20.11.6	172.20.10.5	0	18.4 Kbps	8.2 Kbps	0.85	0.26	0.00
OL_INET_0_0	172.20.11.6	172.20.11.7	0	6.1 Kbps	2 Kbps	0.17	0.09	0.00
OL_MPLS_0	172.20.12.6	172.20.9.5	0	18.4 Kbps	8.2 Kbps	0.84	0.26	0.00
OL_MPLS_0_0	172.20.12.6	172.20.12.7	0	6.1 Kbps	2 Kbps	0.15	0.09	0.00
port2			0	29.7 Kbps	26.6 Kbps	0.13	0.08	0.00
port3			0	29.7 Kbps	26.6 Kbps	0.14	0.09	0.00
port4			0	0 Kbps	0 Kbps	0.00	0.00	100.00
port5			0	0 Kbps	0 Kbps	0.00	0.00	100.00

The following information is available in the widget:

Interface	The name of the interface.
IP	The IP address for the interface.
Remote Gateway	The remote gateway IP address.
Sessions	The number of sessions for the interface.
Upload	The upload speed for the interface.
Download	The download speed for the interface.
Latency (ms)	The latency for the interface.
Jitter (ms)	The jitter for the interface.
Packet Loss (%)	The packet loss for the interface.

3. In the table chart, you can apply the following filters: Interface, IP, and Remote Gateway.



Filter	Sessions	Upload	Download	Latency (ms)	Jitter (ms)	Packet Loss (%)
IP	0	18.4 Kbps	8.2 Kbps	0.85	0.26	0.00
Interface	0	6.1 Kbps	2 Kbps	0.17	0.09	0.00
Remote Gateway	0	18.4 Kbps	8.2 Kbps	0.84	0.26	0.00
OL_MPLS_0_0 172.20.12.6 172.20.12.7	0	6.1 Kbps	2 Kbps	0.15	0.09	0.00
port2	0	29.7 Kbps	26.6 Kbps	0.13	0.08	0.00
port3	0	29.7 Kbps	26.6 Kbps	0.14	0.09	0.00
port4	0	0 Kbps	0 Kbps	0.00	0.00	100.00
port5	0	0 Kbps	0 Kbps	0.00	0.00	100.00

SD-WAN chart for MOS scoring - FAZ



This information is also available in the FortiAnalyzer 7.2 Administration Guide:

- [SD-WAN Summary](#)

An *Audio MOS Score* widget is added to *FortiView > Monitors > Secure SD-WAN Monitor* and *FortiView > Monitors > SD-WAN Summary*. These widgets display logs for the MOS (mean opinion score) of voice and video traffic.

MOS is a method to measure the impact network quality has on the quality of a voice call. It is the industry standard for measuring voice and video quality on a WAN link.

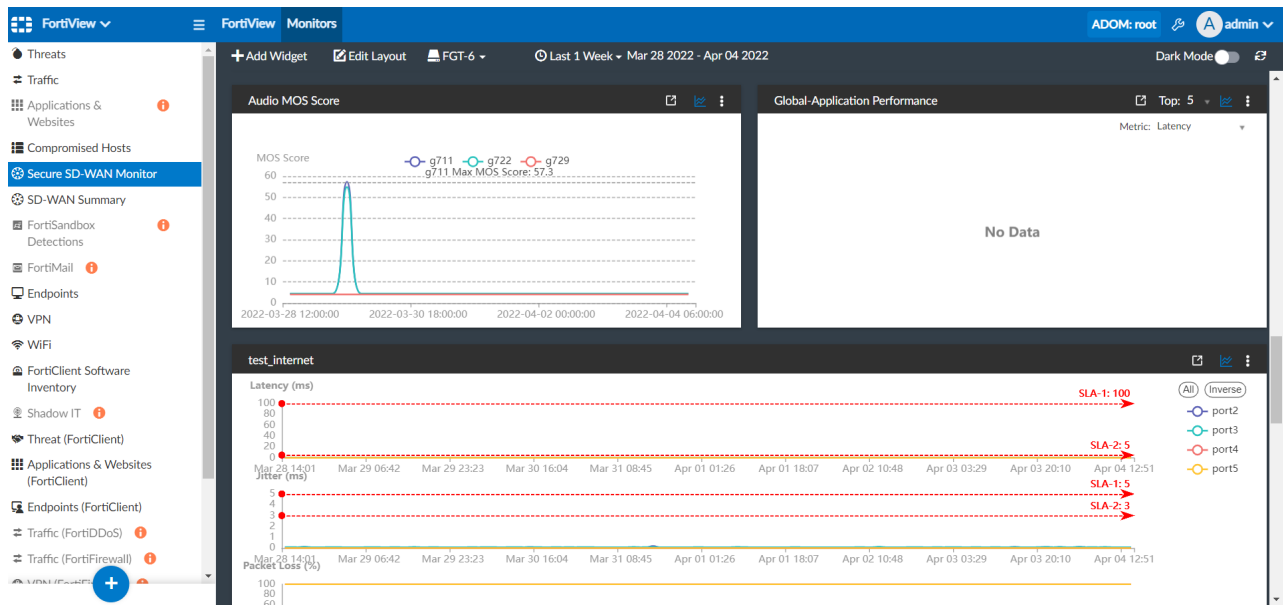


The FortiGate version must be on version 7.2 or later and have the MOS codec and MOS threshold attributes defined for SD-WAN health check in order for FortiAnalyzer to display information in the MOS scoring widgets.

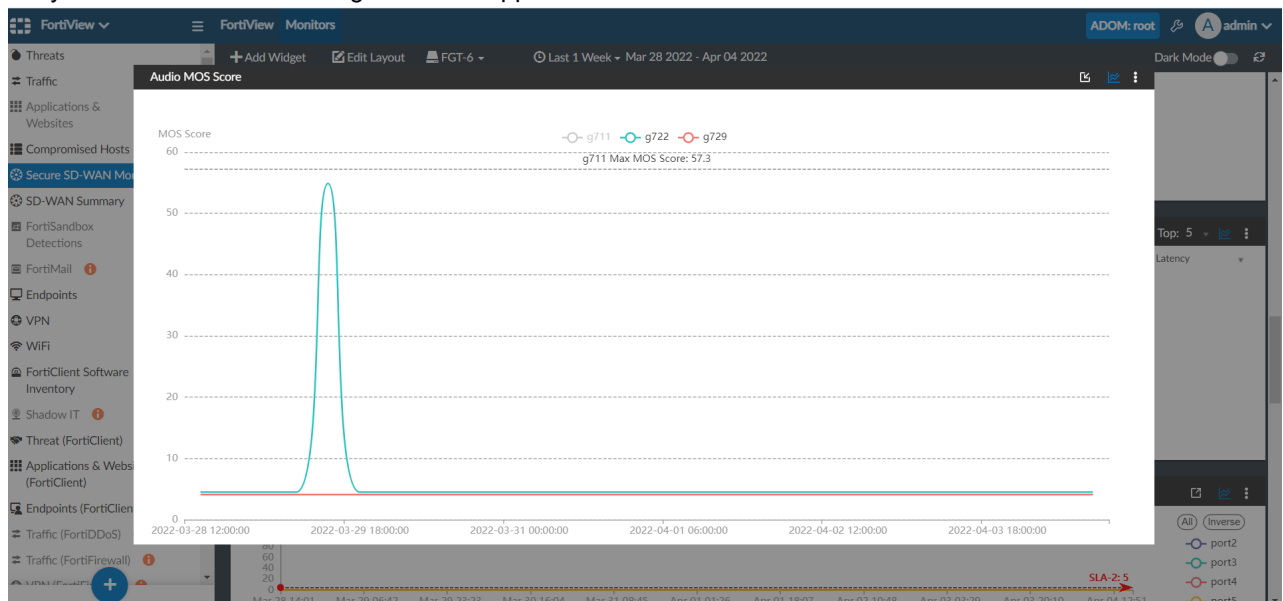
To view the Audio MOS Score for individual devices:

1. Go to *FortiView > Monitors > Secure SD-WAN Monitor*.
2. Click *Add Widget*, and add the *Audio MOS Score* widget.

The widget includes a line graph of the MOS score for different codecs for the selected device over a specified time period.



- Click a codec in the legend to make it appear/disappear on the chart. Greyed-out interfaces on the legend do not appear on the chart.

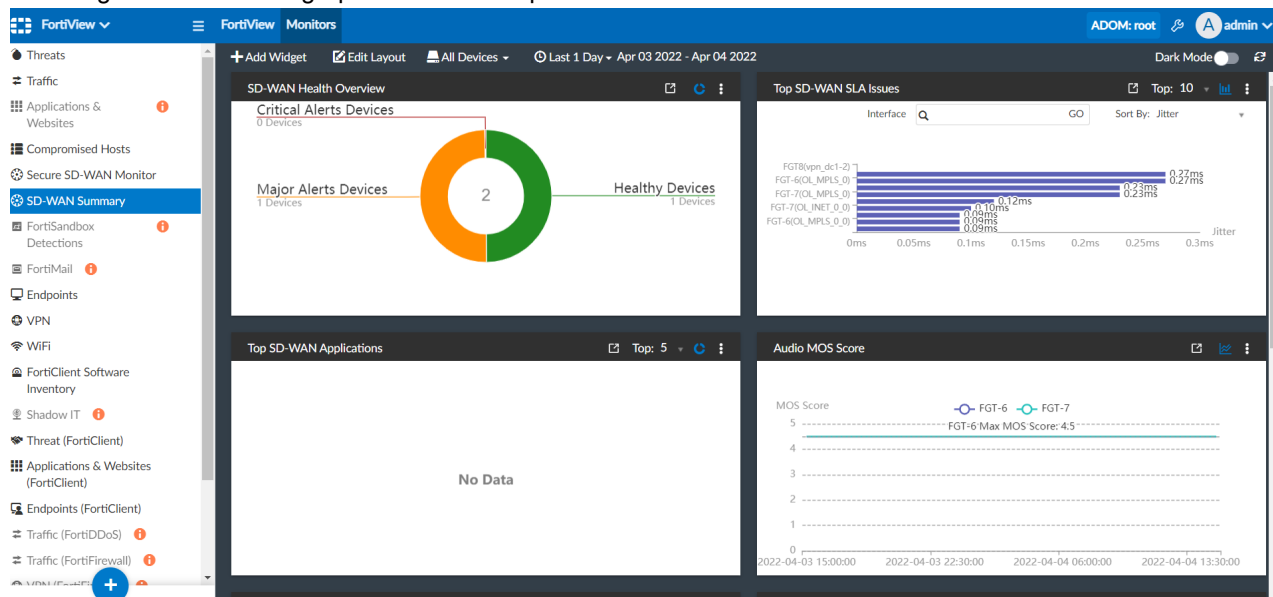


- Hover your cursor over the chart to see a summary at that point. This summary includes the MOS score and the VoIP quality at that time. VoIP quality is divided into levels based on MOS scoring: Excellent = 4.3 - 5.0, Good = 4.0 - 4.3, Fair = 3.6 - 4.0, Poor = 3.1 - 3.6, Bad = 2.6 - 1.0.

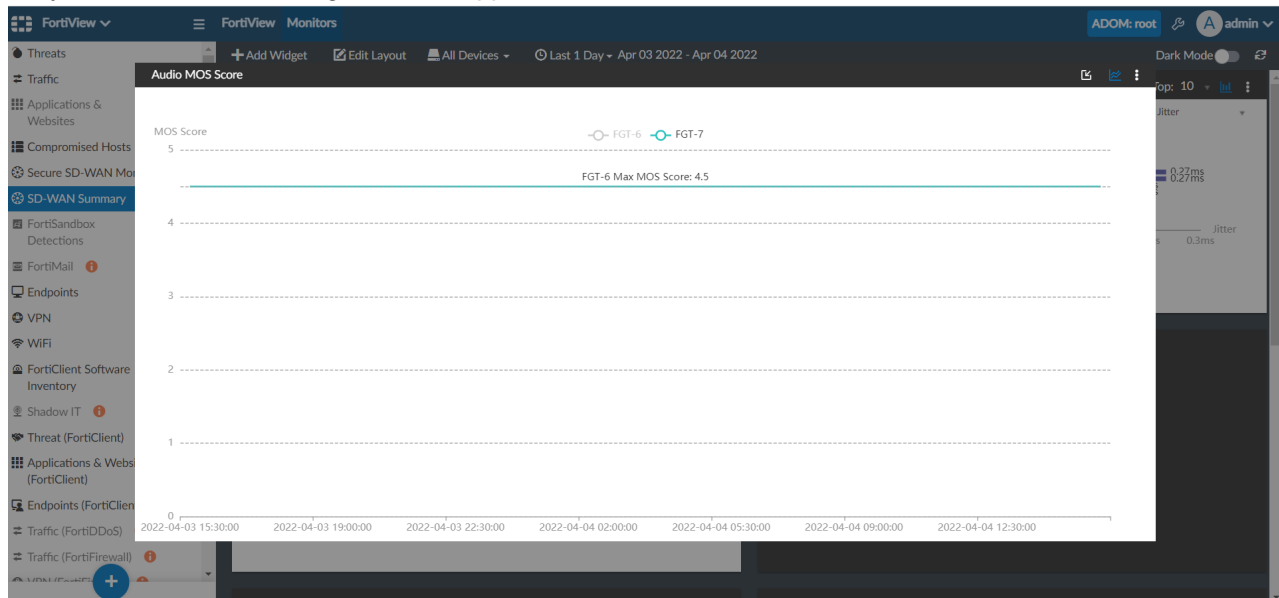


To view the Audio MOS Score across all devices:

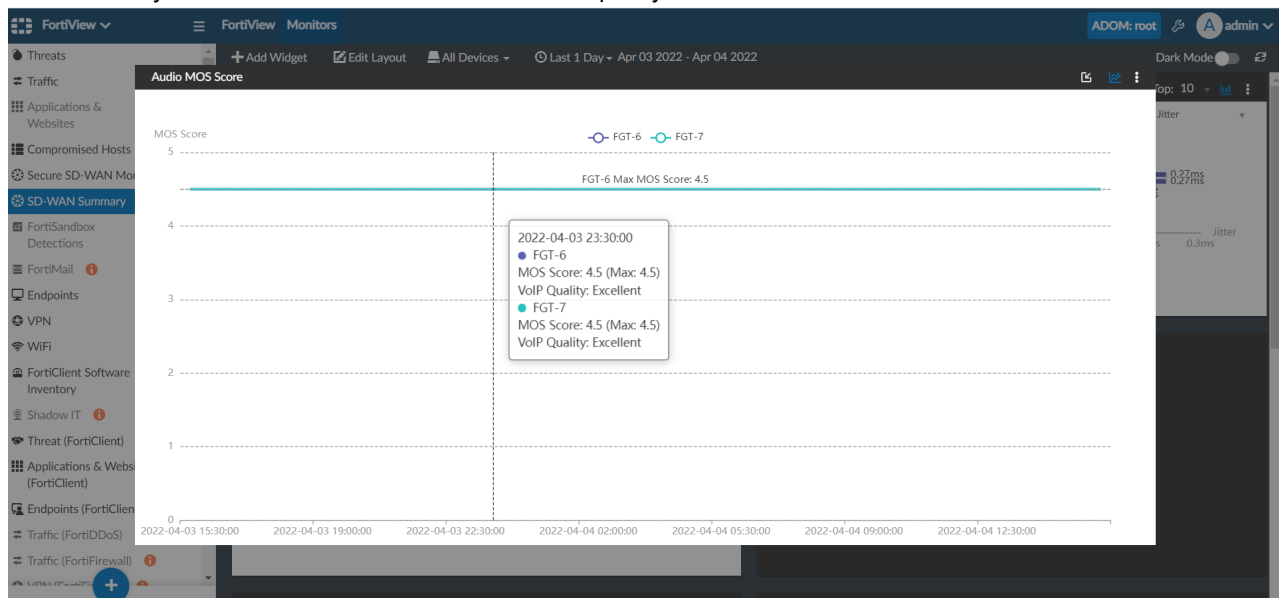
1. Go to *FortiView > Monitors > SD-WAN Summary*.
 2. Click *Add Widget*, and add the *Audio MOS Score* widget.
- The widget includes a line graph of MOS score per device on the network.



- Click a device in the legend to make it appear/disappear on the chart.
Greyed-out devices on the legend do not appear on the chart.



- Hover your cursor over the chart to see a summary at that point.
This summary includes the MOS score and the VoIP quality at that time.



To configure the FortiGate MOS codec and threshold in health check settings:

- Access the FortiGate CLI.
- Enter the following commands:


```
config system sdwan
  config health-check
    edit <name>
      set server {string}
      set sla-fail-log-period {integer}
      set sla-pass-log-period {integer}
```

```

        set members <seq-num1>, <seq-num2>, ...
        set mos-codec [g711|g722|...]
    config sla
        edit <id>
            set link-cost-factor {option1}, {option2}, ...
            set mos-threshold {string}
        next
    end

```

For example:

```

config system sdwan
config health-check
    edit "test_dc"
        set server "10.200.1.1"
        set sla-fail-log-period 15
        set sla-pass-log-period 15
        set members 1 2
        set mos-codec g722
    config sla
        edit 1
            set link-cost-factor latency jitter packet-loss mos
            set mos-threshold "2.0"
        next
    end

```

Bandwidth and applications report update - FAZ 7.2.1

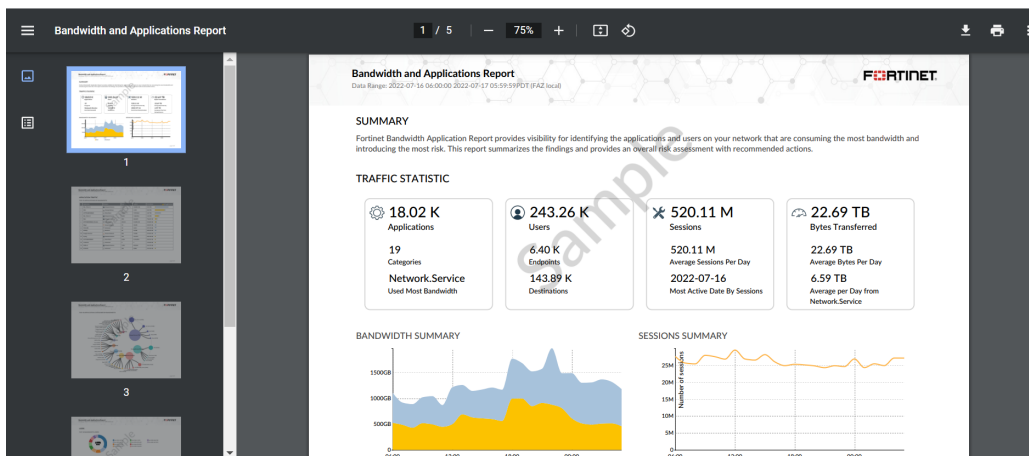


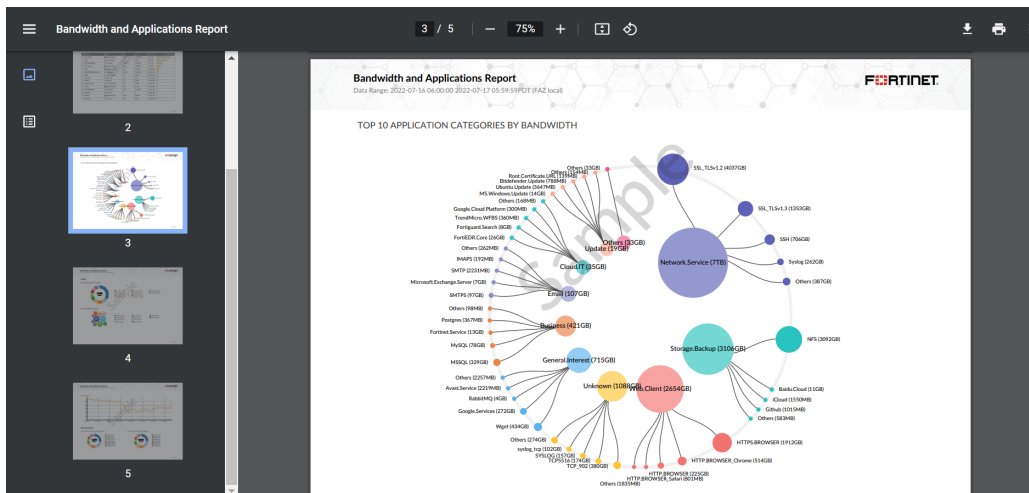
This information is also available in the FortiAnalyzer 7.2 Administration Guide:

- [Report template library](#)

The *Bandwidth and Applications Report* is updated to improve data visualization.

Following is a sample of the report in PDF:





To use the Bandwidth and Applications Report template:

1. Go to **Reports > Report Definitions > Templates**.
From the *Preview* column, you can click **PDF** or **HTML** to preview the report in that format.
2. Select the checkbox for **Template - Bandwidth and Applications Report**.

Reports

ADOM: root-new

admin

Generated Reports

Create NewViewDeleteMore

Search...

Report Definitions

All Reports

Templates

Chart Library

Macro Library

Datasets

Advanced

<input checked="" type="checkbox"/>	Title		Category	Preview
<input type="checkbox"/>	Template - 360 Protection Report	Inventory of the FortiGate devices over a 30 day period.	System	HTML PDF
<input type="checkbox"/>	Template - 360 Security Report	reat, app, user, incident, compromised host and so on.	Security	HTML PDF
<input type="checkbox"/>	Template - 360-Degree Security Review	ontrol, Threat Detection, Data Exfiltration Detection, Endpoint Detection, P	Security	HTML PDF
<input type="checkbox"/>	Template - Admin and System Events Report	m severity event counts.	System	HTML PDF
<input type="checkbox"/>	Template - Application Risk and Control	web categories, vulnerability exploits, virus, botnet, adware malicious attac	Application	HTML PDF
<input type="checkbox"/>	Template - Asset and Identity Report	their users, vulnerabilities, software installed as well as running processes.	Assets	HTML PDF
<input checked="" type="checkbox"/>	Template - Bandwidth and Applications Report	aries - by users and applications	Application	HTML PDF
<input type="checkbox"/>	Template - Client Reputation	user, devices, threat summary.	User	HTML PDF
<input type="checkbox"/>	Template - Cyber Threat Assessment	Control, Threat Detection, Prevention and Recommended Actions.	Security	HTML PDF
<input type="checkbox"/>	Template - Cyber-Bullying Indicators Report		Application	HTML PDF
<input type="checkbox"/>	Template - Daily Summary Report	reat, app, user, incident, compromised host and so on.	Security	HTML PDF
<input type="checkbox"/>	Template - Data Loss Prevention Detailed Report	Web, and FTP.	Security	HTML PDF
<input type="checkbox"/>	Template - Detailed Application Usage and Risk	to-peer, remote access, email, Backup and storage, general access. Includ	Application	HTML PDF
<input type="checkbox"/>	Template - DNS Report	ivity on the network.	System	HTML PDF
<input type="checkbox"/>	Template - DNS Security Report	arity.	Security	HTML PDF
<input type="checkbox"/>	Template - Email Report	ed files of email	Security	HTML PDF

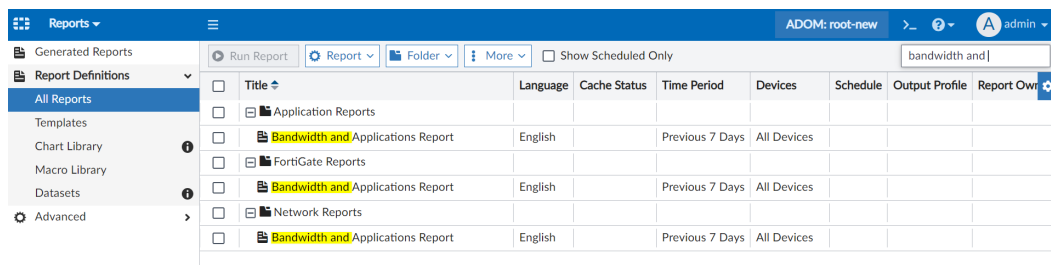
FORTINET

0.0%

3. From the *More* dropdown, click **Clone** to clone template and make adjustments.
You can also click **Create Report** to create a report using the template.

To run the Bandwidth and Applications Report:

1. Go to **Reports > Report Definitions > All Reports**.
2. Double-click the row for **Bandwidth and Applications Report**.
You can find the report using the search bar, for example:



3. In the *Generated Reports* tab, click *Run Report*.
4. Once the report is generated, click a format in the *Format* column to view the report.

Routing

7.2.0

- [SD-WAN segmentation over a single overlay on page 64](#)
- [Multiple members per SD-WAN neighbor configuration on page 79](#)
- [SD-WAN in large scale deployments on page 85](#)
- [Route map rules and BGP routes on page 97](#)
- [BGP socket limit increase on page 97](#)
- [IKE embryonic limit on page 97](#)

7.2.1

- [GUI support for advanced BGP options 7.2.1 on page 97](#)
- [Support BGP AS number input in asdot and asdot+ format 7.2.1 on page 100](#)
- [Support cross-VRF local-in and local-out traffic for local services 7.2.1 on page 102](#)

7.2.4

- [Matching BGP extended community route targets in route maps 7.2.4 on page 104](#)
- [Add static route tag and BGP neighbor password 7.2.4 on page 109](#)

SD-WAN segmentation over a single overlay



This information is also available in the FortiOS 7.2 Administration Guide:

- [Mean opinion score calculation and logging in performance SLA health checks](#)

SD-WAN, VPN, and BGP configurations support L3 VPN segmentation over a single overlay. In these configurations, a hub and spoke SD-WAN deployment requires that branch sites, or spokes, are able to accommodate multiple companies or departments, and each company's subnet is separated by a different VRF. A subnet on one VRF cannot communicate with a subnet on another VRF between different branches, but can communicate with the same VRF.

New SD-WAN options

VRF-aware SD-WAN health checks

SD-WAN on the originating spoke can tag the health check probes with the correct VRF when transmitting to a multi-VRF tunnel. The hub can then forward the probes to the correct health check server in the same VRF as the hub.

```
config system sdwan
  config health-check
    edit <name>
      set vrf <vrf id>
```

```

        set source <address>
    next
end
end

```

vrf <vrf id>	Virtual Routing Forwarding ID.
source <address>	Source IP address used in the health-check packet to the server.

Overlay stickiness

When a hub has multiple overlays, traffic received on one overlay should egress on the same overlay when possible. The `service-sla-tie-break` option ensures overlay stickiness. In SD-WAN service rules, options are available to ensure that traffic received in a zone stays in that zone.

```

config system sdwan
    config zone
        edit <name>
            set service-sla-tie-break input-device
        next
    end
    config service
        edit <id>
            set input-zone <zone>
            set tie-break input-device
        next
    end
end

```

service-sla-tie-break input-device	Members that meet the SLA are selected by matching the input device.
input-zone <zone>	Source input-zone name.
tie-break input-device	Members that meet the SLA are selected by matching the input device.

New IPsec options

Configurable rate limit for shortcut offers sent by the hub

By default, the hub sends a shortcut offer to a spoke every five seconds. If the hub continues to send offers that keep failing, and there are a large number of spokes, this can cause a high load on the hub. This setting makes the interval between shortcut offers configurable.

```

config vpn ipsec phase1-interface
    edit <name>
        set auto-discovery-offer-interval <interval>
    next
end

```

auto-discovery-offer- interval <interval>	Interval between shortcut offer messages, in seconds (1 - 300, default = 5).
--	--

Segmentation over a single overlay

Segmentation requires that VRF info is encapsulated within the IPsec VPN tunnel. This setting enables multi-VRF IPSEC tunnels.

```
config vpn ipsec phase1-interface
    edit <name>
        set encapsulation vpn-id-ipip
    next
end
```

encapsulation vpn-id-ipip VPN ID with IPIP encapsulation.

New VPN configuration for BGP

The role of a VRF can be specified, along with other VRF details. In FortiOS 7.2.0 to 7.2.3, up to 64 VRFs can be configured per VDOM for any device. In FortiOS 7.2.4, up to 252 VRFs can be configured per VDOM for any device.

```
config router bgp
    config vrf
        edit <vrf>
            set role {standalone | ce | pe}
            set rd <string>
            set export-rt <route_target>
            set import-rt <route_target>
            set import-route-map <route_map>
            config leak-target
                edit <vrf>
                    set route-map <route-map>
                    set interface <interface>
                next
            end
        next
    end
end
```

role {standalone ce pe}	VRF role: standalone, customer edge (CE), or provider edge (PE).
rd <string>	Route Distinguisher: AA AA:NN. This option is only available when the role is CE.
export-rt <route_target>	List of export route target. This option is only available when the role is CE.
import-rt <route_target>	List of import route target. This option is only available when the role is CE.
import-route-map <route_map>	Import route map. This option is only available when the role is CE.
route-map <route-map>	Route map of VRF leaking.
interface <interface>	Interface that is used to leak routes to the target VRF.



In FortiOS 7.0, config vrf was config vrf-leak, and config leak-target was config target.

Display BGP routes by VRF and neighbor

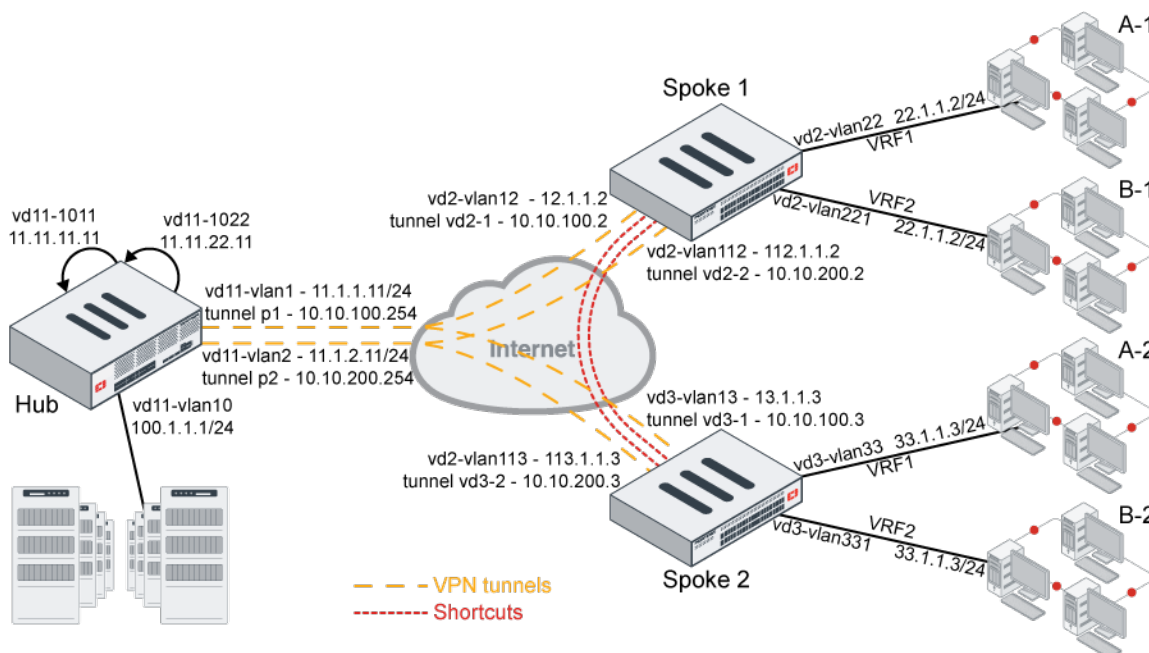
```
# diagnose ip router bgp set-filter vrf <vrf>
# diagnose ip router bgp set-filter neighbor <neighbor address>
# diagnose ip router bgp set-filter reset
# execute router clear bgp vpnv4 unicast soft {in | out}
# get router info filter show
# get router info filter vrf {vrf | all}
```

Examples

In example 1, multiple companies (or departments of a company) share the ADVPN. Company A and company B each have two branches in two different locations. Company A's branches (A-1 and A-2) can talk to each other using the VPN shortcut, but not to company B's branches (B-1 and B-2). Likewise, company B's branches can talk to each other using the VPN shortcut, but not to company A's branches. Traffic can share the tunnels and shortcuts, but cannot be mixed up.

[Example 2](#) shows that performance SLA health checks can be sent from a spoke's VRF to the loopback on the hub that is in the same VRF.

[Example 3](#) shows that when traffic is ingress on the hub on one overlay, it will preferably egress on the same overlay.



Example 1

In this example, two spokes each have two tunnels to the hub.

- Each spoke has two VRFs behind it that can use the same IP address or subnets.
- The computers in VRF1 behind spoke 1 can talk to the computers in VRF1 behind spoke 2, but not to any of the computers in the VRF2s behind either spoke.
- The computers in VRF2 behind spoke 1 can talk to the computers in VRF2 behind spoke 2, but not to any of the computers in the VRF1s behind either spoke.

To configure the hub:

```
config router bgp
  set as 65505
  set router-id 11.11.11.11
  set ibgp-multipath enable
  set additional-path enable
  set additional-path-vpnv4 enable
  set cluster-id 11.12.13.14
  set additional-path-select 3
  config neighbor-group
    edit "gr1"
      set capability-graceful-restart enable
      set capability-default-originate enable
      set next-hop-self-rr enable
      set soft-reconfiguration-vpnv4 enable
      set remote-as 65505
      set additional-path both
      set additional-path-vpnv4 both
      set adv-additional-path 3
      set route-reflector-client enable
      set route-reflector-client-vpnv4 enable
    next
    edit "gr2"
      set capability-graceful-restart enable
      set capability-default-originate enable
      set next-hop-self-rr enable
      set soft-reconfiguration-vpnv4 enable
      set remote-as 65505
      set additional-path both
      set additional-path-vpnv4 both
      set adv-additional-path 3
      set route-reflector-client enable
      set route-reflector-client-vpnv4 enable
    next
  end
  config neighbor-range
    edit 1
      set prefix 10.10.100.0 255.255.255.0
      set neighbor-group "gr1"
    next
    edit 2
      set prefix 10.10.200.0 255.255.255.0
      set neighbor-group "gr2"
    next
  end
  config network
    edit 12
      set prefix 11.11.11.11 255.255.255.255
    next
    edit 22
      set prefix 11.11.22.11 255.255.255.255
    next
    edit 10
      set prefix 100.1.1.0 255.255.255.0
    next
  end
```

```
        edit 33
            set prefix 11.1.1.0 255.255.255.0
        next
    end
    config vrf
        edit "0"
            set role pe
        next
        edit "1"
            set role ce
            set rd "1:1"
            set export-rt "1:1"
            set import-rt "1:1"
        next
        edit "2"
            set role ce
            set rd "2:1"
            set export-rt "2:1"
            set import-rt "2:1"
        next
    end
end

config vpn ipsec phase1-interface
    edit "p1"
        set type dynamic
        set interface "vd11-vlan1"
        set peertype any
        set net-device disable
        set proposal aes128-sha1
        set add-route disable
        set dpd on-idle
        set dhgrp 5
        set auto-discovery-sender enable
        set auto-discovery-offer-interval 10
        set encapsulation vpn-id-ipip
        set psksecret *****
        set dpd-retryinterval 60
    next
    edit "p2"
        set type dynamic
        set interface "vd11-vlan2"
        set peertype any
        set net-device disable
        set proposal aes128-sha1
        set add-route disable
        set dpd on-idle
        set dhgrp 5
        set auto-discovery-sender enable
        set auto-discovery-offer-interval 10
        set encapsulation vpn-id-ipip
        set psksecret *****
        set dpd-retryinterval 60
    next
end
```

```
config vpn ipsec phase2-interface
  edit "p1"
    set phase1name "p1"
    set proposal aes128-sha1
    set dhgrp 5
  next
  edit "p2"
    set phase1name "p2"
    set proposal aes128-sha1
    set dhgrp 5
  next
end
```

To configure a spoke:

```
config router bgp
  set as 65505
  set router-id 2.2.2.2
  set ebgp-multipath enable
  set ibgp-multipath enable
  set network-import-check disable
  set additional-path enable
  set additional-path6 enable
  set additional-path-ipv4 enable
  set recursive-next-hop enable
  set graceful-restart enable
  set additional-path-select 4
config neighbor
  edit "10.10.100.254"
    set capability-dynamic enable
    set capability-graceful-restart-ipv4 enable
    set soft-reconfiguration enable
    set soft-reconfiguration-ipv4 enable
    set remote-as 65505
    set additional-path both
    set additional-path-ipv4 both
    set adv-additional-path 3
  next
  edit "10.10.200.254"
    set capability-dynamic enable
    set capability-graceful-restart-ipv4 enable
    set soft-reconfiguration enable
    set soft-reconfiguration-ipv4 enable
    set remote-as 65505
    set additional-path both
    set additional-path-ipv4 both
    set adv-additional-path 3
  next
end
config network
  edit 3
    set prefix 22.1.1.0 255.255.255.0
  next
  edit 4
    set prefix 12.12.12.0 255.255.255.0
  next
```

```

end
config vrf
    edit "0"
        set role pe
    next
    edit "1"
        set role ce
        set rd "1:1"
        set export-rt "1:1"
        set import-rt "1:1"
    next
    edit "2"
        set role ce
        set rd "2:1"
        set export-rt "2:1"
        set import-rt "2:1"
    next
end
end

config vpn ipsec phase1-interface
    edit "vd2-1"
        set interface "vd2-vlan12"
        set peertype any
        set net-device enable
        set proposal aes128-sha1
        set add-route disable
        set dhgrp 5
        set idle-timeout enable
        set idle-timeoutinterval 5
        set auto-discovery-receiver enable
        set encapsulation vpn-id-ipip
        set remote-gw 11.1.1.11
        set psksecret *****
    next
    edit "vd2-2"
        set interface "vd2-vlan112"
        set peertype any
        set net-device enable
        set proposal aes128-sha1
        set add-route disable
        set dhgrp 5
        set auto-discovery-receiver enable
        set encapsulation vpn-id-ipip
        set remote-gw 11.1.2.11
        set psksecret *****
    next
end

config vpn ipsec phase2-interface
    edit "vd2-1"
        set phase1name "vd2-1"
        set proposal aes128-sha1
        set dhgrp 5
        set auto-negotiate enable
    next
    edit "vd2-2"

```

```

        set phasename "vd2-2"
        set proposal aes128-sha1
        set dhgrp 5
        set auto-negotiate enable
    next
end
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "SASE"
        next
        edit "zon2"
        next
    end
    config members
        edit 1
            set interface "vd2-1"
            set cost 10
        next
        edit 2
            set interface "vd2-2"
            set cost 20
        next
    end
    config health-check
        edit "ping"
            set server "11.11.11.11"
            set members 1 2
            config sla
                edit 1
                    set latency-threshold 200
                    set jitter-threshold 50
                next
            end
        next
        edit "1"
            set server "22.1.1.2"
            set vrf 1
            set members 1 2
        next
    end
    config service
        edit 2
            set mode sla
            set dst "100-200"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 2
            set use-shortcut-sla disable
        next
        edit 1

```

```

        set name "test-tag"
        set mode sla
        set dst "001-100"
        config sla
            edit "ping"
                set id 1
            next
        end
        set priority-members 1 2
    next
end
end

```

To check the spoke 1 routes:

```

# get router info routing-table bgp
Routing table for VRF=0
B*      0.0.0.0/0 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11 vrf 0),
04:42:57, [1/0]
        [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11 vrf 0),
04:42:57, [1/0]
B       1.1.1.1/32 [200/0] via 11.1.1.1 [2] (recursive via 12.1.1.1, vd2-vlan12), 04:42:57,
[1/0]
B       1.222.222.222/32 [200/0] via 11.1.1.1 [2] (recursive via 12.1.1.1, vd2-vlan12),
04:42:57, [1/0]
B       11.11.11.11/32 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11 vrf
0), 04:42:57, [1/0]
        [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11 vrf
0), 04:42:57, [1/0]
B       33.1.1.0/24 [200/0] via 10.10.100.254 [2] (recursive via vd2-1 tunnel 11.1.1.11 vrf
0), 04:42:57, [1/0]
        [200/0] via 10.10.200.254 [2] (recursive via vd2-2 tunnel 11.1.2.11 vrf
0), 04:42:57, [1/0]
B       100.1.1.0/24 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11 vrf 0),
04:42:57, [1/0]
        [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11 vrf 0),
04:42:57, [1/0]

Routing table for VRF=1
B V     33.1.1.0/24 [200/0] via 10.10.100.3 [2] (recursive via vd2-1 tunnel 11.1.1.11 vrf
0), 04:42:57, [1/0]
        [200/0] via 10.10.200.3 [2] (recursive is directly connected, vd2-2_0),
04:42:57, [1/0]

Routing table for VRF=2
B V     33.1.1.0/24 [200/0] via 10.10.100.3 [2] (recursive via vd2-1 tunnel 11.1.1.11 vrf
0), 04:42:56, [1/0]
        [200/0] via 10.10.200.3 [2] (recursive is directly connected, vd2-2_0),
04:42:56, [1/0]

VRF1 routes:

# get router info filter vrf 1
# get router info routing-table bgp
Routing table for VRF=1
B V     33.1.1.0/24 [200/0] via 10.10.100.3 [2] (recursive via vd2-1 tunnel 11.1.1.11 vrf
0), 04:44:11, [1/0]

```

```
[200/0] via 10.10.200.3 [2] (recursive is directly connected, vd2-2_0),
04:44:11, [1/0]
```

To test the configuration on shortcut 1:

1. From VRF1 of spoke 1 ping VRF1 of spoke 2 and from VRF2 of spoke 1 ping VRF2 spoke 2. Both VRF1 and VRF2 source and destination IP addresses are the same, so you can see how the traffic is isolated
2. Check sessions on spoke 1:

The output `vd=<vdom ID>:<VRF ID>` indicates that sessions are created in and stay in the corresponding VRFs.

- User at 22.1.1.22 in VRF1 on spoke 1 pings 33.1.1.33 in VRF1 on spoke2.

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=21 expire=42 timeout=0 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=may_dirty
statistic(bytes/packets/allow_err): org=420/5/1 reply=420/5/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=89->131/131->89
gwy=10.10.200.3/22.1.1.22
hook=pre dir=org act=noop 22.1.1.22:48417->33.1.1.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.33:48417->22.1.1.22:0(0.0.0.0:0)
src_mac=02:4c:a5:fc:6a:7f
misc=0 policy_id=1 pol_uuid_idx=566 auth_info=0 chk_client_info=0 vd=1:1
serial=00092eee tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=1
rpdb_link_id=ff000001 ngfwid=n/a
npu_state=0x5040001 no_offload
no_ofld_reason: disabled-by-policy non-npu-intf
```

- User at 22.1.1.22 in VRF2 on spoke 1 pings 33.1.1.33 in VRF2 on spoke2:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=4 expire=56 timeout=0 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=may_dirty
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 39/0 rx speed(Bps/kbps): 39/0
orgin->sink: org pre->post, reply pre->post dev=113->131/131->113
gwy=10.10.200.3/22.1.1.22
hook=pre dir=org act=noop 22.1.1.22:55841->33.1.1.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.33:55841->22.1.1.22:0(0.0.0.0:0)
src_mac=02:4c:a5:fc:6a:7f
misc=0 policy_id=1 pol_uuid_idx=566 auth_info=0 chk_client_info=0 vd=1:2
serial=00092f77 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=1
rpdb_link_id=ff000001 ngfwid=n/a
npu_state=0x5040001 no_offload
no_ofld_reason: disabled-by-policy non-npu-intf
```

3. Check sessions on spoke 2:

The output `vd=<vdom ID>:<VRF ID>` indicates that sessions are created in and stay in the corresponding VRFs.

- User at 22.1.1.22 in VRF1 on spoke 1 pings 33.1.1.33 in VRF1 on spoke 2:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=11 expire=49 timeout=0 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=may_dirty npu
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 14/0 rx speed(Bps/kbps): 14/0
origin->sink: org pre->post, reply pre->post dev=132->92/92->132
gwy=33.1.1.33/10.10.200.2
hook=pre dir=org act=noop 22.1.1.22:27733->33.1.1.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.33:27733->22.1.1.22:0(0.0.0.0:0)
misc=0 policy_id=1 pol_uid_idx=630 auth_info=0 chk_client_info=0 vd=6:1
serial=000a29fd tos=ff/ff app_list=0 app=0 url_cat=0
rpd_b_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
npu info: flag=0x00/0x82, 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
no_ofld_reason: disabled-by-policy
```

- User at 22.1.1.22 in VRF2 on spoke 1 pings 33.1.1.33 in VRF2 on spoke 2:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=17 expire=43 timeout=0 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=may_dirty npu
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 9/0 rx speed(Bps/kbps): 9/0
origin->sink: org pre->post, reply pre->post dev=132->115/115->132
gwy=33.1.1.33/10.10.200.2
hook=pre dir=org act=noop 22.1.1.22:24917->33.1.1.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.33:24917->22.1.1.22:0(0.0.0.0:0)
dst_mac=02:4c:a5:fc:6a:7f
misc=0 policy_id=1 pol_uid_idx=630 auth_info=0 chk_client_info=0 vd=6:2
serial=000a29ca tos=ff/ff app_list=0 app=0 url_cat=0
rpd_b_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
npu info: flag=0x00/0x82, 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
no_ofld_reason: disabled-by-policy
```

To test the configuration on shortcut 2:

1. From VRF1 of spoke 1 ping VRF1 of spoke 2 and from VRF2 of spoke 1 ping VRF2 spoke 2. Both VRF1 and VRF2 source and destination IP addresses are the same, so you can see how the traffic is isolated
2. Check sessions on spoke 1:

The output `vd=<vdom ID>:<VRF ID>` indicates that sessions are created in and stay in the corresponding VRFs.

- User at 22.1.1.22 in VRF1 on spoke 1 pings 33.1.1.133 in VRF1 on spoke 2:

```
# diagnose sys session listsession info: proto=1 proto_state=00 duration=17 expire=45
timeout=0 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=may_dirty
statistic(bytes/packets/allow_err): org=336/4/1 reply=336/4/1 tuples=2
tx speed(Bps/kbps): 19/0 rx speed(Bps/kbps): 19/0
origin->sink: org pre->post, reply pre->post dev=89->137/137->89
gwy=10.10.200.3/22.1.1.22
hook=pre dir=org act=noop 22.1.1.22:25968->33.1.1.133:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.133:25968->22.1.1.22:0(0.0.0.0:0)
src_mac=02:4c:a5:fc:6a:7f
misc=0 policy_id=1 pol_uuid_idx=566 auth_info=0 chk_client_info=0 vd=1:1
serial=000aa475 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=2
rpdb_link_id=ff000002 ngfwid=n/a
npu_state=0x5040001 no_offload
no_ofld_reason: disabled-by-policy non-npu-intf
```

- User at 22.1.1.22 in VRF2 on spoke 1 pings 33.1.1.133 in VRF2 on spoke 2:

```
# diagnose sys session listsession info: proto=1 proto_state=00 duration=8 expire=53
timeout=0 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=may_dirty
statistic(bytes/packets/allow_err): org=252/3/1 reply=252/3/1 tuples=2
tx speed(Bps/kbps): 30/0 rx speed(Bps/kbps): 30/0
origin->sink: org pre->post, reply pre->post dev=113->137/137->113
gwy=10.10.200.3/22.1.1.22
hook=pre dir=org act=noop 22.1.1.22:28528->33.1.1.133:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.133:28528->22.1.1.22:0(0.0.0.0:0)
src_mac=02:4c:a5:fc:6a:7f
misc=0 policy_id=1 pol_uuid_idx=566 auth_info=0 chk_client_info=0 vd=1:2
serial=000aa49f tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=2
rpdb_link_id=ff000002 ngfwid=n/a
npu_state=0x5040001 no_offload
no_ofld_reason: disabled-by-policy non-npu-intf
```

3. Check sessions on spoke 2:

The output `vd=<vdom ID>:<VRF ID>` indicates that sessions are created in and stay in the corresponding VRFs.

- User at 22.1.1.22 in VRF1 on spoke 1 pings 33.1.1.133 in VRF1 on spoke 2:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=24 expire=38 timeout=0 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=may_dirty npu
statistic(bytes/packets/allow_err): org=336/4/1 reply=336/4/1 tuples=2
tx speed(Bps/kbps): 13/0 rx speed(Bps/kbps): 13/0
origin->sink: org pre->post, reply pre->post dev=138->92/92->138
gwy=33.1.1.133/10.10.200.2
hook=pre dir=org act=noop 22.1.1.22:25968->33.1.1.133:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.133:25968->22.1.1.22:0(0.0.0.0:0)
misc=0 policy_id=1 pol_uuid_idx=630 auth_info=0 chk_client_info=0 vd=6:1
serial=000aa476 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
npu info: flag=0x00/0x82, 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
no_ofld_reason: disabled-by-policy
```

- User at 22.1.1.22 in VRF2 on spoke 1 pings 33.1.1.133 in VRF2 on spoke2:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=15 expire=46 timeout=0 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=may_dirty npu
statistic(bytes/packets/allow_err): org=252/3/1 reply=252/3/1 tuples=2
tx speed(Bps/kbps): 16/0 rx speed(Bps/kbps): 16/0
origin->sink: org pre->post, reply pre->post dev=138->115/115->138
gwy=33.1.1.133/10.10.200.2
hook=pre dir=org act=noop 22.1.1.22:28528->33.1.1.133:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.133:28528->22.1.1.22:0(0.0.0.0:0)
misc=0 policy_id=1 pol_uuid_idx=630 auth_info=0 chk_client_info=0 vd=6:2
serial=000aa4a0 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
npu info: flag=0x00/0x82, 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
no_ofld_reason: disabled-by-policy
```

Example 2

In this example, SLA health checks are sent from a spoke's VRF to the loopback on the hub that is in the same VRF.

To configure the health check:

```

config system sdwan
  config health-check
    edit "1"
      set server "11.11.22.11"
      set vrf 1
      set source 22.1.1.2
      set members 1 2
      config sla
        edit 1
          set latency-threshold 200
          set jitter-threshold 50
        next
      end
    next
  end
end

```

To check the health check status:

```

# diagnose sys sdwan health-check status 1
Health Check(1):
Seq(1 vd2-1): state(alive), packet-loss(0.000%) latency(0.023), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1
Seq(2 vd2-2): state(alive), packet-loss(0.000%) latency(0.022), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1

```

Example 3

In this example, when traffic from spoke 1 arrives at the hub on tunnel 1, it will egress the hub on tunnel 1 to go to other spokes. If traffic arrives on tunnel 2, it will egress on tunnel 2, and not tunnel 1.

To configure SD-WAN on the hub:

```

config system sdwan
  set status enable
  config zone
    edit "virtual-wan-link"
      set service-sla-tie-break input-device
    next
  end
  config members
    edit 1
      set interface "p1"
    next
    edit 2
      set interface "p2"
    next
  end
  config health-check
    edit "1"
      set server "22.1.1.2"
      set members 1 2

```

```

        config sla
            edit 1
            next
        end
    next
end
config service
    edit 1
        set mode sla
        set dst "all"
        config sla
            edit "1"
                set id 1
            next
        end
        set priority-members 1 2
        set tie-break input-device
    next
end
end

```

To verify that traffic stays in the same overlay on ingress and egress on the hub:

1. Confirm that the SD-WAN service rule has `Tie break` set to `input-device` so that, when SLAs are met on all of the members, traffic prefers to egress on the same member as the input device:

```
# diagnose sys sdwan service
```

```
Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
```

```
Tie break: input-device
```

```
Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
```

```
Members(2):
```

```
1: Seq_num(1 p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
```

```
2: Seq_num(2 p2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
```

```
Dst address(1):
```

```
0.0.0.0-255.255.255.255
```

2. Use `diagnose sniffer packet` commands to verify that traffic ingress and egress are on the same overlay.

Multiple members per SD-WAN neighbor configuration



This information is also available in the FortiOS 7.2 Administration Guide:

- [Using multiple members per SD-WAN neighbor configuration](#)

SD-WAN BGP neighbor configurations are used to define the SLA health check in which an SD-WAN member must meet to qualify as being up. When the SD-WAN member meets the SLA threshold, the FortiGate will apply the route map defined in the BGP neighbor's `route-map-out-preferable` option. If the SD-WAN member fails to meet the SLA, the FortiGate will apply the route map defined in the BGP neighbor's `route-map-out` option instead. This allows the FortiGate to advertise the health of the SD-WAN member to its BGP neighbor by advertising different community strings based on its SLA status.



For more information, refer to the following BGP examples in the FortiOS Administration Guide: [Controlling traffic with BGP route mapping and service rules](#) and [Applying BGP route-map to multiple BGP neighbors](#).

In this enhancement, instead of selecting only one SD-WAN member per neighbor, multiple SD-WAN members can be selected. This allows the SD-WAN neighbor feature to support topologies where there are multiple SD-WAN overlays and/or underlays to a neighbor. The `minimum-sla-meet-members` option is used to configure the minimum number of members that must be in an SLA per neighbor for the preferable route map to be used.

```
config system sdwan
  config neighbor
    edit <ip>
      set member {<seq-num_1>} [<seq-num_2>] ... [<seq-num_n>]
      set minimum-sla-meet-members <integer>
    next
  end
end
```

```
member {<seq-num_1>}
      [<seq-num_2>] ...
      [<seq-num_n>]
```

Enter the member sequence number list. Multiple members can be defined.

```
minimum-sla-meet-members
      <integer>
```

Set the minimum number of members that meet SLA when the neighbor is preferred (1 - 255, default = 1).

- If the number of in SLA members is less than the `minimum-sla-meet-members` value, the default route map will be used.
- If the number of in SLA members is equal or larger than the `minimum-sla-meet-members` value, the preferable route map will be used.

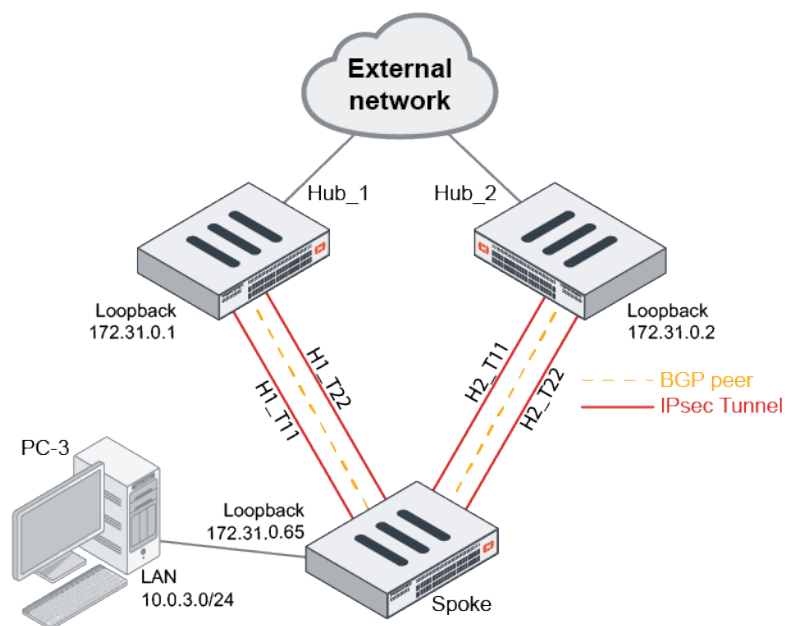
Example

In the following example, the spoke FortiGate has four tunnels: two tunnels to Hub_1 and two tunnels to Hub_2. The spoke has two BGP neighbors: one to Hub_1 and one to Hub-2. BGP neighbors are established on loopback IPs.

The SD-WAN neighbor plus `route-map-out-preferable` configuration is deployed on the spoke to achieve the following:

- If any tunnel to Hub_1 or Hub_2 is in SLA, the preferable route map will be applied on the BGP neighbor to Hub_1 or Hub_2.
- If both tunnels to Hub_1 or Hub_2 are out of SLA, the default route map will be applied on the BGP neighbor to Hub_1 or Hub_2.

The preferable route map and default route map are used to set different custom BGP communities as the spoke advertises its LAN routes to the hub. Each hub can translate communities into different BGP MED or AS prepends and signal them to the external peers to manipulate inbound traffic, thereby routing traffic to the spoke only when the SLAs are met on at least one of two VPN overlays. In this example, community string 10:1 signals to the neighbor that SLAs are met, and 10:2 signals that SLAs are not met.



To configure the BGP route maps and neighbors:

1. Configure an access list of prefixes to be matched:

```
config router access-list
  edit "net10"
    config rule
      edit 1
        set prefix 10.0.3.0 255.255.255.0
      next
    end
  next
end
```

2. Configure route maps for neighbors in SLA (preferable) and out of SLA (default):

```
config router route-map
  edit "in_sla"
    config rule
      edit 1
        set match-ip-address "net10"
        set set-community "10:1"
      next
    end
  next
  edit "out_sla"
    config rule
      edit 1
        set match-ip-address "net10"
        set set-community "10:2"
      next
    end
  next
end
```

3. Configure the BGP neighbors:

```
config router bgp
  set router-id 172.31.0.65
  config neighbor
    edit "172.31.0.1"
      set route-map-out "out_sla"
      set route-map-out-preferable "in_sla"
      set update-source "Loopback0"
    next
    edit "172.31.0.2"
      set route-map-out "out_sla"
      set route-map-out-preferable "in_sla"
      set update-source "Loopback0"
    next
  end
  config network
    edit 1
      set prefix 10.0.3.0 255.255.255.0
    next
  end
end
```

To configure SD-WAN:

1. Configure the SD-WAN members:

```
config system sdwan
  set status enable
  config members
    edit 1
      set interface "H1_T11"
      set source 172.31.0.65
    next
    edit 4
      set interface "H1_T22"
      set source 172.31.0.65
    next
    edit 6
      set interface "H2_T11"
      set source 172.31.0.65
    next
    edit 9
      set interface "H2_T22"
      set source 172.31.0.65
    next
  end
end
```

2. Configure the health check that must be met:

```
config system sdwan
  config health-check
    edit "HUB"
      set server "172.31.100.100"
      set members 0
    config sla
```

```

        edit 1
            set link-cost-factor latency
            set latency-threshold 100
        next
    end
next
end
end

```

3. Configure the SD-WAN neighbors:

```

config system sdwan
config neighbor
    edit "172.31.0.1"
        set member 1 4
        set health-check "HUB"
        set sla-id 1
        set minimum-sla-meet-members 1
    next
    edit "172.31.0.2"
        set member 6 9
        set health-check "HUB"
        set sla-id 1
        set minimum-sla-meet-members 1
    next
end
end

```

To verify that when two members to Hub_1/Hub_2 are in SLA, the preferable route map is be applied on BGP neighbors to Hub_1/Hub_2:

```

Branch1_A_FGT (root) # diagnose sys sdwan health-check
Health Check(HUB):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.209), jitter(0.017), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(4 H1_T22): state(alive), packet-loss(0.000%) latency(0.171), jitter(0.004), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(6 H2_T11): state(alive), packet-loss(0.000%) latency(0.175), jitter(0.014), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(9 H2_T22): state(alive), packet-loss(0.000%) latency(0.176), jitter(0.019), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1

# diagnose sys sdwan neighbor
Neighbor(172.31.0.1): member(1 4 )role(standalone)
    Health-check(HUB:1)  sla-pass selected alive
Neighbor(172.31.0.2): member(6 9 )role(standalone)
    Health-check(HUB:1)  sla-pass selected alive

```

On Hub_1 and Hub_2, the expected communities have been attached into the spoke's LAN route:

```

Hub_1_FGT (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
    Not advertised to any peer
    Original VRF 0
    Local, (Received from a RR-client)
        172.31.0.65 from 172.31.0.65 (172.31.0.65)
            Origin IGP metric 0, localpref 100, valid, internal, best

```

Community: 10:1

Last update: Wed Dec 29 22:38:29 2021

```
Hub_2_FGT (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
Community: 10:1
```

Last update: Wed Dec 29 22:43:10 2021

If one member for each neighbor becomes out of SLA, the preferable route map is still applied:

```
Branch1_A_FGT (root) # diagnose sys sdwan health-check
Health Check(HUB):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(120.207), jitter(0.018), mos
(4.338), bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x0
Seq(4 H1_T22): state(alive), packet-loss(0.000%) latency(0.182), jitter(0.008), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(6 H2_T11): state(alive), packet-loss(0.000%) latency(120.102), jitter(0.009), mos
(4.404), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x0
Seq(9 H2_T22): state(alive), packet-loss(0.000%) latency(0.176), jitter(0.009), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1

# diagnose sys sdwan neighbor
Neighbor(172.31.0.1): member(1 4 )role(standalone)
  Health-check(HUB:1) sla-pass selected alive
Neighbor(172.31.0.2): member(6 9 )role(standalone)
  Health-check(HUB:1) sla-pass selected alive

Hub_1_FGT (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
Community: 10:1
  Last update: Thu Dec 30 10:44:47 2021

Hub_2_FGT (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
Community: 10:1
  Last update: Wed Dec 29 22:43:10 2021
```

If both members for Hub_1 become out of SLA, the default route map is applied:

```

Branch1_A_FGT (root) # diagnose sys sdwan health-check
Health Check(HUB):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(120.194), jitter(0.018), mos
(4.338), bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x0
Seq(4 H1_T22): state(alive), packet-loss(0.000%) latency(120.167), jitter(0.006), mos
(4.338), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x0
Seq(6 H2_T11): state(alive), packet-loss(0.000%) latency(120.180), jitter(0.012), mos
(4.338), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x0
Seq(9 H2_T22): state(alive), packet-loss(0.000%) latency(0.170), jitter(0.005), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1

# diagnose sys sdwan neighbor
Neighbor(172.31.0.1): member(1 4 )role(standalone)
    Health-check(HUB:1) sla-fail alive
Neighbor(172.31.0.2): member(6 9 )role(standalone)
    Health-check(HUB:1) sla-pass selected alive

Hub_1_FGT (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
    Not advertised to any peer
    Original VRF 0
    Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
    Origin IGP metric 0, localpref 100, valid, internal, best
    Community: 10:2
    Last update: Thu Dec 30 10:57:33 2021

Hub_2_FGT (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
    Not advertised to any peer
    Original VRF 0
    Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
    Origin IGP metric 0, localpref 100, valid, internal, best
    Community: 10:1
    Last update: Wed Dec 29 22:43:10 2021

```

SD-WAN in large scale deployments



This information is also available in the FortiOS 7.2 Administration Guide:

- [SD-WAN in large scale deployments](#)

SD-WAN with ADVPN configurations in large-scale deployments is improved.

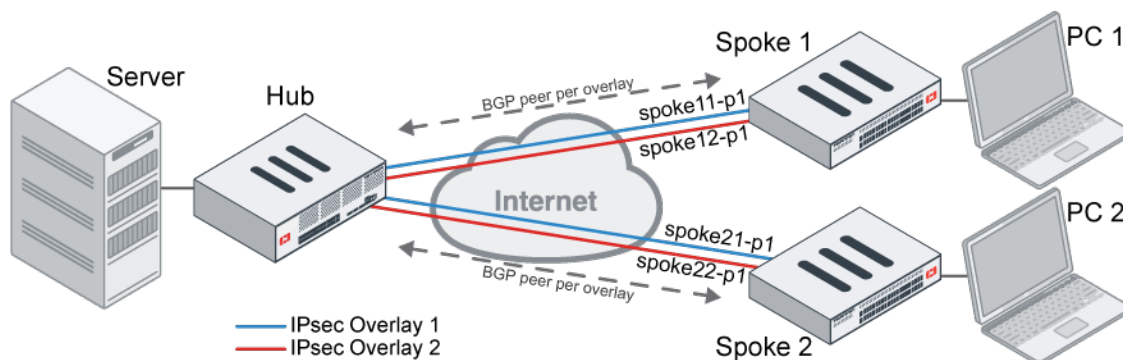
- Phase 2 selectors can be used to inject IKE routes on the ADVPN shortcut tunnel.
When configuration method (`mode-cfg`) is enabled in IPsec phase 1 configuration, enabling `mode-cfg-allow-client-selector` allows custom phase 2 selectors to be configured. By also enabling the addition of a route to the peer destination selector (`add-route`) in the phase 1 configuration, IKE routes based on the phase 2 selectors can be injected. This means that routes do not need to be reflected on the hub to propagate them between spokes,

avoiding possible BGP daemon process load issues and improving network scalability in a large-scale ADVPN network.

- Route map rules can apply priorities to BGP routes.

On the hub, priorities can be set in a route map's rules, and the route map can be applied on BGP routes. This allows the hub to mark the preferred path learned from the spokes with a priority value (lower priority is preferred), instead of using multiple SD-WAN policy routes on the hub. When a preferred outbound route map (`route-map-out-preferable`) is also configured in an SD-WAN neighbor on the spoke, deploying SD-WAN rules on the hub to steer traffic from the hub to a spoke is unnecessary.

- SD-WAN members' local cost can be exchanged on the ADVPN shortcut tunnel so that spokes can use the remote cost as tiebreak to select a preferred shortcut. If multiple shortcuts originate from the same member to different members on the same remote spoke, then the remote cost on the shortcuts is used as the tiebreak to decide which shortcut is preferred.



In this example, SD-WAN is configured on an ADVPN network with a BGP neighbor per overlay.

Instead of reflecting BGP routes with the route-reflector on the hub, when the shortcuts are triggered, IKE routes on the shortcuts are directly injected based on the configured phase 2 selectors to allow routes to be exchanged between spokes.

Routes between the hub and the spokes are exchanged by BGP, and the spokes use the default route to send spoke-to-spoke traffic to the hub and trigger the shortcuts.

Instead of configuring SD-WAN rules on the hub, different priorities are configured on the BGP routes by matching different BGP communities to steer traffic from the hub to the spokes.

To configure Spoke 1:

1. Configure phase 1:

```
config vpn ipsec phase1-interface
edit "spoke11-p1"
...
set ike-version 2
set net-device enable
set add-route enable
set mode-cfg enable
set auto-discovery-receiver enable
set mode-cfg-allow-client-selector enable
...
next
edit "spoke12-p1"
...
```

```

        set ike-version 2
        set net-device enable
        set add-route enable
        set mode-cfg enable
        set auto-discovery-receiver enable
        set mode-cfg-allow-client-selector enable
    next
end

```

2. Configure phase 2:

```

config vpn ipsec phase2-interface
    edit "spoke11-p2"
        ...
        set src-name "LAN_Net"
        set dst-name "all"
    next
    edit "spoke12-p2"
        ...
        set src-name "LAN_Net"
        set dst-name "all"
    next
end

```

3. Configure an address group:

Spoke 1 uses LAN subnet 10.1-3.100.0/24.

```

config firewall addrgrp
    edit "LAN_Net"
        set member "10.1.100.0" "10.2.100.0" "10.3.100.0"
    next
end

```

4. Configure route maps:

- If overlay 1 to the hub is in SLA, attach "65000:1" to the BGP routes advertised to the hub over overlay 1.
- If overlay 2 to the hub is in SLA, attach "65000:2" to the BGP routes advertised to the hub over overlay 2.
- If any overlay to the hub is out of SLA, attach "65000:9999" to the BGP routes advertised to the hub over any overlay.

```

config router route-map
    edit "HUB_CARRIER1"
        config rule
            edit 1
                set set-community "65000:1"
                ...
            next
        end
        ...
    next
    edit "HUB_CARRIER2"
        config rule
            edit 1
                set set-community "65000:2"
                ...
            next
        end
        ...

```

```
    next
    edit "HUB_BAD"
        config rule
            edit 1
                set set-community "65000:9999"
                ...
            next
        end
        ...
    next
end
```

5. Configure BGP and SD-WAN members and neighbors:

```
config router bgp
    set as 65412
    config neighbor
        edit "10.10.15.253"
            set remote-as 65412
            set route-map-out "HUB_BAD"
            set route-map-out-preferable "HUB_CARRIER1"
            ...
        next
        edit "10.10.16.253"
            set remote-as 65412
            set route-map-out "HUB_BAD"
            set route-map-out-preferable "HUB_CARRIER2"
            ...
        next
    end
end

config system sdwan
    config members
        edit 1
            set interface "spoke11-p1"
        next
        edit 2
            set interface "spoke12-p1"
        next
    end
    config neighbor
        edit "10.10.15.253"
            set member 1
            set health-check "1"
            set sla-id 1
        next
        edit "10.10.16.253"
            set member 2
            set health-check "11"
            set sla-id 1
        next
    end
end
```

To configure Spoke 2:

1. Configure phase 1:

```
config vpn ipsec phase1-interface
    edit "spoke21-p1"
        ...
        set ike-version 2
        set net-device enable
        set add-route enable
        set mode-cfg enable
        set auto-discovery-receiver enable
        set mode-cfg-allow-client-selector enable
        ...
    next
    edit "spoke22-p1"
        ...
        set ike-version 2
        set net-device enable
        set add-route enable
        set mode-cfg enable
        set auto-discovery-receiver enable
        set mode-cfg-allow-client-selector enable
    next
end
```

2. Configure phase 2:

```
config vpn ipsec phase2-interface
    edit "spoke21-p2"
        ...
        set src-name "LAN_Net"
        set dst-name "all"
    next
    edit "spoke22-p2"
        ...
        set src-name "LAN_Net"
        set dst-name "all"
    next
end
```

3. Configure an address group:

Spoke 2 uses LAN subnet 192.168.5-7.0/24.

```
config firewall addrgrp
    edit "LAN_Net"
        set member "192.168.5.0" "192.168.6.0" "192.168.7.0"
    next
end
```

4. Configure route maps:

- If overlay 1 to the hub is in SLA, attach "65000:1" to the BGP routes advertised to the hub over overlay 1.
- If overlay 2 to the hub is in SLA, attach "65000:2" to the BGP routes advertised to the hub over overlay 2.
- If any overlay to the hub is out of SLA, attach "65000:9999" to the BGP routes advertised to the hub over any overlay.

```
config router route-map
  edit "HUB_CARRIER1"
    config rule
      edit 1
        set set-community "65000:1"
        ...
      next
    end
    ...
  next
  edit "HUB_CARRIER2"
    config rule
      edit 1
        set set-community "65000:2"
        ...
      next
    end
    ...
  next
  edit "HUB_BAD"
    config rule
      edit 1
        set set-community "65000:9999"
        ...
      next
    end
    ...
  next
end
```

5. Configure BGP and SD-WAN members and neighbors:

```
config router bgp
  set as 65412
  config neighbor
    edit "10.10.15.253"
      set remote-as 65412
      set route-map-out "HUB_BAD"
      set route-map-out-preferable "HUB_CARRIER1"
      ...
    next
    edit "10.10.16.253"
      set remote-as 65412
      set route-map-out "HUB_BAD"
      set route-map-out-preferable "HUB_CARRIER2"
      ...
    next
  end
end

config system sdwan
  config members
    edit 1
      set interface "spoke21-p1"
      set cost 100
    next
    edit 2
```

```

        set interface "spoke22-p1"
        set cost 200
    next
end
config neighbor
    edit "10.10.15.253"
        set member 1
        set health-check "1"
        set sla-id 1
    next
    edit "10.10.16.253"
        set member 2
        set health-check "11"
        set sla-id 1
    next
end
end

```

To configure the hub:

1. Configure the route maps:

- Set the priority to 100 for routes with community 65000:1, indicating that they are in SLA for overlay 1.
- Set the priority to 200 for routes with community 65000:2, indicating that they are in SLA for overlay 2.
- Set the priority to 9999 for routes with community 65000:9999, indicating that they are out of SLA for any overlay.

```

config router route-map
    edit "Set_Pri"
        config rule
            edit 1
                set match-community "comm_65000:1"
                set set-priority 100
            next
            edit 2
                set match-community "comm_65000:2"
                set set-priority 200
            next
            edit 3
                set match-community "comm_65000:9999"
                set set-priority 9999
            next
        end
    next
end

```

2. Configure BGP:

```

config router bgp
    set as 65412
    config neighbor-group
        edit "advpn"
            set remote-as 65412
            set route-map-in "Set_Pri"
            ...
        next
        edit "advpn2"
    end
end

```

```

        set remote-as 65412
        set route-map-in "Set_Pri"
        ...
    next
end
config neighbor-range
    edit 1
        set prefix 10.10.15.0 255.255.255.0
        set neighbor-group "advpn"
    next
    edit 2
        set prefix 10.10.16.0 255.255.255.0
        set neighbor-group "advpn2"
    next
end
end

```

To test the configuration:

1. Check the routing tables on the spokes:

Spoke 1:

```

spoke-1 (root) # get router info routing-table all
B*      0.0.0.0/0 [200/0] via 10.10.15.253 (recursive is directly connected, spoke11-
p1), 00:01:17, [1/0]           // default route to hub
                        [200/0] via 10.10.16.253 (recursive is directly connected,
spoke12-p1), 00:01:17, [1/0]
B       9.0.0.0/24 [200/0] via 10.10.15.253 (recursive is directly connected, spoke11-
p1), 00:01:17, [1/0]           // route to the server behind hub
                        [200/0] via 10.10.16.253 (recursive is directly connected,
spoke12-p1), 00:01:17, [1/0]
C       10.1.100.0/24 is directly connected, port2           // route to PC 1
C       10.10.15.0/24 is directly connected, spoke11-p1      // overlay 1
C       10.10.15.1/32 is directly connected, spoke11-p1
C       10.10.16.0/24 is directly connected, spoke12-p1      // overlay 2
C       10.10.16.1/32 is directly connected, spoke12-p1

```

Spoke 2:

```

spoke-2 (root) # get router info routing-table all
B*      0.0.0.0/0 [200/0] via 10.10.15.253 (recursive is directly connected, spoke21-
p1), 00:46:14, [1/0]           // default route to hub
                        [200/0] via 10.10.16.253 (recursive is directly connected,
spoke22-p1), 00:46:14, [1/0]
B       9.0.0.0/24 [200/0] via 10.10.15.253 (recursive is directly connected, spoke21-
p1), 00:46:18, [1/0]           // route to the server behind hub
                        [200/0] via 10.10.16.253 (recursive is directly connected,
spoke22-p1), 00:46:18, [1/0]
C       10.10.15.0/24 is directly connected, spoke21-p1      // overlay 1
C       10.10.15.2/32 is directly connected, spoke21-p1
C       10.10.16.0/24 is directly connected, spoke22-p1      // overlay 2
C       10.10.16.2/32 is directly connected, spoke22-p1
C       192.168.5.0/24 is directly connected, port2          // route to PC 2

```

2. Send traffic from PC 1 to PC 2 and trigger the shortcut:

The IKE routes on the shortcut are directly injected based on the phase 2 selectors, and spoke-to-spoke traffic then goes directly through the shortcut instead of going through the hub.

Spoke 1:

```
spoke-1 (root) # get router info routing-table static
S      192.168.5.0/24 [15/0] via spoke11-p1_0 tunnel 172.16.200.4 vrf 0, [1/0]
S      192.168.6.0/24 [15/0] via spoke11-p1_0 tunnel 172.16.200.4 vrf 0, [1/0]
S      192.168.7.0/24 [15/0] via spoke11-p1_0 tunnel 172.16.200.4 vrf 0, [1/0]

spoke-1 (root) # diagnose sniffer packet any 'host 192.168.5.44' 4
interfaces=[any]
filters=[host 192.168.5.44]
1.446306 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
1.446327 spoke11-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
1.446521 spoke11-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
1.446536 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
```

Spoke 2:

```
spoke-2 (root) # get router info routing-table static
S      10.1.100.0/24 [15/0] via spoke21-p1_0 tunnel 10.10.15.1 vrf 0, [1/0]
S      10.2.100.0/24 [15/0] via spoke21-p1_0 tunnel 10.10.15.1 vrf 0, [1/0]
S      10.3.100.0/24 [15/0] via spoke21-p1_0 tunnel 10.10.15.1 vrf 0, [1/0]
```

3. Confirm that the overlays are in SLA on the spokes:**Spoke 1:**

```
spoke-1 (root) # diagnose sys sdwan neighbor
Neighbor(10.10.15.253): member(1)role(standalone)
Health-check(1:1) sla-pass selected alive
Neighbor(10.10.16.253): member(2)role(standalone)
Health-check(11:1) sla-pass selected alive
```

Spoke 2:

```
spoke-2 (root) # diagnose sys sdwan neighbor
Neighbor(10.10.15.253): member(1)role(standalone)
Health-check(1:1) sla-pass selected alive
Neighbor(10.10.16.253): member(2)role(standalone)
Health-check(11:1) sla-pass selected alive
```

4. On the hub, check that the routes received from the spokes have the expected priorities:

```
hub (root) # diagnose ip route list | grep proto=11
tab=254 vf=0 scope=0 type=1 proto=11 prio=100 0.0.0.0/0.0.0.0/0->10.1.100.0/24
pref=0.0.0.0 gwy=10.10.15.1 dev=101(hub-phase1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=200 0.0.0.0/0.0.0.0/0->10.1.100.0/24
pref=0.0.0.0 gwy=10.10.16.1 dev=102(hub2-phase1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=100 0.0.0.0/0.0.0.0/0->192.168.5.0/24
pref=0.0.0.0 gwy=10.10.15.2 dev=101(hub-phase1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=200 0.0.0.0/0.0.0.0/0->192.168.5.0/24
pref=0.0.0.0 gwy=10.10.16.2 dev=102(hub2-phase1)
```

The priority set by the hub's route map is based on the community string received from the spoke. The route with a lower priority value is selected, so traffic to Spoke 1 goes out on the hub-phase1 tunnel:

```
hub (root) # diagnose sniffer packet any 'host 9.0.0.2' 4
interfaces=[any]
filters=[host 9.0.0.2]
2.735456 R190 in 9.0.0.2 -> 10.1.100.22: icmp: echo request
2.735508 hub-phase1 out 9.0.0.2 -> 10.1.100.22: icmp: echo request
```

```
2.735813 hub-phase1 in 10.1.100.22 -> 9.0.0.2: icmp: echo reply
2.735854 R190 out 10.1.100.22 -> 9.0.0.2: icmp: echo reply
```

5. If overlay 1 goes out of SLA, the priorities of the routes on the hub are updated and traffic from the hub to Spoke 1 goes through overlay 2:

Spoke 1:

```
spoke-1 (root) # diagnose sys sdwan neighbor
Neighbor(10.10.15.253): member(1)role(standalone)
Health-check(1:1) sla-fail alive
Neighbor(10.10.16.253): member(2)role(standalone)
Health-check(11:1) sla-pass selected alive
```

Spoke 2:

```
spoke-2 (root) # diagnose sys sdwan neighbor
Neighbor(10.10.15.253): member(1)role(standalone)
Health-check(1:1) sla-fail alive
Neighbor(10.10.16.253): member(2)role(standalone)
Health-check(11:1) sla-pass selected alive
```

Hub:

```
hub (root) # diagnose ip route list | grep proto=11
tab=254 vf=0 scope=0 type=1 proto=11 prio=200 0.0.0.0/0.0.0.0/0->10.1.100.0/24
pref=0.0.0.0 gwy=10.10.16.1 dev=102(hub2-phase1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=9999 0.0.0.0/0.0.0.0/0->10.1.100.0/24
pref=0.0.0.0 gwy=10.10.15.1 dev=101(hub-phase1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=200 0.0.0.0/0.0.0.0/0->192.168.5.0/24
pref=0.0.0.0 gwy=10.10.16.2 dev=102(hub2-phase1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=9999 0.0.0.0/0.0.0.0/0->192.168.5.0/24
pref=0.0.0.0 gwy=10.10.15.2 dev=101(hub-phase1)

hub (root) # diagnose sniffer packet any 'host 9.0.0.2' 4
interfaces=[any]
filters=[host 9.0.0.2]
3.550181 R190 in 9.0.0.2 -> 10.1.100.22: icmp: echo request
3.550234 hub2-phase1 out 9.0.0.2 -> 10.1.100.22: icmp: echo request
3.550713 hub2-phase1 in 10.1.100.22 -> 9.0.0.2: icmp: echo reply
3.550735 R190 out 10.1.100.22 -> 9.0.0.2: icmp: echo reply
```

6. Trigger shortcuts between Spoke 1 and Spoke 2:

- Shortcuts spoke11-p1_1 and spoke11-p1_0 originate from spoke11-p1.
- spoke11-p1_1 corresponds to spoke21-p1_0 on Spoke 2.
- spoke11-p1_0 corresponds to spoke22-p1_0 on Spoke 2.

Spoke 1:

```
spoke-1 (root) # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Tie break: cfg
Gen(12), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-number
Service role: standalone
Member sub interface(4):
3: seq_num(1), interface(spoke11-p1):
1: spoke11-p1_0(75)
2: spoke11-p1_1(76)
```

```

Members(4):
  1: Seq_num(1 spoke11-p1_1), alive, sla(0x1), gid(0), remote cost(100), cfg_order(0),
local cost(0), selected
  2: Seq_num(1 spoke11-p1_0), alive, sla(0x1), gid(0), remote cost(200), cfg_order(0),
local cost(0), selected
  3: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
  4: Seq_num(2 spoke12-p1), alive, sla(0x2), gid(0), cfg_order(1), local cost(0),
selected
Src address(1):
  10.1.100.0-10.1.100.255

Dst address(1):
  0.0.0.0-255.255.255.255

```

Spoke 2:

```

spoke-2 (root) # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Tie break: cfg
Gen(9), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-number
Service role: standalone
Member sub interface(4):
  2: seq_num(1), interface(spoke21-p1):
    1: spoke21-p1_0(68)
  4: seq_num(2), interface(spoke22-p1):
    1: spoke22-p1_0(67)
Members(4):
  1: Seq_num(1 spoke21-p1_0), alive, sla(0x1), gid(0), cfg_order(0), local cost(100),
selected
  2: Seq_num(1 spoke21-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(100),
selected
  3: Seq_num(2 spoke22-p1_0), alive, sla(0x2), gid(0), cfg_order(1), local cost(200),
selected
  4: Seq_num(2 spoke22-p1), alive, sla(0x2), gid(0), cfg_order(1), local cost(200),
selected
Src address(1):
  192.168.5.0-192.168.5.255

Dst address(1):
  0.0.0.0-255.255.255.255

```

7. On Spoke 2, increase the cost of spoke21-p1_0 to 300.

```

spoke-2 (root) # config system sdwan
  config members
    edit 1
      set interface "spoke21-p1"
      set cost 300
    next
  end
end

```

The new cost is learned by the spoke11-p1_1 shortcut on Spoke 1, and that shortcut is no longer preferred due to its higher remote cost:

Spoke 1:

```
spoke-1 (root) # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Tie break: cfg
Gen(13), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-number
Service role: standalone
Member sub interface(4):
  3: seq_num(1), interface(spoke11-p1):
    1: spoke11-p1_0(78)
    2: spoke11-p1_1(79)
Members(4):
  1: Seq_num(1 spoke11-p1_0), alive, sla(0x1), gid(0), remote cost(200), cfg_order(0),
local cost(0), selected
  2: Seq_num(1 spoke11-p1_1), alive, sla(0x1), gid(0), remote cost(300), cfg_order(0),
local cost(0), selected
  3: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
  4: Seq_num(2 spoke12-p1), alive, sla(0x2), gid(0), cfg_order(1), local cost(0),
selected
Src address(1):
  10.1.100.0-10.1.100.255

Dst address(1):
  0.0.0.0-255.255.255.255
```

Spoke 2:

```
spoke-2 (root) # diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Tie break: cfg
Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-number
Service role: standalone
Member sub interface(4):
  2: seq_num(2), interface(spoke22-p1):
    1: spoke22-p1_0(70)
  4: seq_num(1), interface(spoke21-p1):
    1: spoke21-p1_0(71)
Members(4):
  1: Seq_num(2 spoke22-p1_0), alive, sla(0x2), gid(0), cfg_order(1), local cost(200),
selected
  2: Seq_num(2 spoke22-p1), alive, sla(0x2), gid(0), cfg_order(1), local cost(200),
selected
  3: Seq_num(1 spoke21-p1_0), alive, sla(0x1), gid(0), cfg_order(0), local cost(300),
selected
  4: Seq_num(1 spoke21-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(300),
selected
Src address(1):
  192.168.5.0-192.168.5.255

Dst address(1):
  0.0.0.0-255.255.255.255
```

Route map rules and BGP routes

Route map rules can apply priorities to BGP routes. On the hub, priorities can be set in a route map's rules, and the route map can be applied on BGP routes. This allows the hub to mark the preferred path learned from the spokes with a priority value (lower priority is preferred), instead of using multiple SD-WAN policy routes on the hub. When a preferred outbound route map (`route-map-out-preferable`) is also configured in an SD-WAN neighbor on the spoke, deploying SD-WAN rules on the hub to steer traffic from the hub to a spoke is unnecessary.

For details, see [SD-WAN in large scale deployments on page 85](#).

BGP socket limit increase

The BGP socket limit has been increased to 80,000.

See also [SD-WAN in large scale deployments on page 85](#).

IKE embryonic limit

Administrators can configure the maximum number of IPsec tunnels to simultaneously negotiate.

To configure the embryonic limit:

```
config system ike
    set embryonic-limit {integer}
end
```

See also [SD-WAN in large scale deployments on page 85](#).

GUI support for advanced BGP options - 7.2.1



See also the FortiOS 7.2 Administration Guide:

- [Advanced routing](#)

Advanced BGP options can be configured in the GUI on the *Network > BGP* page, including: the BGP neighbor local AS, hold time timer, keepalive timer, and enforcing eBGP multihop. The *View in Routing Monitor* buttons in the right-side of the screen can display the BGP neighbors list, the BGP IPv4 routing table, or the BGP IPv6 routing table in a slide-out window instead of redirecting to the monitor page. The *Routing* monitor includes an option to soft reset a neighbor from the BGP neighbors list.

BGP page enhancements

The *View in Routing Monitor* button is available to view neighbors, paths, and IPv6 paths.

Local BGP Options

Local AS: 65412
Router ID: 1.1.1.1

Neighbors

IP	Remote AS
2.2.2.2	65412
3.3.3.3	65412
10.100.1.1	20
6.6.6.6	20
10.100.1.5	20
2000::2:2:2	65412
2000::3:3:3	65412

Neighbor Groups

No results

Neighbor Ranges

Apply

Neighbors

2.2.2.2
3.3.3.3
6.6.6.6
10.100.1.1
10.100.1.5

View in Routing Monitor

Paths

0.0.0.0
0.0.0.0
1.1.1.1
1.3.3.0
1.3.3.0

View in Routing Monitor

IPv6 Paths

2000:172:27:1::

View in Routing Monitor

Additional Information

API Preview
> Edit in CLI

Documentation
Online Help
Video Tutorials

- *Routing* monitor dropdown selected to display *BGP Neighbors*:

Routing

Soft reset View Search

BGP Neighbors

Neighbor IP	Local IP	Remote AS	State	Admin Status
IPv4				
2.2.2.2	1.1.1.1	65412	Established	Enabled
3.3.3.3	1.1.1.1	65412	Established	Enabled
6.6.6.6	0.0.0.0	20	Idle	Disabled
10.100.1.1	0.0.0.0	20	Active	Enabled
10.100.1.5	10.100.1.6	20	Established	Enabled
IPv6				
2000::2:2:2	2000::1:1:1	65412	Established	Enabled
2000::3:3:3	::	65412	Active	Enabled

Updated: 17:20:05

When a neighbor is selected, the *Soft reset* option is available.

- *Routing* monitor dropdown selected to display *BGP Paths*:

Local BGP Options

Local AS: 65412
Router ID: 1.1.1.1

Neighbors

IP	Remote AS
2.2.2.2	65412
3.3.3.3	65412
10.100.1.1	20
6.6.6.6	20
10.100.1.5	20
2000::2:2:2	65412
2000::3:3:3	65412

Neighbor Groups

Name	Remote AS
No results	

Neighbor Ranges

Range	Remote AS
No results	

Routing

View

Search

Prefix	Learned From	Next Hop	Origin	Best Path
0.0.0.0/0	3.3.3.3	3.3.3.3	IGP	No
0.0.0.0/0	0.0.0.0	10.100.1.249	Incomplete	Yes
1.1.1.1/32	0.0.0.0	0.0.0.0	Incomplete	Yes
1.3.3.0/24	2.2.2.2	2.2.2.2	EGP	No
1.3.3.0/24	0.0.0.0	10.100.1.25	EGP	No
1.3.3.0/24	10.100.1.5	10.100.1.5	EGP	Yes
6.6.6.6/32	0.0.0.0	10.100.1.5	Incomplete	Yes
10.1.100.0/24	0.0.0.0	172.16.203.2	Incomplete	Yes
10.100.1.4/30	0.0.0.0	0.0.0.0	Incomplete	Yes
10.100.1.248/29	0.0.0.0	0.0.0.0	Incomplete	Yes
10.100.10.0/24	2.2.2.2	2.2.2.2	EGP	No
10.100.10.0/24	0.0.0.0	10.100.1.25	EGP	No
10.100.10.0/24	10.100.1.5	10.100.1.5	EGP	Yes
10.100.11.0/24	2.2.2.2	2.2.2.2	EGP	No
10.100.11.0/24	0.0.0.0	10.100.1.25	EGP	No
10.100.11.0/24	10.100.1.5	10.100.1.5	EGP	Yes
172.16.95.0/24	0.0.0.0	172.16.200.254	Incomplete	Yes
172.16.100.71/32	0.0.0.0	172.16.200.254	Incomplete	Yes
172.16.200.0/24	0.0.0.0	0.0.0.0	Incomplete	Yes
172.16.203.0/24	0.0.0.0	0.0.0.0	Incomplete	Yes
172.16.204.0/24	0.0.0.0	172.16.200.4	Incomplete	Yes
172.16.205.0/24	0.0.0.0	0.0.0.0	Incomplete	Yes

0% Updated: 17:21:37

- Routing monitor dropdown selected to display *IPv6 BGP Paths*:

Local BGP Options

Local AS: 65412
Router ID: 1.1.1.1

Neighbors

IP	Remote AS
2.2.2.2	65412
3.3.3.3	65412
10.100.1.1	20
6.6.6.6	20
10.100.1.5	20
2000::2:2:2	65412
2000::3:3:3	65412

Neighbor Groups

Name	Remote AS
No results	

Neighbor Ranges

Range	Remote AS
No results	

Routing

View

Search

Prefix	Learned From	Next Hop Local	Next Hop Global	Origin	Best Path
2000::172:27:1::/64	2000::2:2:2	::	2000::2:2:2	IGP	Yes

Updated: 17:22:48

Neighbor configuration page enhancements

There are fields to enter the *Local AS*, *Keep alive timer*, *Hold time timer*, and *Enforce eBGP multihop*.

Support BGP AS number input in asdot and asdot+ format - 7.2.1

BGP Autonomous System (AS) numbers can be inputted in asdot and asdot+ format in compliance with [RFC 5396](#) when configuring the following in the CLI.

- BGP AS, neighbor local and remote AS, and neighbor group local and remote AS:

```
config router bgp
    set as <string>
    config neighbor
        edit <ip>
            set remote-as <string>
            set local-as <string>
        next
    end
    config neighbor-group
        edit <name>
            set remote-as <string>
            set local-as <string>
        next
    end
end
```

as <string>

Enter the router AS number in asplain (1 - 4294967295), asdot, or asdot+ format. Enter 0 to disable BGP.

remote-as <string>

Enter a value in asplain (1 - 4294967295), asdot, or asdot+ format.

local-as <string>

Enter a value in asplain (1 - 4294967295), asdot, or asdot+ format.

- Route map AS path:

```
config router route-map
  edit <name>
    config rule
      edit <id>
        set set-aspath <string>
      next
    end
  next
end
```

```
set-aspath <string>
```

Enter a value in asplain (1 - 4294967295), asdot, or asdot+ format.



get router info bgp summary and other BGP router commands still display the AS numbers in asplain format.

Example

In this example, neighbor 1.1.1.1's remote AS is configured in asdot format. Neighbor 172.16.201.2's remote AS is configured in asdot format, and the local AS in asplain format.

To configure the AS in asdot and asplain formats:

```
config router bgp
  set as 65535.65535
  set router-id 3.3.3.3
  config neighbor
    edit "1.1.1.1"
      set remote-as 65535.65535
    next
    edit "172.16.201.2"
      set remote-as 65050
      set local-as 65516.65516
    next
  end
end
```

To verify the BGP neighbors and routing table:

```
# get router info bgp summary
VRF 0 BGP router identifier 3.3.3.3, local AS number 4294967295
BGP table version is 4
3 BGP AS-PATH entries
0 BGP community entries

Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
1.1.1.1 4 4294967295 21 18 3 0 0 00:04:09 13
172.16.201.2 4 65050 24 28 4 0 0 00:05:42 4

Total number of neighbors 2
```

The BGP AS number 65535 . 65535 in asdot format corresponds to AS number 4294967295 in asplain format in this output.

Support cross-VRF local-in and local-out traffic for local services - 7.2.1



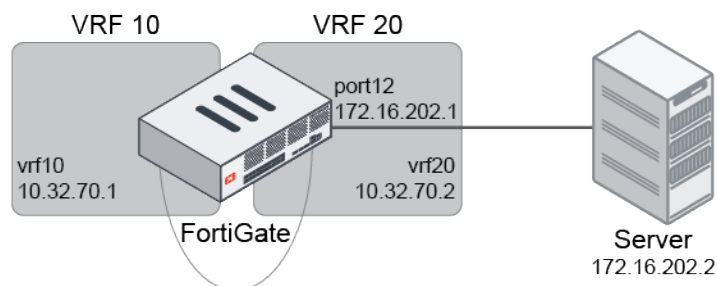
This information is also available in the FortiOS 7.2 Administration Guide:

- [Support cross-VRF local-in and local-out traffic for local services](#)

When local-out traffic such as SD-WAN health checks, SNMP, syslog, and so on are initiated from an interface on one VRF and then pass through interfaces on another VRF, the reply traffic will be successfully forwarded back to the original VRF.

Example

In this example, there is an NPU VDOM link that is configured on the root VDOM. Two VLANs, vrf10 and vrf20, are created on either ends of the NPU VDOM link, each belonging to a different VRF.



When ping from the vrf10 interface in VRF 10 to the destination server 172.16.202.2, since there is a single static route for VRF 10 with a gateway of vrf20/10.32.70.2, traffic is sent to the next hop and subsequently routed through port12 to the server.

As seen in the sniffer trace, the ICMP replies are received on port12 in VRF 20, then pass through vrf20, and are ultimately forwarded back to vrf10 in VRF 10. The traffic flow demonstrates that local-out traffic sourced from one VRF passing through another VRF can return back to the original VRF.

To configure cross-VRF local-out traffic for local services:

1. Configure the interfaces:

```
config system interface
  edit "vrf10"
    set vdom "root"
    set vrf 10
    set ip 10.32.70.1 255.255.255.0
    set allowaccess ping
    set device-identification enable
```

```

        set role lan
        set snmp-index 35
        set interface "np0_vlink0"
        set vlanid 22
    next
    edit "vrf20"
        set vdom "root"
        set vrf 20
        set ip 10.32.70.2 255.255.255.0
        set allowaccess ping
        set device-identification enable
        set role lan
        set snmp-index 36
        set interface "np0_vlink1"
        set vlanid 22
    next
    edit "port12"
        set vdom "root"
        set vrf 20
        set ip 172.16.202.1 255.255.255.0
        set allowaccess ping https ssh snmp http telnet fgfm radius-acct probe-response
fabric ftn speed-test
        set type physical
        set alias "TO_FGT_D_port22"
        set snmp-index 14
        config ipv6
            set ip6-address 2003:172:16:202::1/64
            set ip6-allowaccess ping
        end
    next
end

```

2. Configure the firewall policy:

```

config firewall policy
    edit 1
        set srcintf "vrf20"
        set dstintf "port12"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
    next
end

```

3. Configure the static route:

```

config router static
    edit 2
        set gateway 10.32.70.2
        set distance 3
        set device "vrf10"
    next
end

```

To test the configuration:

1. Execute a ping from the vrf10 interface in VRF 10 to the destination server (172.16.202.2):

```
# execute ping-options interface vrf10
# execute ping 172.16.202.2
PING 172.16.202.2 (172.16.202.2): 56 data bytes
64 bytes from 172.16.202.2: icmp_seq=0 ttl=254 time=0.1 ms
64 bytes from 172.16.202.2: icmp_seq=1 ttl=254 time=0.0 ms

--- 172.16.202.2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.0/0.0/0.1 ms
```

2. Run a sniffer trace on 172.16.202.2 for ICMP:

```
# diagnose sniffer packet any "host 172.16.202.2 and icmp" 4
interfaces=[any]
filters=[host 172.16.202.2 and icmp]
3.393920 vrf10 out 10.32.70.1 -> 172.16.202.2: icmp: echo request
3.393922 npu0_vlink0 out 10.32.70.1 -> 172.16.202.2: icmp: echo request
3.393927 vrf20 in 10.32.70.1 -> 172.16.202.2: icmp: echo request
3.393943 port12 out 10.32.70.1 -> 172.16.202.2: icmp: echo request
3.393977 port12 in 172.16.202.2 -> 10.32.70.1: icmp: echo reply
3.393987 vrf20 out 172.16.202.2 -> 10.32.70.1: icmp: echo reply
3.393988 npu0_vlink1 out 172.16.202.2 -> 10.32.70.1: icmp: echo reply
3.393993 vrf10 in 172.16.202.2 -> 10.32.70.1: icmp: echo reply
4.393941 vrf10 out 10.32.70.1 -> 172.16.202.2: icmp: echo request
4.393942 npu0_vlink0 out 10.32.70.1 -> 172.16.202.2: icmp: echo request
4.393948 vrf20 in 10.32.70.1 -> 172.16.202.2: icmp: echo request
4.393957 port12 out 10.32.70.1 -> 172.16.202.2: icmp: echo request
4.393980 port12 in 172.16.202.2 -> 10.32.70.1: icmp: echo reply
4.393987 vrf20 out 172.16.202.2 -> 10.32.70.1: icmp: echo reply
4.393987 npu0_vlink1 out 172.16.202.2 -> 10.32.70.1: icmp: echo reply
4.393994 vrf10 in 172.16.202.2 -> 10.32.70.1: icmp: echo reply
```

Matching BGP extended community route targets in route maps - 7.2.4



This information is also available in the FortiOS 7.2 Administration Guide:

- [Matching BGP extended community route targets in route maps](#)

BGP extended community route targets can be matched in route maps. This can be applied in a scenario where the BGP route reflector receives routes from many VRFs, and instead of reflecting all routes from all VRFs, users only want to reflect routes based on a specific extended community route target.

To configure the extended community list:

```
config router extcommunity-list
edit <name>
```

```

    set type {standard | expanded}
  config rule
    edit <id>
      set action {deny | permit}
      set type {rt | soo}
      set match <extended_community_specifications>
      set regexp <ordered_list_of_attributes>
    next
  end
next
end

```

type {standard expanded}	Set the extended community list type (standard or expanded).
action {deny permit}	Deny or permit route-based operations based on the route's extended community attribute.
type {rt soo}	Set the extended community type: <ul style="list-style-type: none"> rt: route target soo: site of origin
match <extended_community_specifications>	Set the extended community specifications for matching a reserved extended community (community number in AA:NN format; use quotation marks complex expressions, "123:234 345:456").
regexp <ordered_list_of_attributes>	Set the ordered list of extended community attributes as a regular expression.

To configure the BGP extended community list in the route map:

```

config router route-map
  edit <name>
    config rule
      edit <id>
        set match-extcommunity <list>
        set match-extcommunity-exact {enable | disable}
      next
    end
  next
end

```

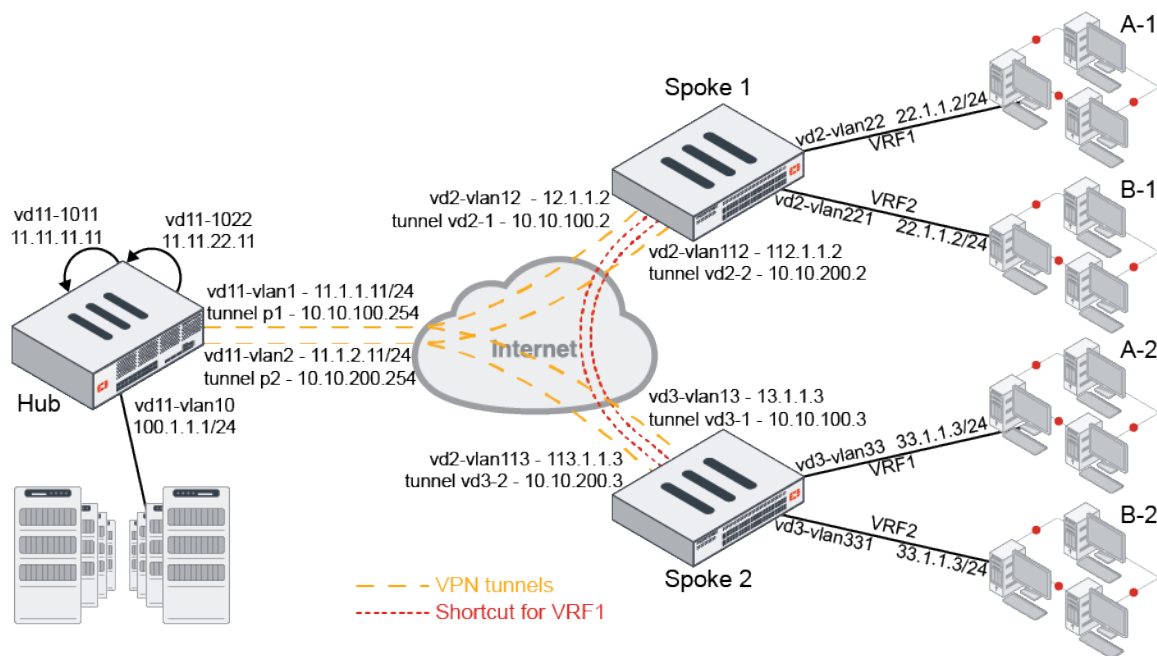
match-extcommunity <list>	Set the BGP extended community list to match to.
match-extcommunity-exact {enable disable}	Enable/disable exact matching of extended communities.

Example

In this example, multiple companies (or departments of a company) share the same hub and spoke VPN infrastructure. Company A and company B each have two branches in two different locations. The goal is for company A's branches (A-1 and A-2) to be able to communicate only with each other over VPN but not with company B's branches. Likewise, company B's branches (B-1 and B-2) can only communicate with each other over VPN but not with company A's. This is accomplished by placing each branch VLAN into their respective VRFs (VRF1 and VRF2), and encapsulating the VRF

information within the VPN tunnel. The hub forms BGP peering with its neighbors, spoke 1 and spoke 2, over each IPsec overlay. The hub's BGP route reflector reflects the routes to the corresponding VRFs, allowing each spoke to form ADVPN shortcuts with the other spoke for each VRF.

However, in this scenario, we want A-1 and A-2 to use an ADVPN shortcut, but we do not want B-1 and B-2 to use ADVPN. A route map is configured on the hub to match the desired extended community route target number where only this route target is permitted, and others are denied. This allows the hub's BGP route reflector to only reflect routes associated with VRF1, allowing the spokes to form an ADVPN shortcut for VRF1. Routes associated with VRF2 are not reflected, and each spoke must route traffic through the hub to reach VRF2 on the other spoke.



Configure the topology by following the instructions of [Example 1](#) in *SD-WAN segmentation over a single overlay*. Note that when checking the spoke 1 routes in example 1, there is a VRF2 route:

```
Spoke 1 # get router info routing-table bgp
...
Routing table for VRF=2
B V      33.1.1.0/24 [200/0] via 10.10.100.3 [2] (recursive via vd2-1 tunnel 11.1.1.11),
00:00:20, [1/0]
                [200/0] via 10.10.200.3 [2] (recursive via vd2-2 tunnel 11.1.2.11),
00:00:20, [1/0]
```

The following procedure applies a route map on the hub's BGP configurations to limit route reflection to only routes matching the external community target of 1:1. This external community target corresponds to BGP paths for VRF1 learned from spoke 1 and spoke 2. The external community target of 2:1 corresponds to BGP paths for VRF2. By not explicitly permitting this target (2:1) in the community list and denying everything other than the permitted target (1:1) in the route map, the VRF2 BGP paths are essentially omitted from being reflected to the spokes.

To configure BGP filtering for an extended community route target on the hub:

1. Identify the external community target of VRF1 to be permitted:

```
# get router info bgp network 33.1.1.0/24
VRF 0 BGP routing table entry for 33.1.1.0/24
```

```

Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    11.1.1.1
  Advertised to peer-groups:
    gr1 gr2
...
VRF 1 BGP routing table entry for 33.1.1.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0 external duplicated
  Local, (Received from a RR-client)
    0.0.0.0 from 10.10.100.3 (3.3.3.3)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:1
      Receive Path ID: 1
      Advertised Path ID: 1
      Last update: Wed Aug 17 10:31:02 2022
  Original VRF 0 external duplicated
  Local, (Received from a RR-client)
    0.0.0.0 from 10.10.200.3 (3.3.3.3)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:1
      Receive Path ID: 1
      Advertised Path ID: 2
      Last update: Wed Aug 17 10:31:02 2022
VRF 2 BGP routing table entry for 33.1.1.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0 external duplicated
  Local, (Received from a RR-client)
    0.0.0.0 from 10.10.100.3 (3.3.3.3)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Extended Community: RT:2:1
      Receive Path ID: 1
      Advertised Path ID: 1
      Last update: Wed Aug 17 10:31:02 2022
  Original VRF 0 external duplicated
  Local, (Received from a RR-client)
    0.0.0.0 from 10.10.200.3 (3.3.3.3)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Extended Community: RT:2:1
      Receive Path ID: 1
      Advertised Path ID: 2
      Last update: Wed Aug 17 10:31:02 2022

```

2. Configure the extended community list:

```

config router extcommunity-list
  edit "extcomm1"
    config rule
      edit 1
        set action permit
        set match "1:1"
        set type rt
      next
    end

```

```

    next
end

```

3. Apply the extended community list to the route map:

```

config router route-map
  edit "rmp11"
    config rule
      edit 1
        set match-extcommunity "extcomm1"
      next
      edit 2
        set action deny
      next
    end
  next
end

```

4. Update the related BGP neighbor group settings:

```

config router bgp
  config neighbor-group
    edit "gr1"
      set route-map-out-vpnv4 "rmp11"
    next
    edit "gr2"
      set route-map-out-vpnv4 "rmp11"
    next
  end
end

```

5. Refresh the routes:

```
# execute router clear bgp all vpnv4 unicast out
```

6. Check the spoke 1 routes. Since the extended community route target is applied, the VFR2 route does not appear in the BGP routing table:

```

# get router info routing-table bgp
Routing table for VRF=0
B*      0.0.0.0/0 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11),
03:47:50, [1/0]
                [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11),
03:47:50, [1/0]
B       1.1.1.1/32 [200/0] via 11.1.1.1 [2] (recursive via 12.1.1.1, vd2-vlan12),
03:47:50, [1/0]
B       1.222.222.222/32 [200/0] via 11.1.1.1 [2] (recursive via 12.1.1.1, vd2-vlan12),
03:47:50, [1/0]
B       11.11.11.11/32 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11),
03:47:50, [1/0]
                [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11),
03:47:50, [1/0]
B       33.1.1.0/24 [200/0] via 10.10.100.254 [2] (recursive via vd2-1 tunnel
11.1.1.11), 03:47:21, [1/0]
                [200/0] via 10.10.200.254 [2] (recursive via vd2-2 tunnel
11.1.2.11), 03:47:21, [1/0]

Routing table for VRF=1

```

```

B V      11.11.22.11/32 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11),
03:47:50, [1/0]
                        [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11),
03:47:50, [1/0]
B V      33.1.1.0/24 [200/0] via 10.10.100.3 [2] (recursive via vd2-1 tunnel 11.1.1.11),
03:47:21, [1/0]
                        [200/0] via 10.10.200.3 [2] (recursive via vd2-2 tunnel 11.1.2.11),
03:47:21, [1/0]
B V      100.1.1.0/24 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11),
03:47:50, [1/0]
                        [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11),
03:47:50, [1/0]

```

Add static route tag and BGP neighbor password - 7.2.4



This information is also available in the FortiOS 7.2 Administration Guide:

- [Static route tags](#)
- [BGP neighbor password](#)

The following routing extensions are added:

- Static route tags:

```

config router static
  edit <seq-num>
    set tag <id>
  next
end

```

- BGP neighbor passwords (used for the neighbor range):

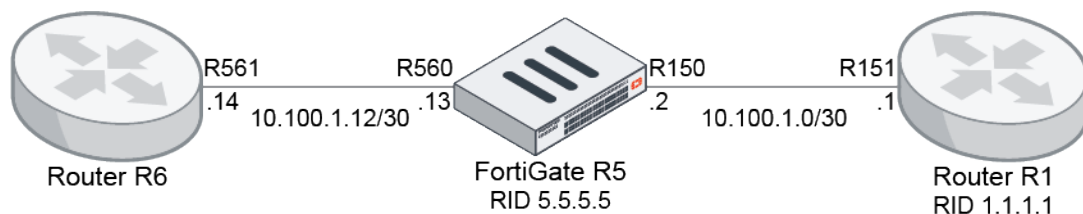
```

config router bgp
  config neighbor-group
    edit <name>
      set password <password>
    next
  end
end

```

Example 1

In this example, a static route is configured with a route tag. The route tag is then matched in the route map, and used to set the route's metric and advertise to the BGP neighbor.



To configure the FortiGate:**1. Configure the static route:**

```
config router static
  edit 1
    set dst 77.7.7.7 255.255.255.255
    set distance 2
    set device "R560"
    set tag 565
  next
end
```

2. Configure the route map:

```
config router route-map
  edit "map1"
    config rule
      edit 2
        set match-tag 565
        set set-metric 2301
      next
    end
  next
end
```

3. Configure the BGP neighbor:

```
config router bgp
  config neighbor
    edit "10.100.1.2"
      set route-map-out "map1"
    next
  end
end
```

On its neighbor side, router R1 receives the advertised route from the FortiGate router R5.

4. Verify the BGP routing table:

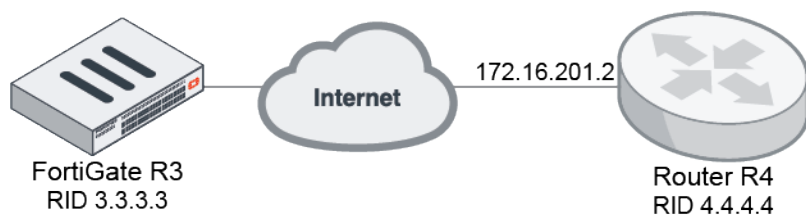
```
# get router info routing-table bgp
Routing table for VRF=0
B       77.7.7.7/32 [20/2301] via 10.100.1.1 (recursive is directly connected, R150),
03:18:53, [1/0]
```

5. Verify the network community:

```
# get router info bgp network 77.7.7.7/32
VRF 0 BGP routing table entry for 77.7.7.7/32
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    2.2.2.2 3.3.3.3 10.100.1.5 2000::2:2:2:2
  Original VRF 0
  20
    10.100.1.1 from 10.100.1.1 (5.5.5.5)
      Origin incomplete metric 2301, localpref 200, valid, external, best
      Last update: Wed Oct  5 16:48:28 2022
```

Example 2

In this example, a BGP group is configured, and it uses a password to establish the neighborhood.



To configure the BGP group:

1. Configure the R3 FortiGate settings:

```

config router bgp
  config neighbor-group
    edit "FGT"
      set soft-reconfiguration enable
      set remote-as 65050
      set local-as 65518
      set local-as-no-prepend enable
      set local-as-replace-as enable
      set route-map-in "del-comm"
      set keep-alive-timer 30
      set holdtime-timer 90
      set update-source "np0_vlink0"
      set weight 1000
      set password ENC *****
    next
  end
  config neighbor-range
    edit 1
      set prefix 172.16.201.0 255.255.255.0
      set max-neighbor-num 10
      set neighbor-group "FGT"
    next
  end
end

```

2. Configure the R4 router settings:

```

config router bgp
  config neighbor
    edit "172.16.201.1"
      set soft-reconfiguration enable
      set remote-as 65518
      set password *****
    next
  end
end

```

SD-WAN steering

7.2.0

- [Allow application category as an option for SD-WAN rule destination on page 112](#)
- [Add mean option score calculation and logging in performance SLA health checks on page 117](#)

7.2.1

- [Allow application category as a GUI option for SD-WAN rule destination 7.2.1 on page 119](#)
- [Embedded SD-WAN SLA information in ICMP probes 7.2.1 on page 121](#)

7.2.2

- [SD-WAN Template added the health-check embedded SLA information FMG 7.2.2 on page 129](#)

Allow application category as an option for SD-WAN rule destination



This feature is available only in the CLI with FortiOS 7.2.0. Starting in FortiOS 7.2.1, this feature is available in the CLI and the GUI. See also [Allow application category as a GUI option for SD-WAN rule destination 7.2.1 on page 119](#).

An application category can be selected as an SD-WAN service rule destination criterion. Previously, only application groups or individual applications could be selected.

```
config system sdwan
  config service
    edit <id>
      set internet-service enable
      set internet-service-app-ctrl-category <id_1> <id_2> ... <id_n>
    next
  end
end
```

To view the detected application categories details based on category ID, use `diagnose sys sdwan internet-service-app-ctrl-category-list <id>`.

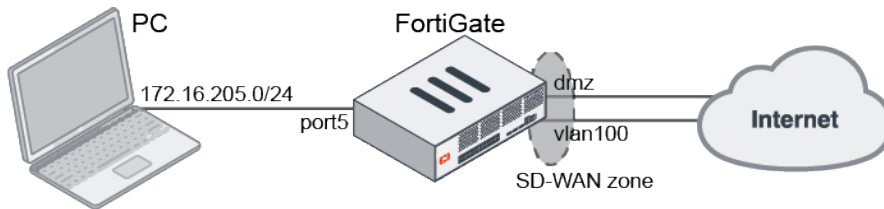


This information is also available in the FortiOS 7.2 Administration Guide:

- [Use an application category as an SD-WAN rule destination](#)
-

Example

In this example, traffic steering is applied to traffic detected as video/audio (category ID 5) or email (category ID 21) and applies the lowest cost (SLA) strategy to this traffic. When costs are tied, the priority goes to member 1, dmz.



To configure application categories as an SD-WAN rule destination in the CLI:

1. Configure the SD-WAN settings:

```

config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "dmz"
            set gateway 172.16.208.2
        next
        edit 2
            set interface "vlan100"
            set gateway 172.16.206.2
        next
    end
    config health-check
        edit "1"
            set server "8.8.8.8"
            set protocol dns
            set members 0
            config sla
                edit 1
                next
            end
        next
    end
end
end
end

```

2. Configure the SD-WAN rule to use application categories 5 and 21:

```

config system sdwan
    config service
        edit 1
            set name "1"
            set mode sla
            set src "172.16.205.0"
            set internet-service enable
            set internet-service-app-ctrl-category 5 21
        end
    end
end

```

```

        config sla
            edit "1"
                set id 1
            next
        end
        set priority-members 1 2
    next
end
end

```

3. Configure the firewall policy:

```

config firewall policy
    edit 1
        set srcintf "port5"
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr 172.16.205.0
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set application-list "g-default"
    next
end

```

4. Verify that the traffic is sent over dmz:

```

# diagnose firewall proute list
list route policy info(vf=root):
id=2133590017(0x7f2c0001) vwl_service=1(1) vwl_mbr_seq=1 2 dscp_tag=0xff 0xff flags=0x0
tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-65535 path(2) oif=5(dmz)
oif=95(vlan100)
source(1): 172.16.205.0-172.16.205.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(2): (null)(0,5,0,0,0) (null)(0,21,0,0,0)
hit_count=469 last_used=2021-12-15 15:06:05

```

5. View some videos and emails on the PC, then verify the detected application details for each category:

```

# diagnose sys sdwan internet-service-app-ctrl-category-list 5
YouTube(31077 4294838537): 142.250.217.110 6 443 Wed Dec 15 15:39:50 2021
YouTube(31077 4294838537): 173.194.152.89 6 443 Wed Dec 15 15:37:20 2021
YouTube(31077 4294838537): 173.194.152.170 6 443 Wed Dec 15 15:37:37 2021
YouTube(31077 4294838537): 209.52.146.205 6 443 Wed Dec 15 15:37:19 2021

# diagnose sys sdwan internet-service-app-ctrl-category-list 21
Gmail(15817 4294836957): 172.217.14.197 6 443 Wed Dec 15 15:39:47 2021

```

6. Verify that the captured email traffic is sent over dmz:

```

# diagnose sniffer packet any 'host 172.217.14.197' 4
interfaces=[any]
filters=[host 172.217.14.197]
5.079814 dmz out 172.16.205.100.60592 -> 172.217.14.197.443: psh 2961561240 ack
2277134591

```

7. Edit the SD-WAN rule so that dmz has a higher cost and vlan100 is preferred.

8. Verify that the traffic is now sent over vlan100:

```
# diagnose firewall proute list
list route policy info(vf=root):
id=2134048769(0x7f330001) vwl_service=1(1) vwl_mbr_seq=2 1 dscp_tag=0xff 0xff flags=0x0
tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-65535 path(2) oif=95
(vlan100) oif=5(dmz)
source(1): 172.16.205.0-172.16.205.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(2): (null)(0,5,0,0,0) (null)(0,21,0,0,0)
hit_count=635 last_used=2021-12-15 15:55:43

# diagnose sniffer packet any 'host 172.217.14.197' 4
interfaces=[any]
filters=[host 172.217.14.197]
304.625168 vlan100 in 172.16.205.100.60592 -> 172.217.14.197.443: psh 2961572711 ack
2277139565
```

To configure application categories as an SD-WAN rule destination in the GUI:



This functionality is available in FortiOS 7.2.1 and later. Prior to 7.2.1, individual applications can be selected in SD-WAN rules by default.

After upgrading to 7.2.1 or later, the GUI functionality is available if applications are already configured in SD-WAN rules prior to upgrading. Otherwise, by default, individual applications and application groups cannot be selected in SD-WAN rules. To enable this functionality, see step 1 in the following procedure.

1. Enable the feature visibility:

- a. Go to *System > Feature Visibility*.
- b. In the *Additional Features* section, enable *Application Detection Based SD-WAN*.
- c. Click *Apply*.



To enable GUI visibility of application detection based SD-WAN in the CLI:

```
config system global
    set gui-app-detection-sdwan enable
end
```

2. Configure the SD-WAN members:

- a. Go to *Network > SD-WAN*, select the *SD-WAN Zones* tab, and click *Create New > SD-WAN Member*.
- b. Set the *Interface* to *dmz*, and set the *Gateway* to *172.16.208.2*.
- c. Click *OK*.
- d. Repeat these steps to create another member for the *vlan100* interface with gateway *172.16.206.2*.

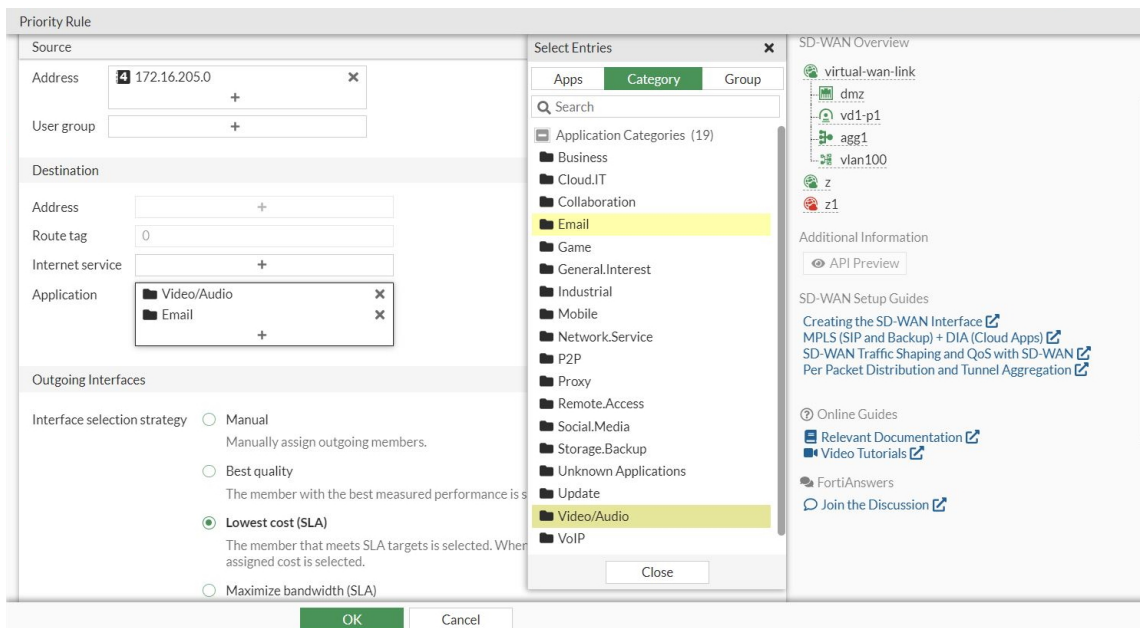
3. Configure the performance SLA (health check):

- a. Go to *Network > SD-WAN*, and select the *Performance SLAs* tab, and click *Create New*.
- b. Configure the following settings:

Name	1
Protocol	DNS

Server	8.8.8.8
SLA Target	Enable

- c. Click OK.
4. Configure the SD-WAN rule to use the video/audio and email application categories:
 - a. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
 - b. In the *Destination* section, click the + in the *Application* field.
 - c. Click *Category*, and select *Video/Audio* and *Email*.



- d. Configure the other settings as needed.
- e. Click OK.
5. Configure the firewall policy:
 - a. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
 - b. Configure the following settings:

Incoming Interface	<i>port5</i>
Outgoing Interface	<i>virtual-wan-link</i>
Source	<i>172.16.205.0</i>
Destination	<i>all</i>
Schedule	<i>always</i>
Service	<i>ALL</i>
Action	<i>ACCEPT</i>
Application Control	<i>g-default</i>
SSL Inspection	<i>certificate-inspection</i>

- c. Click OK.

Add mean opinion score calculation and logging in performance SLA health checks



This information is also available in the FortiOS 7.2 Administration Guide:

- [Mean opinion score calculation and logging in performance SLA health checks](#)

The mean opinion score (MOS) is a method of measuring voice quality using a formula that takes latency, jitter, packet loss, and the codec into account to produce a score from zero to five (0 - 5). The G.711, G.729, and G.722 codecs can be selected in the health check configurations, and an MOS threshold can be entered to indicate the minimum MOS score for the SLA to pass. The maximum MOS score will depend on which codec is used, since each codec has a theoretical maximum limit.

```
config system sdwan
  config health-check
    edit <name>
      set mos-codec {g711 | g729 | g722}
      config sla
        edit <id>
          set link-cost-factor {latency jitter packet-loss mos}
          set mos-threshold <value>
        next
      end
    next
  end
end
```

mos-codec {g711 g729 g722}	Set the VoIP codec to use for the MOS calculation (default = g711).
link-cost-factor {latency jitter packet-loss mos}	Set the criteria to base the link selection on.
mos-threshold <value>	Set the minimum MOS for the SLA to be marked as pass (1.0 - 5.0, default = 3.6).



Currently, the MOS cannot be used as the `link-cost-factor` to steer traffic in an SD-WAN rule.

To configure a health check to calculate the MOS:

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

```

        set gateway 172.16.208.2
    next
    edit 2
        set interface "port15"
        set gateway 172.16.209.2
    next
end
config health-check
    edit "Test_MOS"
        set server "2.2.2.2"
        set sla-fail-log-period 30
        set sla-pass-log-period 30
        set members 0
        set mos-codec g729
        config sla
            edit 1
                set link-cost-factor mos
                set mos-threshold "4.0"
            next
        end
    next
end
end

```

To verify the MOS calculation results:

1. Verify the health check diagnostics:

```

# diagnose sys sdwan health-check
Health Check(Test_MOS):
Seq(1 dmz): state(alive), packet-loss(0.000%) latency(0.114), jitter(0.026), mos(4.123),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
Seq(2 port15): state(alive), packet-loss(0.000%) latency(0.100), jitter(0.008), mos
(4.123), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1

# diagnose sys sdwan sla-log Test_MOS 1
Timestamp: Tue Jan  4 11:23:06 2022, vdom root, health-check Test_MOS, interface: dmz,
status: up, latency: 0.151, jitter: 0.040, packet loss: 0.000%, mos: 4.123.
Timestamp: Tue Jan  4 11:23:07 2022, vdom root, health-check Test_MOS, interface: dmz,
status: up, latency: 0.149, jitter: 0.041, packet loss: 0.000%, mos: 4.123.

# diagnose sys sdwan sla-log Test_MOS 2
Timestamp: Tue Jan  4 11:25:09 2022, vdom root, health-check Test_MOS, interface:
port15, status: up, latency: 0.097, jitter: 0.009, packet loss: 0.000%, mos: 4.123.
Timestamp: Tue Jan  4 11:25:10 2022, vdom root, health-check Test_MOS, interface:
port15, status: up, latency: 0.097, jitter: 0.008, packet loss: 0.000%, mos: 4.123.

```

2. Change the mos-codec to g722. The diagnostics will now display different MOS values:

```

# diagnose sys sdwan health-check
Health Check(Test_MOS):
Seq(1 dmz): state(alive), packet-loss(0.000%) latency(0.150), jitter(0.031), mos(4.453),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
Seq(2 port15): state(alive), packet-loss(0.000%) latency(0.104), jitter(0.008), mos
(4.453), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1

```

3. Increase the latency on the link in port15. The calculated MOS value will decrease accordingly. In this example, port15 is out of SLA since its MOS value is now less than the 4.0 minimum:

```
# diagnose sys sdwan health-check
Health Check(Test_MOS):
Seq(1 dmz): state(alive), packet-loss(0.000%) latency(0.106), jitter(0.022), mos(4.453),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
Seq(2 port15): state(alive), packet-loss(0.000%) latency(300.119), jitter(0.012), mos
(3.905), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x0
```

Sample logs

```
date=2022-01-04 time=11:57:54 eventtime=1641326274876828300 tz="-0800" logid="0113022933"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN SLA notification"
eventtype="SLA" healthcheck="Test_MOS" slatargetid=1 interface="port15" status="up"
latency="300.118" jitter="0.013" packetloss="0.000" mos="3.905"
inbandwidthavailable="1000.00Mbps" outbandwidthavailable="1000.00Mbps"
bibandwidthavailable="2.00Gbps" inbandwidthused="0kbps" outbandwidthused="0kbps"
bibandwidthused="0kbps" slamap="0x0" metric="mos" msg="Health Check SLA status. SLA failed
due to being over the performance metric threshold."
```

```
date=2022-01-04 time=11:57:24 eventtime=1641326244286635920 tz="-0800" logid="0113022923"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status"
eventtype="Health Check" healthcheck="Test_MOS" slatargetid=1 oldvalue="2" newvalue="1"
msg="Number of pass member changed."
```

```
date=2022-01-04 time=11:57:24 eventtime=1641326244286627260 tz="-0800" logid="0113022923"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status"
eventtype="Health Check" healthcheck="Test_MOS" slatargetid=1 member="2" msg="Member status
changed. Member out-of-sla."
```

```
date=2022-01-04 time=11:57:02 eventtime=1641326222516756500 tz="-0800" logid="0113022925"
type="event" subtype="sdwan" level="information" vd="root" logdesc="SDWAN SLA information"
eventtype="SLA" healthcheck="Test_MOS" slatargetid=1 interface="port15" status="up"
latency="0.106" jitter="0.007" packetloss="0.000" mos="4.453"
inbandwidthavailable="1000.00Mbps" outbandwidthavailable="1000.00Mbps"
bibandwidthavailable="2.00Gbps" inbandwidthused="0kbps" outbandwidthused="0kbps"
bibandwidthused="0kbps" slamap="0x1" msg="Health Check SLA status."
```

Allow application category as a GUI option for SD-WAN rule destination - 7.2.1



This feature is available only in the CLI with FortiOS 7.2.0. Starting in FortiOS 7.2.1, this feature is available in the CLI and the GUI. See also [Allow application category as an option for SD-WAN rule destination on page 112](#).

An application category can be selected as an SD-WAN service rule destination criterion. Previously, only application groups or individual applications could be selected.



This information is also available in the FortiOS 7.2 Administration Guide:

- [Use an application category as an SD-WAN rule destination](#)

To configure application categories as an SD-WAN rule destination in the GUI:

1. Enable the feature visibility:
 - a. Go to *System > Feature Visibility*.
 - b. In the *Additional Features* section, enable *Application Detection Based SD-WAN*.
 - c. Click *Apply*.
2. Configure the SD-WAN members:
 - a. Go to *Network > SD-WAN*, select the *SD-WAN Zones* tab, and click *Create New > SD-WAN Member*.
 - b. Set the *Interface* to *dmz*, and set the *Gateway* to *172.16.208.2*.
 - c. Click *OK*.
 - d. Repeat these steps to create another member for the *vlan100* interface with gateway *172.16.206.2*.
3. Configure the performance SLA (health check):
 - a. Go to *Network > SD-WAN*, and select the *Performance SLAs* tab, and click *Create New*.
 - b. Configure the following settings:

Name	1
Protocol	DNS
Server	8.8.8.8
SLA Target	Enable

- c. Click *OK*.
4. Configure the SD-WAN rule to use the video/audio and email application categories:
 - a. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
 - b. In the *Destination* section, click the + in the *Application* field.
 - c. Click *Category*, and select *Video/Audio* and *Email*.

The screenshot displays the FortiManager GUI for configuring an SD-WAN rule. The main configuration area is titled 'Priority Rule'. Under the 'Source' section, the 'Address' field is populated with '172.16.205.0'. The 'Destination' section shows the 'Application' field with a dropdown menu open, displaying 'Video/Audio' and 'Email' as selected options. The 'Outgoing Interfaces' section has the 'Interface selection strategy' set to 'Lowest cost (SLA)'. A 'Select Entries' dialog box is overlaid on the configuration, showing a list of application categories. The 'Email' and 'Video/Audio' categories are highlighted. The 'SD-WAN Overview' panel on the right provides a summary of the SD-WAN setup, including the virtual-wan-link and its associated interfaces: dmz, vd1-p1, agg1, and vlan100.

- d. Configure the other settings as needed.
- e. Click *OK*.

5. Configure the firewall policy:
 - a. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
 - b. Configure the following settings:

Incoming Interface	<i>port5</i>
Outgoing Interface	<i>virtual-wan-link</i>
Source	<i>172.16.205.0</i>
Destination	<i>all</i>
Schedule	<i>always</i>
Service	<i>ALL</i>
Action	<i>ACCEPT</i>
Application Control	<i>g-default</i>
SSL Inspection	<i>certificate-inspection</i>

- c. Click *OK*.

Embedded SD-WAN SLA information in ICMP probes - 7.2.1



This information is also available in the FortiOS 7.2 Administration Guide:

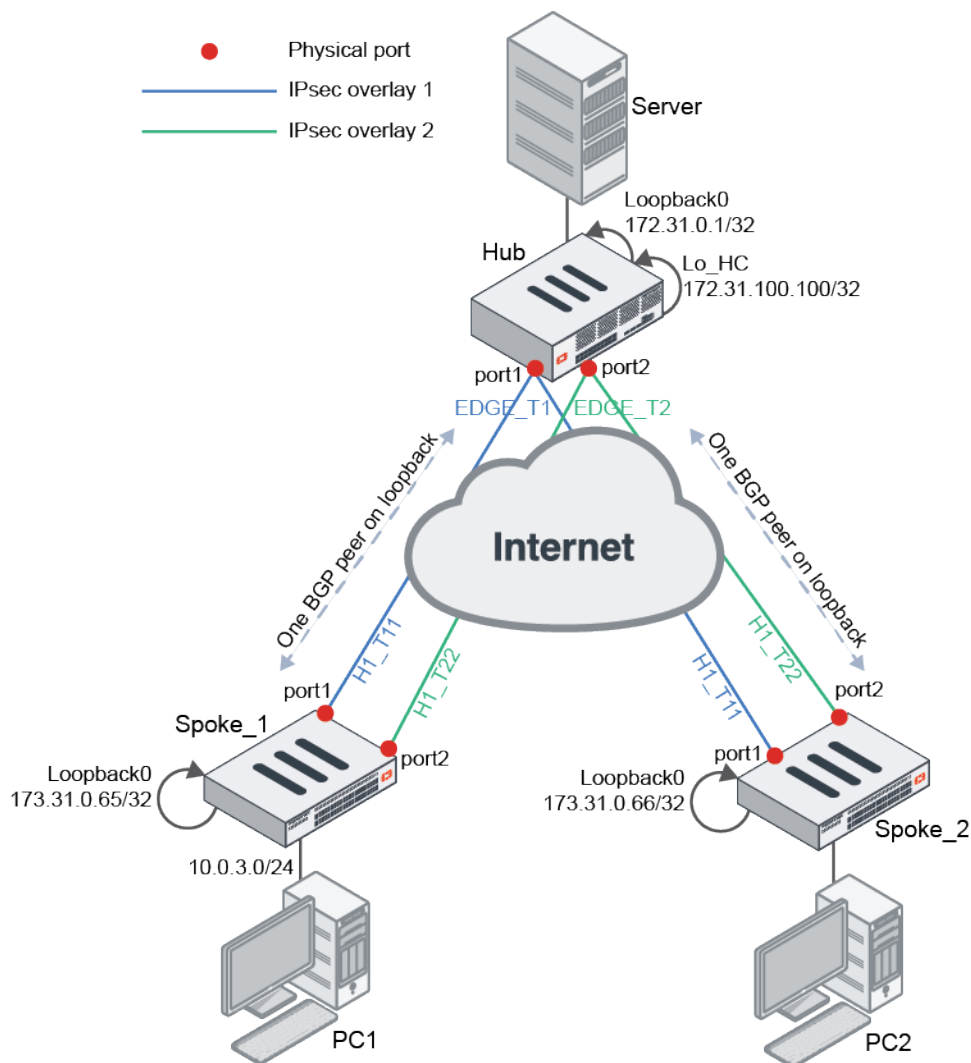
- [Embedded SD-WAN SLA information in ICMP probes](#)

In the hub and spoke SD-WAN design, in order for traffic to pass symmetrically from spoke to hub and hub to spoke, it is essential for the hub to know which IPsec overlay is in SLA and out of SLA. Prior to introducing embedded SLA information in ICMP probes, it is common practice for spokes to use the SD-WAN neighbor feature and `route-map-out-preferable` setting to signal the health of each overlay to the hub. However, this requires BGP to be configured per overlay, and to manipulate BGP routes using custom BGP communities.

With embedded SLA information in ICMP probes, spokes can communicate their SLA for each overlay directly through ICMP probes to the hub. The hub learns these SLAs and maps the status for each spoke and its corresponding overlays.

The hub uses the SLA status to apply priorities to the IKE routes, giving routes over IPsec overlays that are within SLAs a lower priority value and routes over overlays out of SLAs a higher priority value. If BGP is used, recursively resolved BGP routes can inherit the priority from its parent.

Embedded SLA information in ICMP probes allows hub and spoke SD-WAN to be designed with a BGP on loopback topology, or without BGP at all. The following topology outlines an example of the BGP on loopback design where each spoke is peered with the hub and route reflector on the loopback interface.



In this topology, each FortiGate's BGP router ID is based on its Loopback0 interface. Each spoke has SLA health checks defined to send ICMP probes to the server's Lo_HC interface on 172.31.100.100. The ICMP probes include embedded SLA information for each SD-WAN overlay member.

Related SD-WAN settings:

```
config system sdwan
  config health-check
    edit <name>
      set detect-mode {active | passive | prefer-passive | remote}
      set embed-measured-health {enable | disable}
    config sla
      edit <id>
        set priority-in-sla <integer>
        set priority-out-sla <integer>
      next
    end
    set sla-id-redistribute <id>
  next
```

```
end
end
```

<pre>detect-mode {active passive prefer- passive remote}</pre>	<p>Set the mode that determines how to detect the server:</p> <ul style="list-style-type: none"> • active: the probes are sent actively (default). • passive: the traffic measures health without probes. • prefer-passive: the probes are sent in case of no new traffic. • remote: the link health is obtained from remote peers.
<pre>embed-measured-health {enable disable}</pre>	<p>Enable/disable embedding SLA information in ICMP probes (default = disable).</p>
<pre>set priority-in-sla <integer></pre>	<p>Set the priority that will be set to the IKE route when the corresponding overlay is in SLA (0 - 65535).</p>
<pre>set priority-out-sla <integer></pre>	<p>Set the priority that will be set to the IKE route when the corresponding overlay is out of SLA (0 - 65535).</p>
<pre>sla-id-redistribute <id></pre>	<p>Set the SLA entry (ID) that will be applied to the IKE routes (0 - 32, default = 0).</p>

Related BGP setting:

```
config router bgp
  set recursive-inherit-priority {enable | disable}
end
```

<pre>recursive-inherit- priority {enable disable}</pre>	<p>Enable/disable allowing recursive resolved BGP routes to inherit priority from its parent (default = disable).</p>
---	---

Example with BGP on loopback SD-WAN

This example demonstrates the configurations needed to configure the SD-WAN and BGP settings for the preceding topology. It is assumed that IPsec VPN overlays are already configured per the topology, and that loopback interfaces are already configured on each FortiGate.

Configuring the Spoke_1 FortiGate

In the SD-WAN settings, note the following requirements:

1. Configure the SD-WAN zones and members. For each SD-WAN member, define the source of its probes to be the Loopback0 interface IP.
2. Configure the SLA health check to point to the Hub's Lo_HC interface and IP. Enable `embed-measured-health`.
3. Configure an SD-WAN service rule to route traffic based on the maximize bandwidth (SLA) algorithm to prefer member H1_T11 over H1_T22.
4. Configure `set exchange-interface-ip enable` and `set exchange-ip-addr4` to the Loopback0 interface IP. The `exchange-interface-ip` option is automatically turned on when ADVPN has already been configured. If ADVPN has not been configured, then `set exchange-interface-ip enable` must be configured before `set exchange-ip-addr4` can be configured.

To configure the SD-WAN settings:

```
config system sdwan
  set status enable
  config zone
    edit "virtual-wan-link"
    next
    edit "overlay"
    next
  end
  config members
    edit 1
      set interface "H1_T11"
      set zone "overlay"
      set source 172.31.0.65
    next
    edit 4
      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 embed-measured-health enable
      set members 0
      config sla
        edit 1
          set link-cost-factor latency
          set latency-threshold 100
        next
      end
    next
  end
  config service
    edit 1
      set mode sla
      set dst "CORP_LAN"
      set src "CORP_LAN"
      config sla
        edit "HUB"
          set id 1
        next
      end
      set priority-members 1 4
    next
  end
end
```

To configure the BGP settings:

```
config router bgp
  set as 65001
  set router-id 172.31.0.65
  config neighbor
```

```

        edit "172.31.0.1"
            set remote-as 65001
            set update-source "Loopback0"
        next
    end
    config network
        edit 1
            set prefix 10.0.3.0 255.255.255.0
        next
    end
end

```

To add the loopback IP to the IPsec interface settings:

```

config vpn ipsec phase1-interface
    edit "H1_T11"
        set exchange-interface-ip enable
        set exchange-ip-addr4 172.31.0.65
    next
    edit "H1_T22"
        set exchange-interface-ip enable
        set exchange-ip-addr4 172.31.0.65
    next
end

```

Configuring the hub FortiGate

In the SD-WAN settings, note the following requirements:

1. Configure the SD-WAN zone and members.
2. Configure the SLA health checks to detect SLAs based on the remote site (spoke). This must be defined for each SD-WAN member:
 - a. For the SLA, specify the same link cost factor and metric as the spoke (100).
 - b. Define the IKE route priority for in and out of SLA. Lower priority values have higher priority than higher priority values.
3. Define the SLA entry ID that will be applied to the IKE routes.
4. Configure `set exchange-interface-ip enable` and `set exchange-ip-addr4` to the Loopback0 interface IP. The `exchange-interface-ip` option is automatically turned on when ADVPN has already been configured. If ADVPN has not been configured, then `set exchange-interface-ip enable` must be configured before `set exchange-ip-addr4` can be configured.

To configure the SD-WAN settings:

```

config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "EDGE_T1"
        next
    end
end

```

```

        edit 2
            set interface "EDGE_T2"
        next
    end
    config health-check
        edit "1"
            set detect-mode remote
            set sla-id-redistribute 1
            set members 1
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                    set priority-in-sla 10
                    set priority-out-sla 20
                next
            end
        next
    next
    edit "2"
        set detect-mode remote
        set sla-id-redistribute 1
        set members 2
        config sla
            edit 1
                set link-cost-factor latency
                set latency-threshold 100
                set priority-in-sla 15
                set priority-out-sla 25
            next
        end
    next
end
end

```

In the BGP settings, note the following requirements:

1. Enable `recursive-inherit-priority` to inherit the route priority from its parent, which is the priority defined in the health check SLA settings.
2. Configure the other BGP settings similar to a regular BGP hub.

To configure the BGP settings:

```

config router bgp
    set as 65001
    set router-id 172.31.0.1
    set recursive-inherit-priority enable
    config neighbor-group
        edit "EDGE"
            set remote-as 65001
            set update-source "Loopback0"
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 172.31.0.64 255.255.255.192
            set neighbor-group "EDGE"
        next
    end
end

```

```

        next
    end
end

```

To add the loopback IP to the IPsec interface settings:

```

config vpn ipsec phase1-interface
    edit "EDGE_T1"
        set exchange-interface-ip enable
        set exchange-ip-addr4 172.31.0.1
    next
    edit "EDGE_T2"
        set exchange-interface-ip enable
        set exchange-ip-addr4 172.31.0.1
    next
end

```

Testing and verification

Once the hub and spokes are configured, verify that SLA statuses are passed from the spoke to the hub.

To verify that the SLA statuses are passed from the spoke to the hub:

1. On Spoke_1, display the status of the health-checks for H1_T11 and H1_T22:

```

# diagnose sys sdwan health-check
Health Check(HUB):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.228), jitter(0.018), mos
(4.404), bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x1
Seq(4 H1_T22): state(alive), packet-loss(0.000%) latency(0.205), jitter(0.007), mos
(4.404), bandwidth-up(999998), bandwidth-dw(1000000), bandwidth-bi(1999998) sla_map=0x1

```

2. On Spoke_1, display the status and order of the overlays in the SD-WAN service rule:

```

# diagnose sys sdwan service
Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Tie break: cfg
Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
Members(2):
    1: Seq_num(1 H1_T11), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    2: Seq_num(4 H1_T22), alive, sla(0x1), gid(0), cfg_order(3), local cost(0), selected

Src address(1):
    10.0.0.0-10.255.255.255
Dst address(1):
    10.0.0.0-10.255.255.255

```

Both overlays are within SLA, so H1_T11 is preferred due to its `cfg-order`.

Spoke_1's SLA information for H1_T11 and H1_T22 is embedded into the ICMP probes destined for the hub's Lo_HC interface. The hub receives this information and maps the SLAs correspondingly per spoke and overlay based on the same SLA targets.

As a result, since all SLAs are within target, the hub sets the routes over each overlay as follows:

Hub SD-WAN member	Overlay	SLA status	Priority for IKE routes
1	EDGE_T1	0x1 – within SLA	10
2	EDGE_T2	0x1 – within SLA	15

Simultaneously, BGP recursive routes inherit the priority based on the parent IKE routes. The recursively resolved BGP routes that pass through EDGE_T1 will have a priority of 10, and routes that pass through EDGE_T2 will have a priority of 15. Therefore, traffic from the hub to spoke will be routed to EDGE_T1.

3. Verify the routing tables.

a. Static:

```
# get router info routing-table static
Routing table for VRF=0
S      172.31.0.65/32 [15/0] via EDGE_T1 tunnel 10.0.0.69 vrf 0, [10/0]
                                           [15/0] via EDGE_T2 tunnel 172.31.0.65 vrf 0, [15/0]
```

b. BGP:

```
# get router info routing-table bgp
Routing table for VRF=0
B      10.0.3.0/24 [200/0] via 172.31.0.65 (recursive via EDGE_T1 tunnel 10.0.0.69
vrf 0 [10]), 04:32:53
                                           (recursive via EDGE_T2 tunnel 172.31.0.65
vrf 0 [15]), 04:32:53, [1/0]
```

Next, test by making the health checks over the spokes' H1_T11 tunnel out of SLA. This should trigger traffic to start flowing from the spokes' H1_T22 tunnel. Consequently, the SLA statuses are passed from the spoke to the hub, and the hub will start routing traffic to EDGE_T2.

To verify that the hub will start routing traffic to EDGE_T2 when the spoke H1_T11 tunnel is out of SLA:

1. On Spoke_1, display the status of the health checks for H1_T11 and H1_T22:

```
# diagnose sys sdwan health-check
Health Check(HUB):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(120.228), jitter(0.013), mos
(4.338), bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x0
Seq(4 H1_T22): state(alive), packet-loss(0.000%) latency(0.220), jitter(0.008), mos
(4.404), bandwidth-up(999998), bandwidth-dw(1000000), bandwidth-bi(1999998) sla_map=0x1
```

2. Verify the routing tables.

a. Static:

```
# get router info routing-table static
Routing table for VRF=0
S      172.31.0.65/32 [15/0] via EDGE_T2 tunnel 172.31.0.65 vrf 0, [15/0]
                                           [15/0] via EDGE_T1 tunnel 10.0.0.69 vrf 0, [20/0]
```

The priority for EDGE_T1 has changed from 10 to 20.

b. BGP:

```
# get router info routing-table bgp
Routing table for VRF=0
B      10.0.3.0/24 [200/0] via 172.31.0.65 (recursive via EDGE_T2 tunnel 172.31.0.65
vrf 0 [15]), 00:01:19
```

```
(recursive via EDGE_T1 tunnel 10.0.0.69
vrf 0 [20]), 00:01:19, [1/0]
```

EDGE_T2 is now preferred. The priority for EDGE_T1 has changed from 10 to 20.

Spoke_1's SLA information for H1_T11 embedded into the ICMP probes has now changed.

As a result, the hub sets the routes over each overlay as follows:

Hub SD-WAN member	Overlay	SLA status	Priority for IKE routes
1	EDGE_T1	0x0 – out of SLA	20
2	EDGE_T2	0x1 – within SLA	15

The BGP recursive routes inherit the priority based on the parent IKE routes. Since priority for IKE routes on EDGE_T1 has changed to 20, recursively resolved BGP routes passing through EDGE_T1 has also dropped to 20. As a result, hub to spoke_1 traffic will go over EDGE_T2.

SD-WAN Template added the health-check embedded SLA information - FMG 7.2.2



This information is also available in the FortiManager 7.2 Administration Guide:

- [Performance SLA](#)

SD-WAN Template added the health-check embedded SLA information used to avoid asymmetric routing on the return traffic.

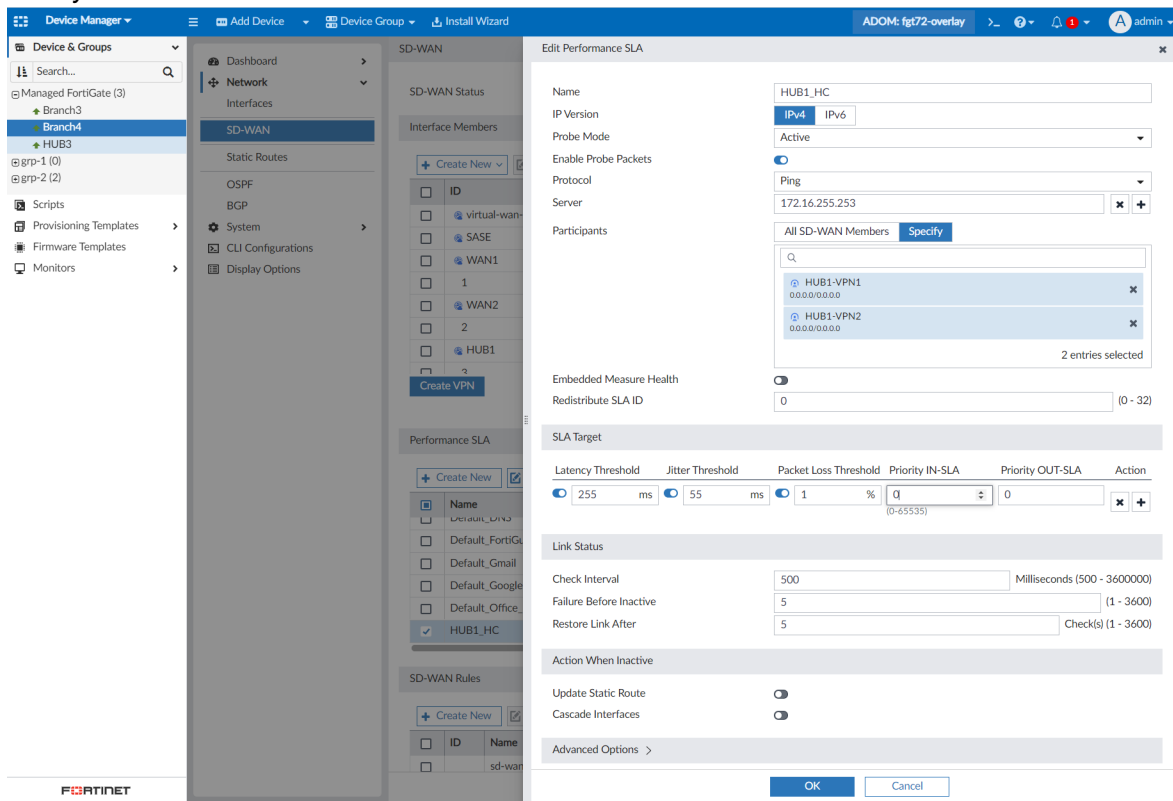
To view the new settings in FortiManager:

1. Enter a FortiManager 7.2 ADOM.
2. Go to *Device Manager > Managed Devices*, and select a managed device.
3. In the device database, go to *Network > SD-WAN*, and configure a *Performance SLA*.

The following settings have been added:

- Detection Mode: *Remote*
- *Embedded Health Measure*
- *Redistribute SLA ID*

- **Priority IN-SLA/OUT-SLA**



4. Go to *Device Manager > Provisioning Templates > SD-WAN Templates*.

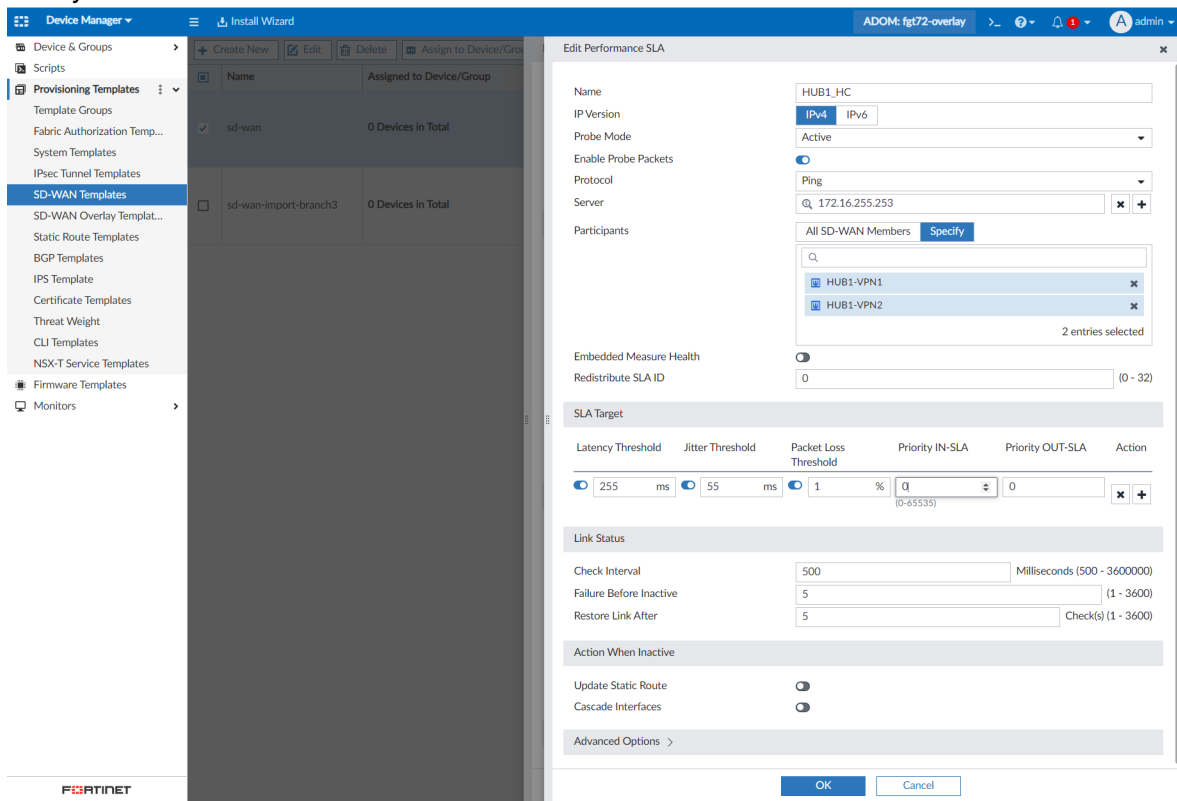
5. Edit or create an SD-WAN template.

6. Edit a *Performance SLA*.

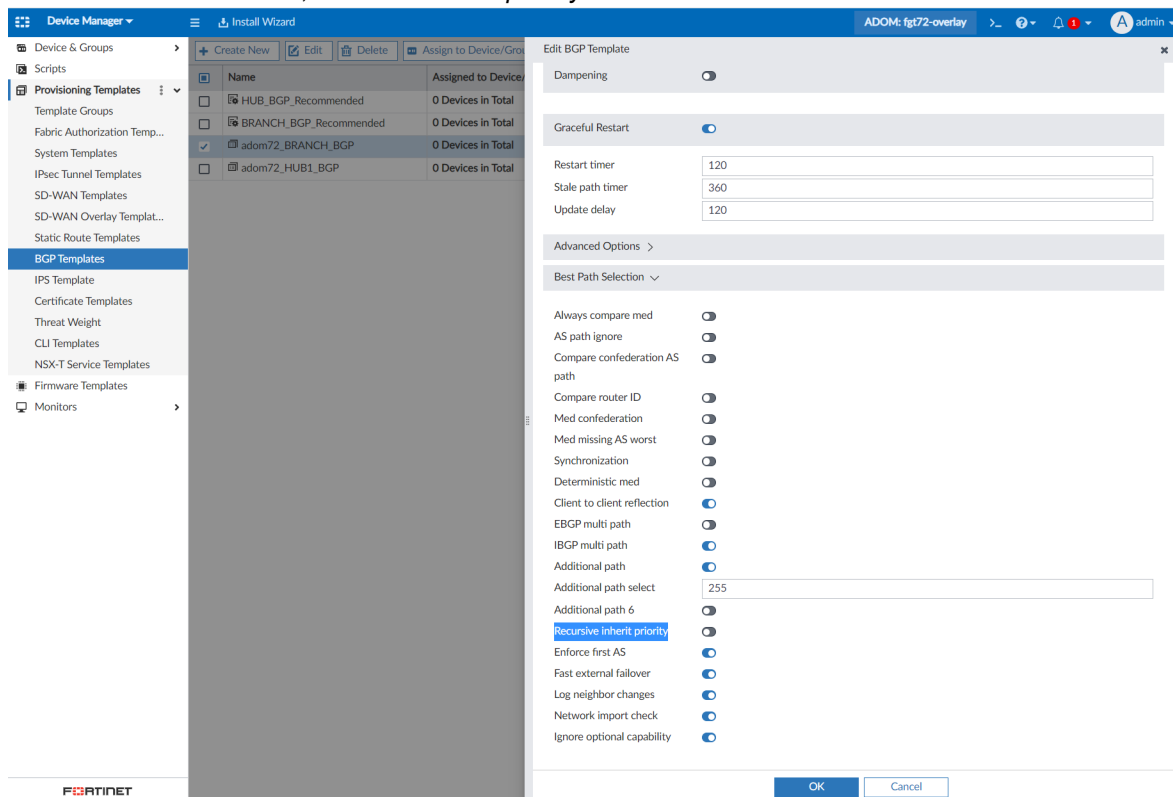
The following settings have been added:

- Detection Mode: *Remote*
- *Embedded Health Measure*
- *Redistribute SLA ID*

- **Priority IN-SLA/OUT-SLA**



7. Go to **Device Manager > Provisioning Templates > BGP Templates**. Under **Best Path Selection**, **Recursive inherit priority** has been added.



Visibility

7.2.1

- [Traffic shaping charts FAZ 7.2.1 on page 132](#)
- [High bandwidth application usage report update FAZ 7.2.1 on page 135](#)

Traffic shaping charts - FAZ 7.2.1



This information is also available in the FortiAnalyzer 7.2 Administration Guide:

- [FortiView Monitors dashboards](#)

The *Traffic Shaping Monitor* dashboard is added to *FortiView > Monitors*.

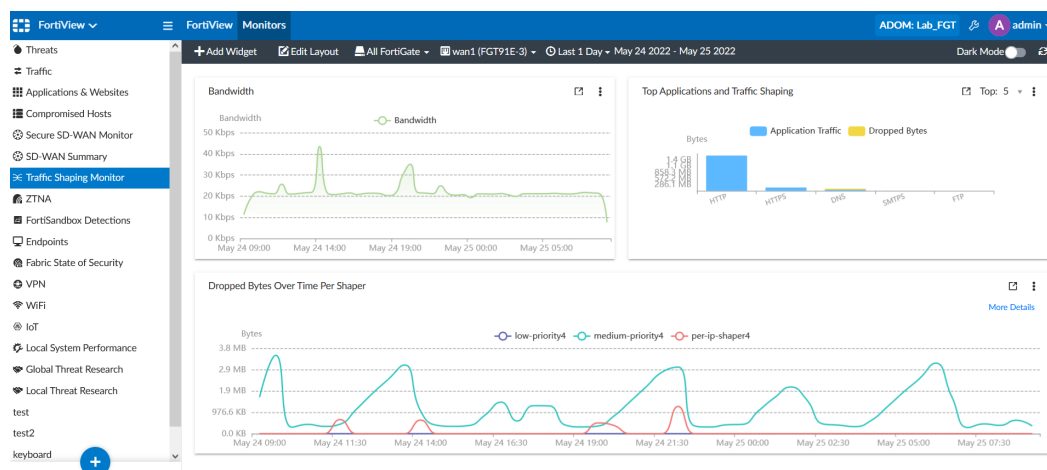
This dashboard includes the following widgets:

- **Bandwidth:** Displays the bandwidth of traffic shapers over time in a line chart.
- **Top Applications and Traffic Shaping:** Displays the traffic volume and dropped bytes for the top applications in a stacked bar chart.
- **Dropped Bytes Over Time Per Shaper:** Displays the dropped bytes for different traffic shapers on a selected interface over a period of time in a line chart.

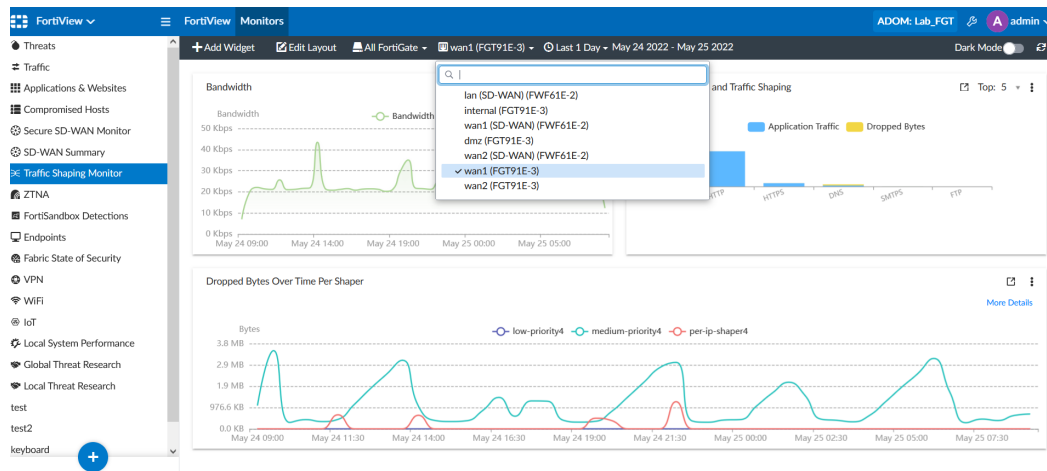


To use the *Traffic Shaping Monitor*, you must configure per-IP traffic shaper and shared traffic shaper on the FortiGate device. Then, configure a traffic shaping policy. For more information, see the [FortiGate Administration Guide](#).

To use these widgets, go to *FortiView > Monitors > Traffic Shaping Monitor*.

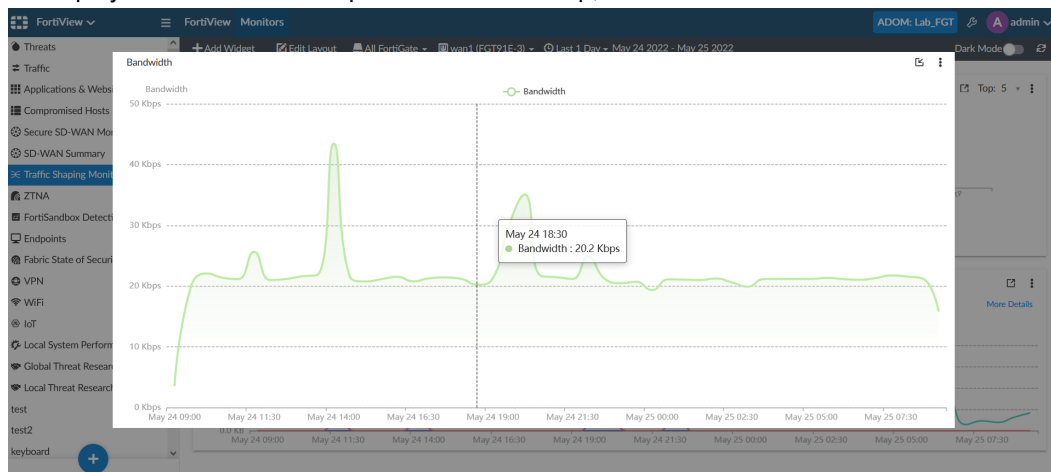


From the toolbar, select the devices, the traffic shaping interface, and a time range for the monitor. In the image below, the user selects *wan1 (FGT91E-3)* as the traffic shaping interface.



To use the *Bandwidth* widget:

1. To display the bandwidth at a specific time in a tooltip, mouse over the line chart.

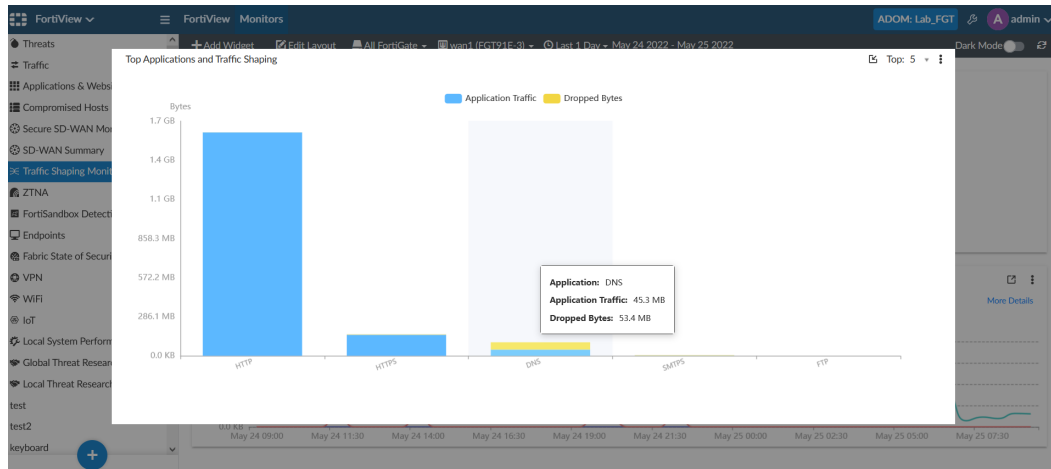


You can also click the bandwidth icon in the legend to hide/show the corresponding line in the chart.

To use the *Top Applications and Traffic Shaping* widget:

1. From the *Top* dropdown, select the number of top applications (5/10/15/20) to display in the widget . The widget displays the top five applications by default.

- To display a summary of application traffic and dropped bytes in a tooltip, mouse over the stacked bar chart.



Application Traffic and *Dropped Bytes* can be hidden/shown in the bar chart by click the corresponding icon in the legend.

- To display the *Application Users* table view, click a bar in the chart. This view includes a summary of the application traffic, including the number of sessions and bytes (sent/received) by user.

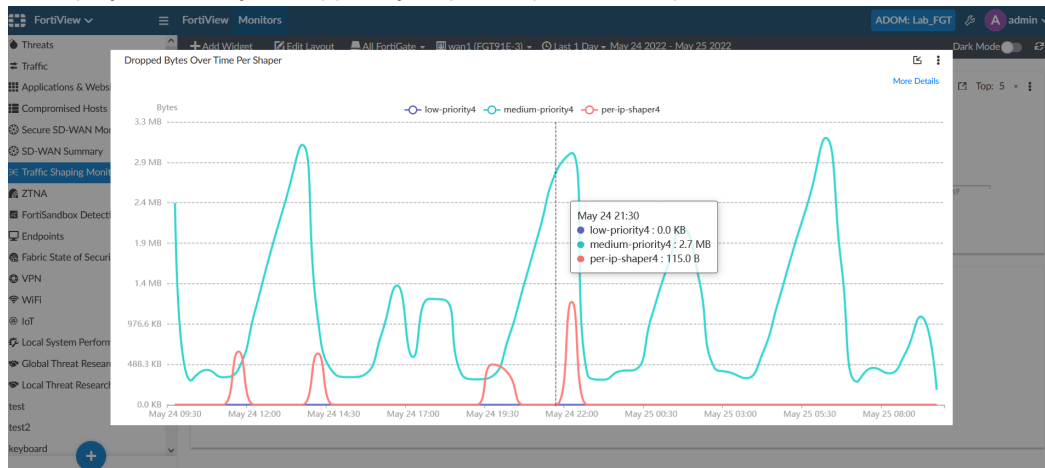
The screenshot shows the FortiView Monitors interface with the 'Application Users' table view selected. The table displays user information, sessions, and bytes sent/received for the DNS application. A summary box for the DNS application is also visible.

#	User Name	IP	Sessions	Bytes (Sent/Received)
1	10.0.0.100	10.0.0.100	4,078	190.8 KB/342.8 KB
2	10.0.0.100	10.0.0.100	2,783	206.2 KB/289.3 KB
3	10.0.0.100	10.0.0.100	1,993	131.7 KB/186.0 KB
4	10.0.0.100	10.0.0.100	905	128.2 KB/148.5 KB
5	10.0.0.100	10.0.0.100	368	924.6 KB/2.3 MB

- To return to the widget, click *Top Applications and Traffic Shaping*.

To use the *Dropped Bytes Over Time Per Shaper* widget:

1. To display a summary of dropped bytes per shaper in a tooltip, mouse over the line chart.



2. Click a shaper in the legend to hide/show it in the line chart. Greyed-out shapers in the legend are hidden in the line chart.
3. Click *More details* to display the *Traffic Shaping Policy Hits* table view. This table includes the total sessions and bytes (sent/received) by shaping policy.

#	Shaping Policy	Source Interface	Shared Shaper	Reverse Shaper	Per IP Shaper	Service	Applications	Sessions	Bytes (Sent/Received)
1	1	wan2			per-ip-shaper4	HTTP/HTTPS	HTTP HTTPS	102,721	199.5 MB/1.5 GB
2	3	wan2	medium-priority4	medium-priority4		DNS.SMTPS	DNS SMTPS	10,627	20.3 MB/26.6 MB
3	2	wan2	low-priority4	low-priority4		FTP	FTP	1	1.3 KB/1.4 KB

4. To return to the chart, click *Dropped Bytes Over Time Per Shaper*. Note that shared shapers, reverse shapers, and per-ip shapers are supported in this widget.

High bandwidth application usage report update - FAZ 7.2.1

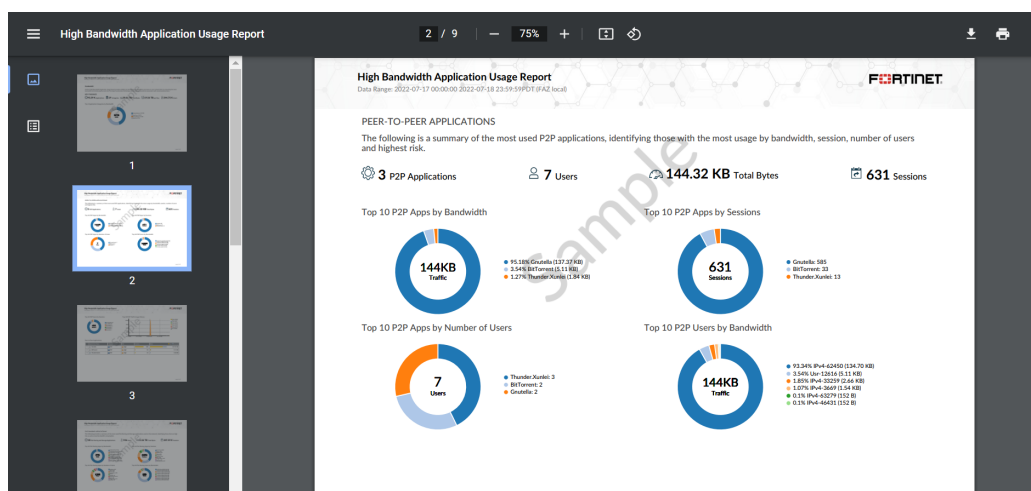
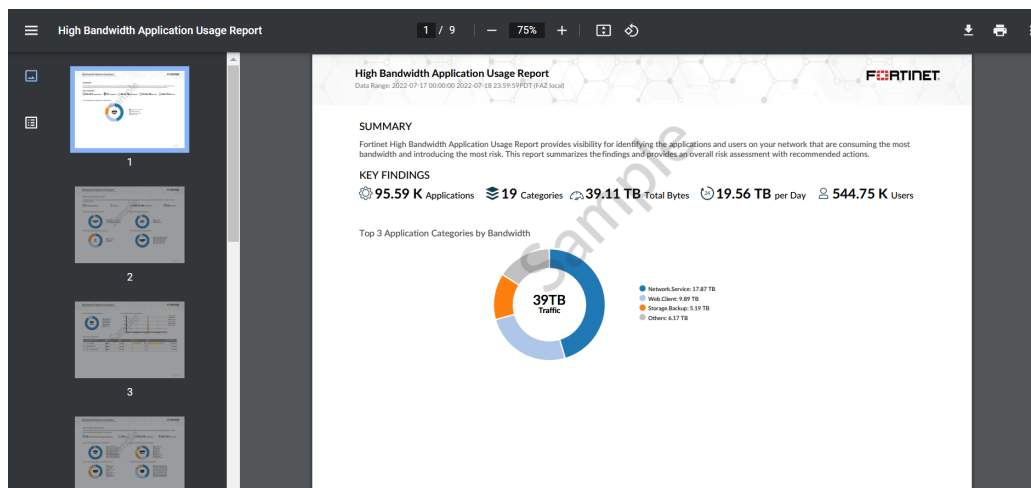


This information is also available in the FortiAnalyzer 7.2 Administration Guide:

- [Report template library](#)

The *High Bandwidth Application Usage Report* is updated to improve data visualization.

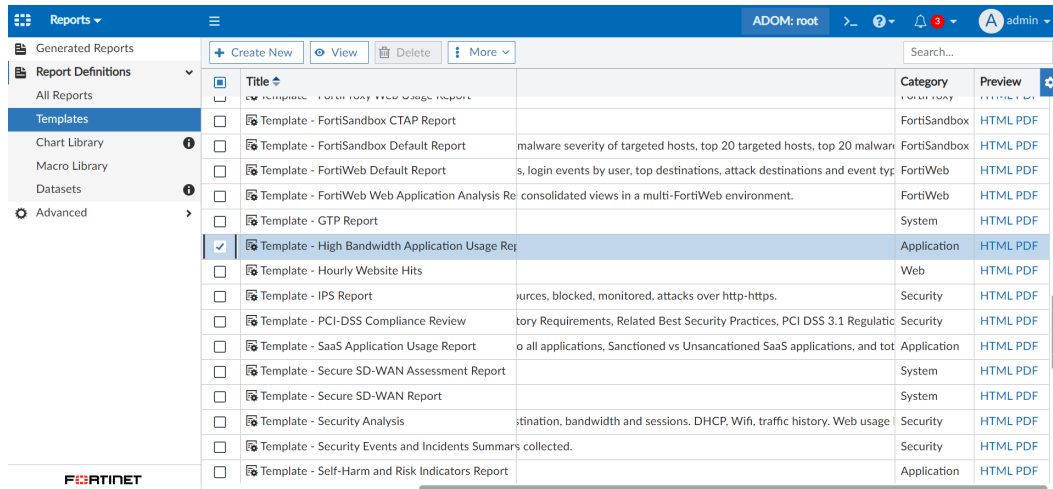
Following is a sample of the report in PDF:



To use the High Bandwidth Application Usage Report template:

1. Go to **Reports > Report Definitions > Templates**.
From the *Preview* column, you can click **PDF** or **HTML** to preview the report in that format.

2. Select the checkbox for *Template - High Bandwidth Application Usage Report*.



Reports				ADOM: root	admin
Generated Reports	+ Create New View Delete More Search...				
Report Definitions					
All Reports	<input type="checkbox"/>	Title		Category	Preview
Templates	<input type="checkbox"/>	Template - FortiSandbox CTAP Report		FortiSandbox	HTML PDF
Chart Library	<input type="checkbox"/>	Template - FortiSandbox Default Report	malware severity of targeted hosts, top 20 targeted hosts, top 20 malware	FortiSandbox	HTML PDF
Macro Library	<input type="checkbox"/>	Template - FortiWeb Default Report	s, login events by user, top destinations, attack destinations and event typ	FortiWeb	HTML PDF
Datasets	<input type="checkbox"/>	Template - FortiWeb Web Application Analysis Re	consolidated views in a multi-FortiWeb environment.	FortiWeb	HTML PDF
Advanced	<input type="checkbox"/>	Template - GTP Report		System	HTML PDF
	<input checked="" type="checkbox"/>	Template - High Bandwidth Application Usage Rep		Application	HTML PDF
	<input type="checkbox"/>	Template - Hourly Website Hits		Web	HTML PDF
	<input type="checkbox"/>	Template - IPS Report	urces, blocked, monitored, attacks over http-https.	Security	HTML PDF
	<input type="checkbox"/>	Template - PCI-DSS Compliance Review	tory Requirements, Related Best Security Practices, PCI DSS 3.1 Regulatio	Security	HTML PDF
	<input type="checkbox"/>	Template - SaaS Application Usage Report	o all applications, Sanctioned vs Unsancationed SaaS applications, and tot	Application	HTML PDF
	<input type="checkbox"/>	Template - Secure SD-WAN Assessment Report		System	HTML PDF
	<input type="checkbox"/>	Template - Secure SD-WAN Report		System	HTML PDF
	<input type="checkbox"/>	Template - Security Analysis	rtination, bandwidth and sessions. DHCP, Wifi, traffic history. Web usage	Security	HTML PDF
	<input type="checkbox"/>	Template - Security Events and Incidents Summary collected.		Security	HTML PDF
	<input type="checkbox"/>	Template - Self-Harm and Risk Indicators Report		Application	HTML PDF

3. From the *More* dropdown, click *Clone* to clone template and make adjustments. You can also click *Create Report* to create a report using the template.

To run the High Bandwidth Application Usage Report:

1. Go to *Reports > Report Definitions > All Reports*.
2. Double-click the row for *High Bandwidth Application Usage Report*. You can find the report using the search bar, for example:

Reports

ADOM: root-new

admin

Generated Reports

Report Definitions

All Reports

Templates

Chart Library

Macro Library

Datasets

Advanced

Run Report

Report

Folder

More

Show Scheduled Only

High

Title

Application Reports

High Bandwidth Application Usage Report

FortiGate Reports

High Bandwidth Application Usage Report

Network Reports

High Bandwidth Application Usage Report

Language

English

English

English

English

English

Cache Status

Time Period

Previous 7 Days

Previous 7 Days

Previous 7 Days

Previous 7 Days

Devices

All Devices

All Devices

All Devices

All Devices

Schedule

Output Profile

Report Own

3. In the *Generated Reports* tab, click *Run Report*.
4. Once the report is generated, click a format in the *Format* column to view the report.

WAN remediation

7.2.0

- [Duplication on-demand when SLAs in the configured service are matched on page 138](#)

Duplication on-demand when SLAs in the configured service are matched

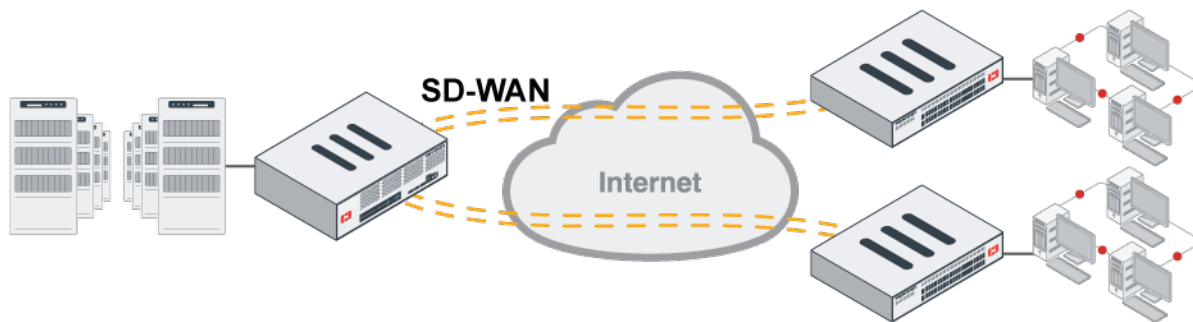


This information is also available in the FortiOS 7.2 Administration Guide:

- [Duplicate packets based on SD-WAN rules](#)

SD-WAN packet duplication can be configured to be performed on-demand only when SLAs in the configured service are matched. When enabled, only the SLA health checks and targets that are used in the service rule are used to trigger the packet duplication.

```
config system sdwan
  config duplication
    edit 1
      set service-id 1
      set packet-duplication on-demand
      set sla-match-service {enable | disable}
    next
  end
end
```



In this example, two performance SLA health checks are configured, health1 and health2. The health1 SLA is used in an SD-WAN service rule called rule1. Packet duplication uses on-demand mode, so packets for duplication are matched based on rule1. It triggers duplication based on the status of the health checks.

Results are shown for various combinations of health check statuses when the SLA match service is enabled or disabled.

To configure SD-WAN:

```
config system sdwan
  set status enable
  set load-balance-mode usage-based
  config zone
    edit "virtual-wan-link"
    next
    edit "SASE"
    next
  end
  config members
    edit 1
      set interface "port5"
      set gateway 10.100.1.1
    next
    edit 2
      set interface "port4"
    next
  end
  config health-check
    edit "health1"
      set server "10.100.2.22"
      set members 0
      config sla
        edit 1
        next
      end
    next
    edit "health2"
      set server "10.100.2.23"
      set members 0
      config sla
        edit 1
        next
      end
    next
  end
  config service
    edit 1
      set name "rule1"
      set mode sla
      set dst "10.100.20.0"
      config sla
        edit "health1"
          set id 1
        next
      end
      set priority-members 2 1
    next
  end
  config duplication
    edit 1
      set service-id 1
      set packet-duplication on-demand
      set sla-match-service enable
```

```

        next
    end
end

```

Results

- When health1 (used in rule1) is out of SLA (sla_map=0x0) and health2 (not used) is in SLA (sla_map=0x1), the packet is duplicated (dup=0x1 (dup)):

```

# diagnose sys sdwan health-check
Health Check(health1):
Seq(1 port5): state(alive), packet-loss(6.000%) latency(5.718), jitter(0.086), mos
(4.404), bandwidth-up(99995), bandwidth-dw(99995), bandwidth-bi(199990) sla_map=0x0
Seq(2 port4): state(alive), packet-loss(3.000%) latency(7.242), jitter(0.025), mos
(4.404), bandwidth-up(99998), bandwidth-dw(99999), bandwidth-bi(199997) sla_map=0x0
Health Check(health2):
Seq(1 port5): state(alive), packet-loss(0.000%) latency(0.700), jitter(0.075), mos
(4.404), bandwidth-up(99995), bandwidth-dw(99995), bandwidth-bi(199990) sla_map=0x1
Seq(2 port4): state(alive), packet-loss(0.000%) latency(0.244), jitter(0.021), mos
(4.404), bandwidth-up(99998), bandwidth-dw(99999), bandwidth-bi(199997) sla_map=0x1

# diagnose firewall proute list
id=2135031809(0x7f420001) vwl_service=1(rule1) vwl_mbr_seq=2 1 dscp_tag=0xfc 0xfc
flags=0x0 tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-65535 path(2)
oif=12(port4) measure=0x0(not measured) dup=0x1 (dup) oif=13(port5) measure=0x0(not
measured) dup=0x1 (dup)
destination(1): 10.100.20.0-10.100.20.255
source wildcard(1): 0.0.0.0/0.0.0.0

```

The sniffer output shows packets leaving from both interfaces in the zone:

```

# diagnose sniffer packet any "port 90" 4
interfaces=[any]
filters=[port 90]
2.403506 port2 in 172.16.205.11.59624 -> 10.100.20.33.90: syn 2098685816
2.403522 port5 out 10.100.1.250.59624 -> 10.100.20.33.90: syn 2098685816
2.403523 port4 out 10.100.1.250.59624 -> 10.100.20.33.90: syn 2098685816

# diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Tie break: cfg
Gen(6), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
Members(2):
1: Seq_num(2 port4), alive, sla(0x0), gid(0), cfg_order(0), cost(0), selected
2: Seq_num(1 port5), alive, sla(0x0), gid(0), cfg_order(1), cost(0), selected
Dst address(1):
10.100.20.0-10.100.20.255

```

- When health1 (used in rule1) is in SLA (sla_map=0x1) and health2 (not used) is out of SLA (sla_map=0x0), the packet is not duplicated (dup=0x0 (not dup)):

```

# diagnose sys sdwan health-check
Health Check(health1):
Seq(1 port5): state(alive), packet-loss(0.000%) latency(0.684), jitter(0.064), mos
(4.404), bandwidth-up(99995), bandwidth-dw(99995), bandwidth-bi(199990) sla_map=0x1
Seq(2 port4): state(alive), packet-loss(0.000%) latency(0.222), jitter(0.015), mos

```

```
(4.404), bandwidth-up(99998), bandwidth-dw(99999), bandwidth-bi(199997) sla_map=0x1
Health Check(health2):
Seq(1 port5): state(alive), packet-loss(6.000%) latency(2.911), jitter(2.328), mos
(1.787), bandwidth-up(99995), bandwidth-dw(99996), bandwidth-bi(199990) sla_map=0x0
Seq(2 port4): state(alive), packet-loss(6.000%) latency(2.566), jitter(2.307), mos
(1.786), bandwidth-up(99998), bandwidth-dw(99999), bandwidth-bi(199997) sla_map=0x0

# diagnose firewall proute list
id=2135031809(0x7f420001) vwl_service=1(rule1) vwl_mbr_seq=2 1 dscp_tag=0xfc 0xfc
flags=0x0 tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-65535 path(2)
oif=12(port4) measure=0x0(not measured) dup=0x0(not dup) oif=13(port5) measure=0x0(not
measured) dup=0x0(not dup)
destination(1): 10.100.20.0-10.100.20.255
source wildcard(1): 0.0.0.0/0.0.0.0
```

The sniffer output shows packets leaving from only one interface:

```
# diagnose sniffer packet any "port 90" 4
interfaces=[any]
filters=[port 90]
3.330376 port2 in 172.16.205.11.38318 -> 10.100.21.33.90: syn 381919014
3.330395 port5 out 10.100.1.2.38318 -> 10.100.21.33.90: syn 381919014
4.327851 port2 in 172.16.205.11.38318 -> 10.100.21.33.90: syn 381919014
4.327855 port5 out 10.100.1.2.38318 -> 10.100.21.33.90: syn 381919014

# diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Tie break: cfg
Gen(4), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
Members(2):
  1: Seq_num(2 port4), alive, sla(0x1), gid(0), cfg_order(0), cost(0), selected
  2: Seq_num(1 port5), alive, sla(0x1), gid(0), cfg_order(1), cost(0), selected
Dst address(1):
  10.100.20.0-10.100.20.255
```

- When the SLA match service is disabled, packets are only duplicated with all of the health checks are out of SLA:

```
config system sdwan
  config duplication
    edit 1
      set service-id 1
      set packet-duplication on-demand
      set sla-match-service disable
    next
  end
end
```

- When health1 is out of SLA (sla_map=0x0) and health2 is in SLA (sla_map=0x1), the packet is not duplicated (dup=0x0(not dup)):

```
# diagnose sys sdwan health-check
Health Check(health1):
Seq(1 port5): state(alive), packet-loss(5.000%) latency(6.587), jitter(0.096), mos
(4.404), bandwidth-up(99995), bandwidth-dw(99995), bandwidth-bi(199990) sla_map=0x0
Seq(2 port4): state(alive), packet-loss(3.000%) latency(3.365), jitter(0.085), mos
(4.404), bandwidth-up(99998), bandwidth-dw(99999), bandwidth-bi(199997) sla_map=0x0
Health Check(health2):
```

```

Seq(1 port5): state(alive), packet-loss(0.000%) latency(0.837), jitter(0.192), mos
(4.404), bandwidth-up(99995), bandwidth-dw(99995), bandwidth-bi(199990) sla_map=0x1
Seq(2 port4): state(alive), packet-loss(0.000%) latency(0.330), jitter(0.081), mos
(4.404), bandwidth-up(99998), bandwidth-dw(99999), bandwidth-bi(199997) sla_map=0x1

# diagnose firewall proute list
list route policy info(vf=root):

id=2135097345(0x7f430001) vwl_service=1(rule1) vwl_mbr_seq=2 1 dscp_tag=0xfc 0xfc
flags=0x0 tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-65535 path(2)
oif=12(port4) measure=0x0(not measured) dup=0x0(not dup) oif=13(port5) measure=0x0
(not measured) dup=0x0(not dup)
destination(1): 10.100.20.0-10.100.20.255
source wildcard(1): 0.0.0.0/0.0.0.0

# diagnose sys sdwan service

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Tie break: cfg
Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
Members(2):
  1: Seq_num(2 port4), alive, sla(0x1), gid(0), cfg_order(0), cost(0), selected
  2: Seq_num(1 port5), alive, sla(0x1), gid(0), cfg_order(1), cost(0), selected
Dst address(1):
  10.100.20.0-10.100.20.255

```

- When both health1 and health2 are out of SLA (sla_map=0x0), the packet is duplicated (dup=0x1 (dup)).



If there are multiple targets in a performance SLA health check, and only one of the targets is used in the service that is defined in the duplication rule, and the SLA match service is disabled, then only that target triggers packet duplication. It is not required for all of the targets in the health check to miss SLA.



www.fortinet.com

Copyright© 2024 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.